

UNIVERSIDADE FEDERAL DE ITAJUBÁ
PROGRAMA DE PÓS-GRADUAÇÃO EM
ENGENHARIA DE PRODUÇÃO

DESENVOLVIMENTO DE *FRAMEWORKS* PARA A MODELAGEM DO
RISCO DE CRÉDITO POR MEIO DE ALGORITMOS DE
CLASSIFICAÇÃO

André Aoun Montevechi

Itajubá, julho de 2023

UNIVERSIDADE FEDERAL DE ITAJUBÁ
PROGRAMA DE PÓS-GRADUAÇÃO EM
ENGENHARIA DE PRODUÇÃO

André Aoun Montevechi

DESENVOLVIMENTO DE *FRAMEWORKS* PARA A MODELAGEM DO
RISCO DE CRÉDITO POR MEIO DE ALGORITMOS DE
CLASSIFICAÇÃO

Dissertação avaliada por banca examinadora em 04 de
julho de 2023.

Orientador: Prof. Dr. Rafael de Carvalho Miranda

Coorientador: Prof. Dr. André Luiz Medeiros

Banca examinadora:

Prof. Dr. José Henrique De Freitas Gomes (UNIFEI)

Prof. Dr. Aneirson Francisco da Silva (UNESP)

Julho de 2023

Itajubá

DEDICATÓRIA

A Deus, minha família e amigos, que sempre estiveram ao meu lado.

AGRADECIMENTOS

A toda minha família, pelo apoio e amor incondicional durante toda minha vida. Em especial à minha mãe, Lea Cristina Aoun Montevechi; Lucas Aoun Montevechi, meu irmão; e José Arnaldo Barra Montevechi, meu pai, mentor e maior inspiração pessoal e profissional. Sem vocês eu não sou, e com vocês aprendi que um lar não é um lugar físico, mas um estado de espírito.

Ao meu orientador, Rafael de Carvalho Miranda, cuja orientação precede meu ingresso na academia, não só pela imensurável contribuição e orientação deste trabalho, mas por mais de uma década de amizade e aprendizados que vão muito além das salas de aula. O professor de verdade inspira e transforma, e com o seu auxílio passei a desejar ser não só um profissional, mas uma pessoa cada vez melhor.

Ao meu coorientador, André Luiz Medeiros, que também tem feito diferença em minha vida antes mesmo do início de minha jornada universitária, pelas valiosas contribuições a este trabalho, por minhas primeiras oportunidades de experiência profissional, assim como sua eterna amizade, disposição, altruísmo e entusiasmo para ensinar e ajudar.

A todos professores e servidores da UNIFEI que, direta ou indiretamente, há anos têm viabilizado o meu crescimento não só como profissional, mas como cidadão. Em especial professores como o Gonzaga e José Leonardo, que foram essenciais na minha trajetória como engenheiro.

Aos amigos da UNIFEI, em especial os companheiros de NEAAD, Alyson, Carlos Henrique, Guilherme, João Victor, Serafim, Wesley, Milena, Arthur e Alice. Em sua companhia, incontáveis dias e noites no laboratório se tornaram não só mais felizes, mas algo para ansiar ao longo das semanas.

Ao Sicoob Sudeste Mais, em especial ao Fernando Carvalho e Erwin Rolf Mádisson Júnior, pelo apoio, ensinamentos, e inesquecível contribuição para o início da minha carreira profissional.

Aos colegas de Embraer, em especial os amigos Gustavo e Henrique.

À CAPES, FAPEMIG e CNPq pelo apoio e incentivo incomensuráveis à pesquisa e desenvolvimento científico no país.

À Universidade Federal de Itajubá, a que devo eterna gratidão, bem como toda a rede de ensino público do nosso Brasil.

A todos vocês, meu muito obrigado!

EPÍGRAFE

“Só conheço uma liberdade, e essa é a liberdade do pensamento.”

Antoine de Saint-Exupéry

RESUMO

A concessão de crédito é uma atividade vital da indústria financeira. Para o funcionamento e sucesso das instituições financeiras, assim como a manutenção do equilíbrio do sistema creditício, a modelagem de risco de crédito tem o papel de avaliar a probabilidade de inadimplência de potenciais devedores com base em dados históricos. Algoritmos de classificação apresentam uma abordagem interessante para esta finalidade na elaboração de modelos para *Credit Scoring*. Desde o surgimento das metodologias analíticas e quantitativas para esta modelagem, persistem na indústria modelos estatísticos, dotados de maior interpretabilidade e fácil implementação. Contudo, com o desenvolvimento na área de *Machine Learning* (ML), surgiram novos algoritmos capazes de trabalhar com um maior volume de dados e com melhor performance preditiva. Estes algoritmos, apesar de nem sempre prontamente transferíveis da academia para a indústria, apresentam uma oportunidade para o desenvolvimento da modelagem do risco de crédito, tendo conseqüentemente despertado um interesse de pesquisadores na área. A literatura, por sua vez, se enfoca na performance dos modelos, dificilmente estabelecendo diretrizes para a otimização do processo de modelagem ou se atentando às regulamentações vigentes para a sua aplicação prática na indústria financeira. Desta forma, esta dissertação, embasada por uma revisão sistemática de literatura, propõe *frameworks* para a modelagem do risco de crédito incorporando o uso de técnicas de ML, pré-processamento e balanceamento de dados, *feature selection* (FS) e otimização de hiperparâmetros (OHP). Além da pesquisa bibliográfica, que possibilita uma familiarização com os principais algoritmos de classificação e as etapas de modelagem apropriadas, o desenvolvimento dos *frameworks* também é fundamentado pela elaboração de centenas de modelos para classificação do risco de crédito, partindo dos algoritmos de Regressão Logística (*Logistic Regression* - LR), Árvores de Decisão (*Decision Trees* - DT), *Support Vector Machines* (SVM), *Random Forest* (RF), assim como *ensembles* de *boosting* e *stacking*, para direcionar de maneira eficiente a construção de modelos robustos e parcimoniosos para a análise do risco na concessão de crédito ao consumidor.

Palavras-chave: Risco de Crédito ao Consumidor, Modelagem do Risco de Crédito; Gestão do Risco de Crédito; Algoritmos de Classificação; *Machine learning*; *Frameworks*.

ABSTRACT

Granting credit is a vital activity in the financial industry. For the success of financial institutions, as well as the equilibrium of the credit system as a whole, it is important that credit risk management systems efficiently evaluate the probability of default of potential debtors based on their historical data. Classification algorithms are an interesting approach to this problem in the form of Credit Scoring models. Since the emergence of quantitative analytical methods with this purpose, statistical models persist as the most commonly chosen method, given their easier implementation and inherent interpretability. However, advances in Machine Learning have developed new and more complex algorithms capable of handling a bigger amount of data, often with an increase in predictive power. These new approaches, although not always readily transferable to practical applications in the financial industry, present an opportunity for the development of credit risk modeling and have piqued the interest of researchers in the field. Nonetheless, researchers seem to focus on model performance, not appropriately setting up guidelines to optimize the modeling process or considering the present regulation for model implementation. Thereby, this dissertation establishes frameworks for consumer credit risk modeling based on classification algorithms while guided by a systematic literature review on the topic. The proposed frameworks incorporate ML techniques, data preprocessing and balancing, feature selection (FS), and hyperparameter optimization (HPO). In addition to the bibliographic research, which introduces us to the main classification algorithms and appropriate modeling steps, the development of the frameworks is also based on experiments with hundreds of models for credit risk classification, using Logistic Regression (LR), Decision Trees (DT), Support Vector Machines (SVM), Random Forest (RF), as well as boosting and stacking ensembles, to efficiently guide the construction of robust and parsimonious models for credit risk analysis in consumer lending.

Keywords: *Consumer Credit Risk, Credit Risk Management; Credit Risk Modelling; Classification Algorithms; Machine learning, Frameworks.*

LISTA DE FIGURAS

Figura 1.1 – Classificação da pesquisa.....	24
Figura 1.2 – Sistema de resolução de problemas.....	26
Figura 1.3 – Etapas da pesquisa.....	27
Figura 2.1 – Taxonomia dos riscos da indústria financeira.....	30
Figura 2.2 – Pilares do Acordo de Basiléia II	37
Figura 2.3 – Framework dos 5Cs na análise de crédito ao consumidor	40
Figura 2.4 – Curva Sigmoide.....	42
Figura 2.5 – Exemplo de uma DT	45
Figura 2.6 – DT aplicada em contexto de CS.....	45
Figura 2.7 – Ilustração do SVM	47
Figura 2.8 – Abordagem de algoritmos de <i>boosting</i>	49
Figura 3.1 – Processo de busca/triagem	62
Figura 3.2 – Produção científica anual de acordo com a base Scopus em 2022	63
Figura 3.3 – Top 10 fontes de pesquisas no tema.....	64
Figura 3.4 – Top 10 países em produção científica no tema	65
Figura 3.5 – Rede de co-citações dos principais autores.....	65
Figura 3.6 – Principais palavras-chave.....	66
Figura 3.7 – Principais classificadores divididos por grupo.....	76
Figura 3.8 – Origem dos <i>datasets</i> empregados na literatura	77
Figura 3.9 – Tipos de pré-processamento encontrados na literatura	81
Figura 3.10 – Métodos de FS encontrados na literatura	84
Figura 3.11 – Algoritmos de <i>resampling</i> encontrados na literatura	86
Figura 3.12 – Algoritmos de OHP encontrados na literatura	87
Figura 3.13 – Métricas de performance mais utilizadas na literatura.....	89
Figura 3.14– Exemplo de curvas ROC.....	91
Figura 4.1 – Planejamento para a modelagem de CSMs.....	104
Figura 4.2 – Distribuição por classe entre atributos do <i>dataset Australian</i> do UCI ML <i>Repository</i>	106
Figura 4.3 – Distribuição por classe entre atributos do <i>dataset German</i> do UCI ML <i>Repository</i>	107
Figura 4.4 – Distribuição por classe entre atributos do <i>dataset Japanese</i> do UCI ML <i>Repository</i>	107

Figura 4.5 – <i>Rank</i> final médio de cada um dos modelos LR.....	112
Figura 4.6 – <i>Rank</i> final médio de cada um dos modelos DT.....	115
Figura 4.7 – <i>Rank</i> final médio de cada um dos modelos SVM	118
Figura 4.8 – <i>Rank</i> final médio de cada um dos modelos RF	121
Figura 4.9 – <i>Rank</i> final médio de cada um dos modelos AdaBoost.....	124
Figura 4.10 – <i>Rank</i> final médio de cada um dos modelos de <i>stacking</i>	127
Figura 5.1 – <i>Rank</i> final dos modelos no <i>dataset</i> UCI <i>australian</i>	130
Figura 5.2 – <i>Rank</i> final dos modelos no <i>dataset</i> UCI <i>german</i>	130
Figura 5.3 – <i>Rank</i> final dos modelos no <i>dataset</i> UCI <i>japanese</i>	131
Figura 5.4 – Resultado da análise de importância de etapas por meio de RF para (a) LR (b) DT (c) SVM (d) RF (e) AdaBoost (f) <i>Stacking</i>	132
Figura 5.5 – <i>Framework</i> de modelagem de CSMs para aplicação prática.....	133
Figura 5.6 – Divisão dos classificadores conforme complexidade, interpretabilidade e performance	135
Figura 5.7 – <i>Framework</i> geral e irrestrito para modelagem de CSMs	138

LISTA DE TABELAS

Tabela 3.1 – Interpretações do valor de AUC	91
Tabela 3.2 – Resultados das 5 principais métricas na literatura para o <i>australian</i> UCI dataset	93
Tabela 3.3 – Resultados das 5 principais métricas na literatura para o <i>german</i> UCI dataset ..	95
Tabela 3.4 – Resultados das 5 principais métricas na literatura para o <i>japanese</i> UCI dataset	98
Tabela 4.1– Descrição dos <i>datasets</i>	106
Tabela 4.2 – Resultados para os modelos LR.....	109
Tabela 4.3 – Parâmetros da LR e amplitude das faixas de variação na <i>grid search</i>	111
Tabela 4.4 – Resultados para os melhores modelos LR.....	112
Tabela 4.5 – Resultados para os modelos DT	113
Tabela 4.6 – Parâmetros da DT e amplitude das faixas de variação na <i>grid search</i>	115
Tabela 4.7 – Resultados para os melhores modelos DT.....	115
Tabela 4.8 – Resultados para os modelos SVM	116
Tabela 4.9 – Parâmetros da SVM e amplitude das faixas de variação na <i>grid search</i>	118
Tabela 4.10 – Resultados para os melhores modelos SVM	118
Tabela 4.11 – Resultados para os modelos RF	119
Tabela 4.12 – Parâmetros da RF e amplitude das faixas de variação na <i>grid search</i>	121
Tabela 4.13 – Resultados para os melhores modelos RF	121
Tabela 4.14 – Resultados para os modelos AdaBoost.....	122
Tabela 4.15 – Resultados para os melhores modelos AdaBoost	124
Tabela 4.16 – Resultados para os modelos de <i>stacking</i>	125
Tabela 4.17 – Resultados para os melhores modelos de <i>stacking</i>	127
Tabela 5.1 – Resultados médios dos modelos por <i>dataset</i>	128
Tabela 5.2 – Resultados dos melhores modelos por métrica no <i>dataset</i> UCI <i>australian</i>	128
Tabela 5.3 – Resultados dos melhores modelos por métrica no <i>dataset</i> UCI <i>german</i>	129
Tabela 5.4 – Resultados dos melhores modelos por métrica no <i>dataset</i> UCI <i>japanese</i>	129

LISTA DE QUADROS

Quadro 2.1 – Comparação entre algoritmos de agregação <i>bagging</i> e <i>boosting</i>	50
Quadro 3.1 – Classificadores utilizados em artigos que empregaram dados de crédito corporativo.....	67
Quadro 3.2 – Classificadores novel utilizados em artigos que empregaram dados de crédito ao consumidor	70
Quadro 3.3 – Modelos estatísticos <i>baseline</i> em artigos sobre crédito ao consumidor	73
Quadro 3.4 – Modelos de ML individuais <i>baseline</i> encontrados em artigos sobre crédito ao consumidor	73
Quadro 3.5 – Modelos <i>ensemble baseline</i> de ML encontrados em artigos sobre crédito ao consumidor	75
Quadro 3.6 – Etapas de pré-processamento encontradas na literatura	81
Quadro 3.7 – Métodos de FS encontrados na literatura	82
Quadro 3.8 – Métodos de balanceamento de classes encontrados na literatura	85
Quadro 3.9 – Métodos de OHP encontrados na literatura	88
Quadro 3.10 – Matriz de Confusão	90
Quadro 4.1– Algoritmos de classificação utilizados	105

LISTA DE ABREVIações

AdaBoost	<i>Adaptive Boosting</i>
AIB	<i>Advanced Internal Ratings-Based Approach</i>
AIC	<i>Akaike Information Criterion</i>
ALasso	<i>Adaptive LASSO</i>
ASVM	<i>Additive SVM</i>
AUPRC	<i>Area Under the Precision-Recall Curve</i>
B&B	<i>Branch & Bound</i>
BA	<i>Bat Algorithm</i>
BAC	<i>Balanced Accuracy</i>
BBHFS	<i>B&B Based Hybrid Feature Selection</i>
BEEF	<i>Balanced English Explanations of Forecasts</i>
BIC	<i>Bayesian Information Criterion</i>
BMTS	<i>Bisection Method based on Tabu Search</i>
BNET	<i>Bayesian Network</i>
BOA	<i>Bayesian Hyper-parameter Optimization</i>
BRT	<i>Boosted Regression Trees</i>
BS	<i>Brier Score</i>
CART	<i>Classification And Regression Tree</i>
CF	<i>Collaborative Filtering</i>
CFS	<i>Correlation-based Feature Selection</i>
ChAID	<i>Chi-square Automatic Interaction Detection</i>
CI	<i>Conditional Inference Trees</i>
CIF	<i>Canonical Interval Forest</i>
CKNN	<i>Citation-KNN</i>
CS	<i>Credit Scoring</i>
CSM	<i>Credit Scoring Model</i>
DT	<i>Árvore de Decisão</i>
EAD	<i>Exposure at Default</i>
ECAI	<i>External Credit Assessment Institution</i>
EL	<i>Expected Loss</i>
EMDD	<i>Emotion Mode Diversity Density</i>

ET	<i>Extra Trees</i>
FIRB	<i>Basic Internal Rating</i>
FN	Falso Negativo
FNN	<i>Feedforward NN</i>
FNR	Taxa de Falsos Negativos
FPR	Falso Positivo
FPR	Taxa de Falsos Positivos
GA	Algoritmo Genético
GAM	<i>Generalized Additive Model</i>
GBDT	<i>Gradient Boosted DT</i>
GBM	<i>Gradient Boosting</i>
GC	<i>Gini Coefficient</i>
HCES	<i>Hill-Climbing Ensemble Selection</i>
HWA	<i>Hyper-parameter WA</i>
IA	Inteligência Artificial
ICS	<i>Improved Cuckoo Search</i>
IEEE	<i>Institute of Electrical and Electronics Engineers</i>
IRB	<i>Internal Ratings-Based Approach</i>
KDD	<i>Knowledge Discovery Data</i>
KNN	<i>k-Nearest Neighbors</i>
KSS	<i>Kolmogorov-Smirnov Statistic</i>
LASSO	<i>Least Absolute Shrinkage Selection Operator</i>
LDA	Análise Discriminante Linear
LDP	<i>Low Default Portfolio</i>
LGD	<i>Loss Given Default</i>
LIME	<i>Local Interpretable Model-agnostic Explanations</i>
LORE	<i>Local Rule-based Explanation</i>
LR	Regressão Logística
MARS	<i>Multivariate Adaptive Regression Spline</i>
MaxRel	<i>Maximum Relevance</i>
MCC	<i>Matthew's Correlation Coefficient</i>
MCMC	Markov Chain Monte-Carlo
MDA	<i>Multiple Discriminant Analysis</i>

ML	<i>Machine Learning</i>
MLP	<i>Multilayer Perceptron</i>
mRMR	<i>Minimum Redundancy Maximum Relevance</i>
MSVM	<i>Multi-class SVM</i>
MV	<i>Majority Vote</i>
NAP	RF based New Approach
NB	Naïve Bayes
NN	Redes Neurais
NPV	<i>Negative Predictive Value</i>
OE	Objeto de Estudo
OCC	<i>One-class Classification</i>
ORF	<i>Optimized RF</i>
PART	<i>Partial DT</i>
PCA	<i>Principal Component Analysis</i>
PCC	<i>Percentage of Correctly Classified Cases</i>
PD	<i>Probability of Default</i>
PGI	<i>Partial Gini Index</i>
PLR	<i>Penalized LR</i>
PSO	<i>Particle Swarm Optimization</i>
QP	Questão de Pesquisa
RBF	<i>Radial Basis Function</i>
RC	<i>Random Committee</i>
RF	<i>Random Forest</i>
RFE	<i>Recursive Feature Elimination</i>
RL	Revisões de Literatura
ROS	<i>Random Oversampling</i>
RS	<i>Random Subspace</i>
RSF	<i>Random Survival Forest</i>
RSL	Revisão Sistemática de Literatura
RUS	<i>Random Undersampling</i>
SA	<i>Standardized Approach</i>
SBS	<i>Sequential Backward Selection</i>
SFS	<i>Sequential Forward Selection</i>

SMOTE	<i>Synthetic Minority Oversampling Technique</i>
SOM	<i>Self-organizing Maps</i>
SU	<i>Symmetrical Uncertainty</i>
SU-FCBC	<i>SU-Fast Correlation-Based Filter</i>
SVDF	<i>Simple MV Decision Forest</i>
SVM	<i>Support Vector Machine</i>
SVM-L	SVM Linear
SVR	<i>Support Vector Regression</i>
T&F	<i>Taylor and Francis</i>
TFFS	<i>Two-stage Filter Feature Selection</i>
TN	Verdadeiro Negativo
TNR	Taxa de Verdadeiros Negativos
TP	Verdadeiro Positivo
TPR	Taxa de Verdadeiros Positivos
UCI	<i>University of California Irvine</i>
WA	<i>Weighted Averaging</i>
WoS	<i>Web of Science</i>
WVDF	<i>Weighted MV Decision Forest</i>
XAI	<i>Explainable AI</i>
XGBoost	<i>Extreme Gradient Boosting</i>

SUMÁRIO

1. INTRODUÇÃO.....	18
1.1 Objetivos.....	21
1.2 Justificativa.....	21
1.3 Método de Pesquisa.....	24
1.3.1 Classificação da Pesquisa.....	24
1.3.2 Modelagem e Simulação.....	25
1.3.3 Etapas do Método a Serem Desenvolvidas.....	26
1.4 Delimitação da Pesquisa.....	28
1.5 Estrutura do Trabalho.....	28
2. FUNDAMENTAÇÃO TEÓRICA.....	29
2.1 Risco de Crédito.....	29
2.2 Administração do Risco de Crédito.....	31
2.2.1 <i>Credit Scoring</i>	31
2.2.2 Regulamentação.....	35
2.3 Metodologias Convencionais de <i>Credit Scoring</i>	39
2.3.1 Abordagem Intuitiva.....	39
2.3.2 Regressão Logística.....	41
2.4 Algoritmos de <i>Machine Learning</i>	42
2.4.1 Modelos Individuais.....	44
2.4.2 <i>Ensemble Learning</i>	47
2.5 Balanceamento de Classes.....	51
2.5.1 SMOTE E Tomek's <i>links</i>	53
2.5.2 <i>One-class classification</i>	53
2.6 Otimização de Hiper Parâmetros.....	54
2.7 <i>Feature Selection</i>	55
2.8 Interpretabilidade.....	56
3. REVISÃO SISTEMÁTICA DE LITERATURA.....	59
3.1 O Método de Revisão Sistemática de Literatura.....	59
3.2 Aplicação do Método de Revisão Sistemática de Literatura.....	59
3.2.1 Planejamento.....	60
3.2.2 Pesquisa e Triagem dos Artigos.....	61

3.2.3	Análise e Síntese	62
3.2.4	Apresentação dos Resultados	63
3.3	Resultados e discussões da Revisão Sistemática de Literatura.....	63
3.3.1	Modelagem.....	66
3.4	Considerações Finais da Revisão Sistemática de Literatura.....	101
4.	MODELAGEM	104
4.1	Descrição e Análise Exploratória dos Dados.....	106
4.2	Pré-processamento dos <i>datasets</i>	108
4.3	Aplicação e Resultados	108
4.3.1	Modelos LR.....	109
4.3.2	Modelos DT.....	113
4.3.3	Modelos SVM	116
4.3.4	Modelos RF.....	119
4.3.5	Modelos AdaBoost.....	122
4.3.6	Modelos de <i>Stacking</i>	125
5.	ANÁLISE DOS RESULTADOS E DESENVOLVIMENTO DOS <i>FRAMEWORKS</i> 128	
5.1	<i>FRAMEWORKS</i> PROPOSTOS	133
6.	CONCLUSÕES E PROPOSTAS PARA ESTUDOS FUTUROS	139
	REFERÊNCIAS BIBLIOGRÁFICAS	143
	ANEXO A – MELHORES MÉTRICAS DE PERFORMANCE PARA TODOS OS ARTIGOS DE CLASSIFICAÇÃO DE RISCO DE CRÉDITO AO CONSUMIDOR ..	159
	ANEXO B – MELHORES PARÂMETROS PARA OS MODELOS LR COM A ETAPA DE OHP	166
	ANEXO C – PARÂMETROS DOS MODELOS LR COM OHP.....	167
	ANEXO D – PARÂMETROS DOS MODELOS SVM COM OHP	169
	ANEXO E – PARÂMETROS DOS MODELOS RF COM OHP.....	170
	APÊNDICE - Artigos Publicados em Congressos e Periódicos	172

1. INTRODUÇÃO

A palavra “crédito” originou-se do Latim *creditum*, cujo significado é “algo passado em confiança a outrem”, particípio passado do verbo *credere*, que significa acreditar ou confiar. Na indústria financeira, a concessão de crédito representa o fornecimento de recursos financeiros para um contratante, que assume o papel de devedor (ou tomador), pelo titular do crédito, chamado de “credor” e que, com legítima expectativa do adimplemento do empréstimo e suas prestações convencionadas, consolida a relação contratual obrigacional entre ambas as partes.

O sistema de concessão de crédito existe desde os primórdios das civilizações humanas (THOMAS *et al.*, 2002), representando uma atividade intrinsecamente ligada a história do comércio (ZHOU *et al.*, 2021). Configurando sua principal atividade comercial, a indústria financeira sobrevive e progride emprestando recursos financeiros para aqueles que os requerem, estando diretamente envolvida na saúde econômica de um país (GIRI *et al.*, 2021). Atualmente, instituições financeiras liberam créditos diariamente para pessoas físicas e jurídicas por diferentes meios, especialmente em países como o Brasil, em que a maior parte do capital que circula no mercado é oriundo de linhas de crédito bancárias. O sistema de concessão de crédito é essencial e indissociável da indústria financeira (COENEN *et al.*, 2022). Contudo, configura um procedimento com risco inerente para o credor: o não recebimento devido do valor emprestado ou risco de crédito.

De acordo com Munkhdalai *et al.* (2019), o risco de crédito é um dos principais riscos enfrentados pela indústria financeira no mercado e, ao longo dos anos, diversos fatores contribuíram para uma importância cada vez maior de sua gestão apropriada (NIKLIS; DOUMPOS; ZOPOUNIDIS, 2014). Controlando-o precisamente, uma instituição de crédito pode não só melhor administrar sua sustentabilidade e estabilidade financeira (MUNKHDALAI *et al.*, 2019), como contribuir para a manutenção do equilíbrio do sistema creditício como um todo (KHEMAKHEM; BEN SAID; BOUJELBENE, 2018). Por outro lado, um aumento nas taxas de inadimplência resulta em grandes perdas para estas instituições, assim como profundos impactos econômicos e, de acordo com Shi, Zhane e Qiu (2013), é inviável que elas recusem a maior parte dos clientes na tentativa de evitar tal risco.

O risco de crédito pode ser reduzido tanto na diversificação do portfólio de crédito da instituição, otimizando-o de forma a minimizar o risco e aumentar a sua competitividade; quanto na forma com que o risco é mensurado, usando ferramentas capazes de determinar precisamente a probabilidade de inadimplência de um tomador (ORLOVA, 2020). Assim, surge

na indústria o desafio para as instituições credoras de avaliar o risco de inadimplência de possíveis tomadores e distingui-los conforme a expectativa de seu comportamento frente ao potencial contrato, identificando os melhores candidatos para a concessão de linhas de crédito com o menor risco possível (TRIVEDI, 2020). O desenvolvimento de soluções para tal é de suma importância, desempenhando um papel crucial no seu funcionamento e sucesso.

As diferentes formas de se aferir o risco de potenciais devedores caracterizam então um processo importante na gestão do risco de crédito pela indústria financeira. A este processo dá-se o nome de *Credit Scoring* (CS), ferramenta comum de auxílio à tomada de decisão que busca prever a probabilidade de o tomador apresentar determinado comportamento com base em um grupo de direcionadores de risco (FLOREZ-LOPEZ; RAMON-JERONIMO, 2015).

Credit scores de consumidores são elaborados antes da concessão do crédito para observar o risco envolvido, tendo o seu setor responsável, de acordo com Trivedi (2020), uma importância econômica substancial. O CS, por sua vez, surgiu muito após os sistemas de concessão de crédito. Enquanto estima-se que já existiam sistemas de empréstimo no século XX a.C., metodologias de CS consolidaram-se apenas cerca de seis décadas atrás. No início, os *credit scores* de potenciais clientes eram preparados por instituições de crédito a partir de seus respectivos pedidos, tarefa atribuída aos analistas de crédito (ZHOU *et al.*, 2021). Contudo, trabalhando manualmente e com análises individuais de possíveis tomadores, como em qualquer procedimento intuitivo, analistas de crédito encontram-se suscetíveis a decisões inconsistentes, embasadas fortemente em suas próprias experiências (MALHOTRA; MALHOTRA, 2000). Além disso, esta abordagem também se revelou não vantajosa por ser incompatível com a crescente demanda em grandes instituições em razão de sua baixa praticidade e morosidade.

Além da ascendente demanda comercial por crédito, segundo Niklis, Doumpos e Zopounidis (2014), em decorrência de inevitáveis períodos históricos de crise e recessão, cada vez mais, instituições financeiras, assim como governos e agências reguladoras, passaram a reconhecer a crescente necessidade por técnicas analíticas quantitativas para otimizar e monitorar as decisões relacionadas ao risco de crédito (MARTENS *et al.*, 2010). Oportunamente, logo surgiram tentativas de se estimar o risco envolvido em um empréstimo e a probabilidade de o débito ser devidamente pago e, hoje em dia, CS tem sido utilizado amplamente na tomada de decisão para diferentes propósitos (ZHOU *et al.*, 2021; TRIVEDI, 2020; THOMAS, 2000), estendendo-se para diversos setores, da aprovação de pedidos de concessão de crédito até a previsão de falência de empresas (AGHAEIRAD; CHEN; RIBEIRO, 2017).

Um modelo de CS (ou *Credit Scoring Model* - CSM) analisa características e performances de empréstimos passados para prever a probabilidade de inadimplência de um potencial tomador com base em uma série de indicadores – usualmente dados demográficos e financeiros (ZHOU *et al.*, 2021) – e assinalá-lo a um grupo de risco (THOMAS *et al.*, 2002; NIKLIS; DOUMPOS; ZOPOUNIDIS, 2014). Estes modelos são criados pelas próprias instituições ou pelos *bureaus* de proteção ao crédito ou *External Credit Assessment Institutions* (ECAIs) e, segundo Niklis, Doumpos e Zopoudinis (2014), podem ser construídos por meio de uma variedade de técnicas estatísticas, de *data mining* e do campo da pesquisa operacional.

Em geral, a maior parte destes métodos podem ser divididos em abordagens estatísticas, das quais se destacam a Regressão Logística (ou *Logistic Regression* - LR) e a Análise Discriminante Linear (ou *Linear Discriminant Analysis* - LDA), e abordagens de *Machine Learning* (ML) (SHI; ZHANG; QIU, 2013; MARQUÉS *et al.*, 2013; DUMITRESCU, 2021). Enquanto LR e LDA, que são relativamente simples e de fácil interpretação, permanecem a referência na indústria (SHI; ZHANG; QIU, 2013; DUMITRESCU, 2022), as abordagens de ML tem sido motivo de interesse em numerosos estudos voltados para CS (WANG *et al.*, 2020). Isto tem gerado grande impacto sobre as metodologias habituais de classificação, possibilitando o uso de grandes quantidades de dados históricos para a modelagem (LESSMANN *et al.*, 2015) e gerando resultados promissores, principalmente pela habilidade de superar as técnicas clássicas e tradicionais (CHOPRA; BHILARE, 2018). Por isso, esses resultados não podem ser ignorados, uma vez que melhorias em frações percentuais de um CSM podem implicar em economia significativa de recursos (BAESENS *et al.*, 2003; WEST, 2000; HUANG *et al.*, 2007).

Ramo da inteligência artificial, ML estuda algoritmos capazes de aprender com experiência, isto é, programas computacionais capazes de otimizar critérios de performance a partir de dados de treinamento ou experiência passada (ALPAYDIN, 2020). Enquanto modelos de aprendizado estatístico buscam inferir a relação entre variáveis, a prioridade em ML é a performance da previsão, sendo uma de suas aplicações mais frequentes justamente a classificação. Algoritmos de ML como *Support Vector Machines* (SVM), Redes Neurais (ou *Neural Networks* - NN), *Decision Trees* (DT), *Random Forest* (RF), *k-Nearest Neighbors* (KNN) e Naïve Bayes (NB) apresentam relevante potencial discriminatório e já têm sido explorados no contexto de CS. Contudo, não existe um algoritmo invariavelmente superior aos demais, sendo a modelagem do risco de crédito um processo contínuo. Além disso, a escolha do mecanismo de classificação não é a única decisão a ser tomada na modelagem de um CSM, existindo diversas outras etapas que podem não só aperfeiçoar a capacidade preditiva do modelo

como também usufruir das técnicas de ML. Diante disso, este trabalho busca desenvolver *frameworks* para a modelagem do risco de crédito ao consumidor por meio de ML supervisionado, construindo modelos *baseline* a partir de diferentes algoritmos de classificação e combinando-os com as demais etapas de modelagem para buscar desenvolver uma proposta de direcionamento para projetos na área.

1.1 Objetivos

O presente estudo objetiva o desenvolvimento de *frameworks* para a modelagem de CSMs que integra, de forma inovadora, o uso de algoritmos de ML, otimização de hiper parâmetros (OHP), *feature selection* (FS), *resampling* e interpretabilidade para a solução de problemas de CS. Para isso, os seguintes objetivos específicos são apresentados a seguir:

- Conduzir o desenvolvimento, modelagem, otimização, treinamento e validação de diferentes modelos *baseline* partindo de algoritmos de classificação de ML e abordagens estatísticas para *benchmarking*. Objetiva-se variar a combinação entre algoritmos de classificação e as demais etapas de modelagem para compreender melhor o impacto das interações entre eles;
- Avaliar as métricas de performance dos modelos em diferentes bases de dados, observando as melhores combinações entre etapas de modelagem e algoritmos, e integrando-as devidamente ao *framework*;

O trabalho busca, portanto, contribuir para a literatura com a proposta de diretrizes para a condução de projetos de modelagem de CSMs, preenchendo uma lacuna na literatura e possibilitando contribuições multidimensionais, assistindo na avaliação de novas tecnologias no campo da inteligência computacional e *data science* e desenvolvendo um método com potencial contribuição ímpar para a indústria financeira e a economia nacional. Espera-se, desta forma, que esta pesquisa sirva de base e auxilie pesquisadores e profissionais na modelagem de CSMs e na administração do risco de crédito em geral.

1.2 Justificativa

O competitivo cenário em que atuam as instituições financeiras nos dias de hoje configura um mercado extremamente desafiador. Grandes mudanças na indústria financeira pareadas com uma competição cada vez mais intensa e diversificada colocam estas instituições sob constante pressão (CHOPRA; BHILARE, 2018). Segundo Orlova (2020), sua estabilidade

depende de uma gestão adequada e eficiente. Sendo operações de concessão de crédito as principais atividades destas instituições, há de haver uma administração adequada de riscos, com planejamento especial a respeito do nível do risco de crédito, uma de suas maiores e mais sérias ameaças (KHEMAKHEM; BEN SAID; BOUJELBENE, 2018; LESSMANN *et al.*, 2015).

Conseqüentemente, instituições financeiras enfrentam um dilema ao tentar aumentar o volume de operações de crédito sem aumentar sua exposição e vulnerabilidade aos efeitos de seu inadimplemento (MALHOTRA; MALHOTRA, 2000). Desta forma, a análise do risco de crédito é um assunto importante no campo da gestão de riscos financeiros, tendo se tornado o principal objetivo do setor bancário (HAN; HAN; ZHAO, 2013) e sendo indispensável para a sobrevivência de inúmeros negócios (FLOREZ-LOPEZ; RAMON-JERONIMO, 2015). Para isso, são requeridas técnicas sofisticadas de modelagem na elaboração dos CSMs (CHEN; XIANG, 2017; LESSMANN *et al.*, 2015). Enquanto critérios permissivos demais estimulam a inadimplência e insolvência, critérios excessivamente rigorosos no processo de concessão de crédito podem dissuadir potenciais clientes (PŁAWIAK; ABDAR; ACHARYA, 2019; FLOREZ-LOPEZ; RAMON-JERONIMO, 2015). Assim, novas formas de construir e otimizar CSMs constituem um assunto de grande importância e interesse para a academia e indústria.

Leo, Sharma e Maddulety (2019) afirmam que, até 2025, a forma com que a indústria financeira enfrenta os seus principais riscos deve ser diferente da forma como ocorre tradicionalmente. O surgimento e aprofundamento de normas regulamentadoras, aumento das expectativas dos clientes e evolução dos riscos enfrentados estimulam cada vez mais mudanças dentro o campo da administração de risco. Constantemente, novos produtos, serviços e técnicas gerenciais devem e têm sido desenvolvidas por meio da aplicação de novas tecnologias e avanços de metodologias analíticas capazes de auxiliar a gestão. Em seu trabalho, Finlay (2011) determinou que uma grande quantia do capital em perdas por dívidas de consumidores poderia ter sido evitada por meio de melhoras na aferição do risco de crédito. À vista disso, a implementação destas inovações adquire um caráter de urgência, demonstrando potencial não só para aprimorar a performance das instituições financeiras como da economia do país como um todo (ZHOU *et al.*, 2021), sendo o endividamento e inadimplência problemas que continuam afetando profundamente o consumidor brasileiro.

Para referência, em setembro de 2022, a Confederação Nacional do Comércio de Bens, Serviços e Turismo (CNC) estimou que o endividamento afetava 79% das famílias no Brasil. Este número vem de sua Pesquisa de Endividamento e Inadimplência do Consumidor (PEIC), que considera dívidas a vencer no cheque pré-datado, cheque especial, carnês, crédito

consignado, empréstimos e prestações de casa e carro, e evidenciou uma aceleração no atraso do pagamento de contas de consumo ou dívidas, atingindo 29,6% dos lares no país, em distintas faixas de renda, e registrando alta da série histórica iniciada em 2010.

Apesar de se tratar de dados afetados por inúmeros fatores socioeconômicos, esta alta pode ser atenuada por melhorias na modelagem do risco de crédito por parte das instituições de crédito no país, seja por uma aprovação mais rigorosa de tomadores ou na própria determinação das taxas de juros. Neste contexto, Zheng (2019) alerta que, para se garantir a segurança de transações e quantificar o risco de crédito envolvido nas operações financeiras apropriadamente, existe uma demanda inadiável por algoritmos de ML capazes de auxiliar na compreensão da situação financeira de potenciais devedores e de prever a probabilidade de sua inadimplência em tempo limitado.

Na última década, técnicas de ML têm sido cada vez mais utilizadas por bancos e *fintechs* para a elaboração de CSMs (DUMITRESCU *et al.*, 2022; HURLIN; PÉRIGNON, 2019). Historicamente, CSMs computadorizados são altamente confiáveis no aperfeiçoamento da performance de ativos financeiros (MARQUES *et al.*, 2013). Trata-se de sistemas automatizados que se mostram superiores às metodologias convencionais, trazendo grandes vantagens como maior objetividade e confiabilidade e a redução dos custos envolvidos na afeição do risco de novos clientes (GOH; LEE, 2019). Sendo a melhoria da performance dos modelos a prioridade em CS (GIRI *et al.*, 2021), pode-se afirmar então que, apesar de não constituírem o padrão na indústria, técnicas de ML são uma realidade cada vez maior nas abordagens de CS.

Por fim, na elaboração de CSMs, tradicionalmente não existe uma única técnica estatística ou de ML com performance invariavelmente superior, sendo que sua escolha depende dos detalhes do conjunto de dados analisado (HAND; HENLEY, 1997; ZHOU *et al.*, 2021). Os resultados também variam de acordo com as etapas que integram ou não o processo de modelagem. Estas etapas – destacando-se aqui especialmente o pré-processamento de dados, *resampling*, FS e OHP – também podem, segundo Zhou *et al.* (2021), aperfeiçoar significativamente a performance de um modelo.

Apesar de existirem na literatura numerosos exemplos de estudos que examinam e comparam diferentes metodologias de classificação e seu desempenho em CS mediante diferentes combinações destas outras variáveis do processo, há uma escassez na literatura da proposta de *frameworks* elaborados metodicamente e que direcionam a modelagem neste contexto, determinando e estruturando suas etapas de forma a otimizar os seus resultados. Capaz de guiar as ações do pesquisador ou profissional ao longo do processo de modelagem de

CSMs, *frameworks* bem definidos para tal podem facilitar o desenvolvimento e otimizar os resultados dos modelos, enfocando-se nos algoritmos e etapas apropriados e evitando a inclusão de procedimentos incorretos ou ineficazes, além de tornar mais célere e robusto o processo de modelagem.

Logo, este estudo tem importância prática e teórica, abordando um assunto de valor e relevância para a indústria financeira e a economia nacional sob uma ótica de tecnologias emergentes com aplicações cada vez mais evidentes na área, e buscando preencher uma lacuna encontrada na literatura consultada para contribuir para a consolidação das aplicações de ML em CS.

1.3 Método de Pesquisa

1.3.1 Classificação da Pesquisa

A classificação do método de pesquisa empregados neste trabalho corrobora com a proposta de classificação de pesquisa apresentada na Figura 1.1.

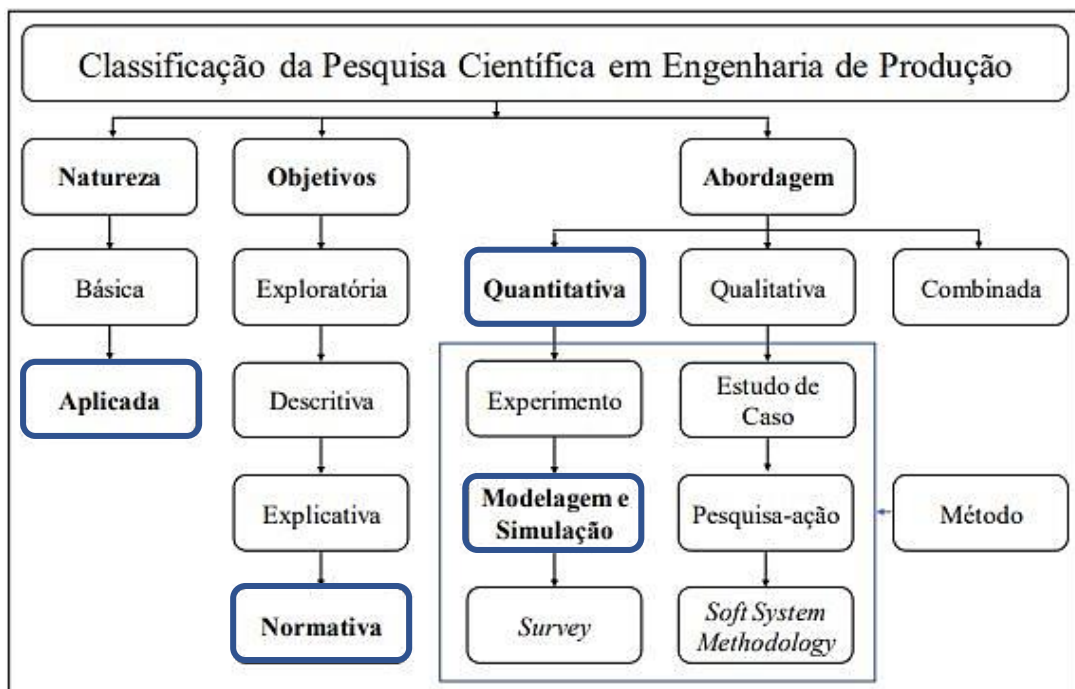


Figura 1.1 – Classificação da pesquisa
Fonte: Adaptado de Miguel *et al.* (2018)

Turrioni e Mello (2012) classificam as pesquisas científicas na área de engenharia de produção em quatro dimensões básicas: quanto à natureza, objetivos, abordagem e método. Esta

pesquisa pode ser classificada como “aplicada”, pois segundo Turrioni e Mello (2012) uma pesquisa aplicada “caracteriza-se por seu interesse prático, isto é, que os resultados sejam aplicados ou utilizados imediatamente na solução de problemas que ocorrem na realidade”.

Quanto ao objetivo da pesquisa, pode-se classificá-lo como exploratório, descritivo, explicativo ou normativo. De acordo com Bertrand e Fransoo (2002), uma pesquisa normativa é aquela que busca encontrar soluções ótimas para novas definições ou está interessada no desenvolvimento de políticas e estratégias para aperfeiçoar resultados encontrados na literatura. Logo, esta pesquisa possui objetivos normativos, uma vez que objetiva propor um *framework* para a modelagem de CSMs, visando a obtenção de resultados ótimos em problemas de CS.

Além disto, este trabalho pode ser entendido como uma abordagem quantitativa, a qual parte da teoria para se formular hipóteses, coletar dados por meio de observações, analisá-los e gerar resultados (MIGUEL *et al.*, 2018). Dentro desta abordagem existem diversos métodos e, para esta pesquisa, o método de modelagem e simulação mostrou-se o mais adequado, uma vez que, trata-se de uma abordagem em que se manipulam variáveis em um modelo.

Acrescentando às classificações anteriores, Gil (2002) propõe a classificação da pesquisa quanto aos procedimentos técnicos utilizados, em especial referente ao meio de coleta de dados. Neste critério, o autor sugere que uma pesquisa pode ser classificada em bibliográfica, documental, experimental, *ex-post facto*, coorte, levantamento, de campo, estudo de caso, pesquisa-ação e participante. De acordo com suas peculiaridades esta dissertação pode ser entendida como experimental, o que, segundo Gil (2002), é aquela que “consiste em determinar um objeto de estudo, selecionar as variáveis que seriam capazes de influenciá-lo, definir as formas de controle e de observação dos efeitos que a variável produz no objeto”.

1.3.2 Modelagem e Simulação

Mitroff *et al.* (1974) propuseram uma metodologia científica de resolução de problemas voltadas para área de pesquisa operacional, e em especial para modelagem e simulação. Os autores criaram uma subdivisão do projeto em quatro etapas, a saber: conceitualização, modelagem, solução do modelo e implementação. Esta dissertação adotará este método visto que apresenta robustez e é amplamente discutido pela literatura na área de pesquisa operacional, sendo ainda considerado um método válido e aplicável nos tempos atuais (WOLLMANN; TORTATO, 2019). A Figura 2.2 apresenta estas fases e as ações necessárias para evoluir entre as fases.

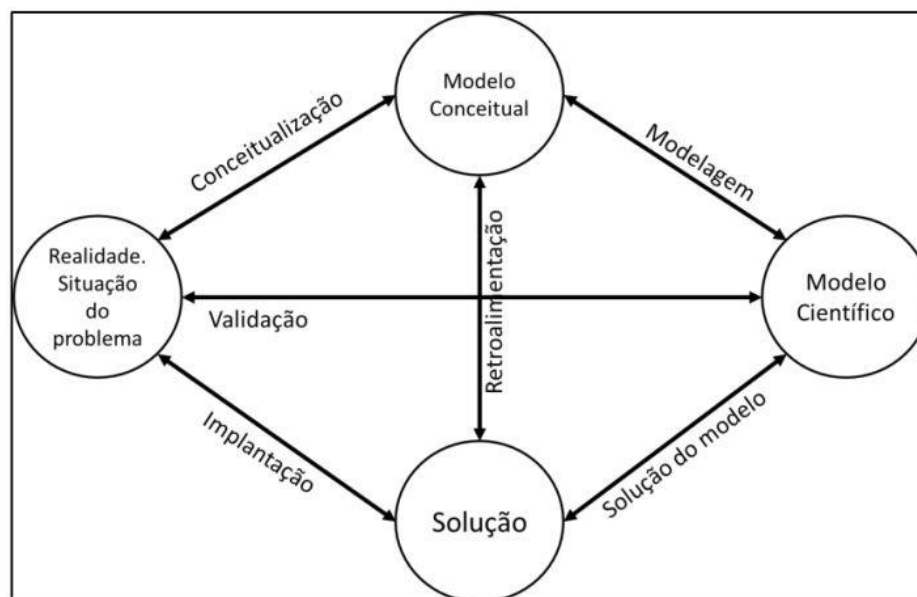


Figura 1.2 – Sistema de resolução de problemas
 Fonte: Adaptado de Mitroff *et al.* (1974)

Bertrand e Fransoo (2002) ainda propõem a classificação das pesquisas quantitativas com base nas etapas da metodologia de Mitroff *et al.* (1974), sendo divididas em quatro tipos a depender do ciclo do projeto, a saber:

- Empírica descritiva: projeto em que o pesquisador envolve o ciclo conceitualização, modelagem e validação;
- Empírica normativa: engloba as ações de conceitualização, modelagem, solução do modelo e implantação;
- Axiomática descritiva: envolve a ação de modelagem, passando do modelo conceitual para o modelo científico;
- Axiomática normativa: envolve as ações de modelagem e solução do modelo.

Dada a classificação proposta, esta pesquisa pode ser entendida como empírica normativa, na medida em que aborda todas as etapas desde a conceitualização do problema até a modelagem e solução de diferentes modelos de CS para então, finalmente, se desenvolver, implantar e validar uma proposta de *framework* para a modelagem de CSMs.

1.3.3 Etapas do Método a Serem Desenvolvidas

O método para o desenvolvimento da pesquisa é integrado por múltiplas etapas. Neste sentido, a Figura 1.3 apresenta cada uma das etapas a serem seguidas na elaboração da pesquisa, que serão exploradas individualmente ao longo desta dissertação.

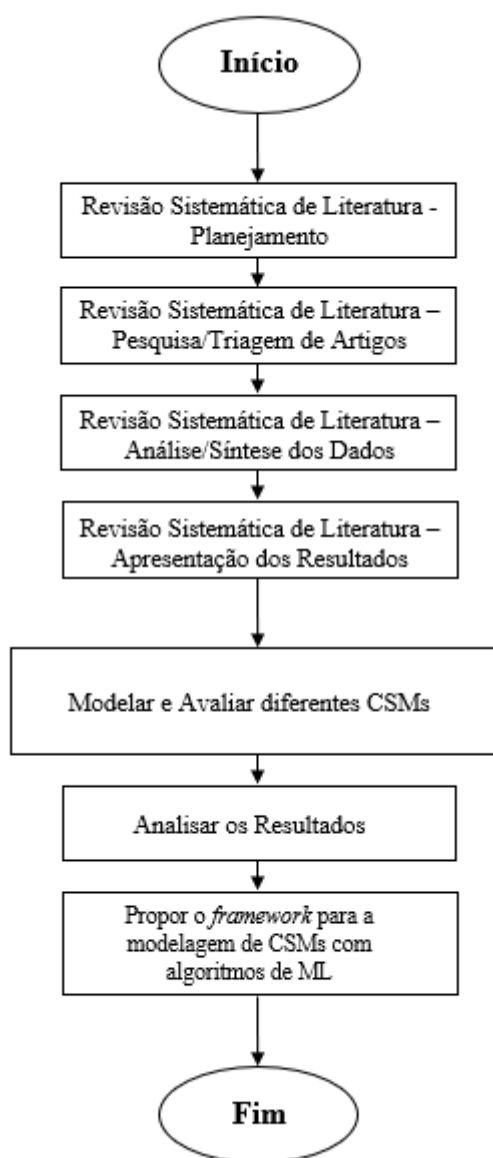


Figura 1.3 – Etapas da pesquisa
Fonte: Autoria própria (2023)

A primeira etapa do método de pesquisa adotado, de contextualização, se dará pela análise da literatura acerca do tema. Nesta etapa será conduzida uma revisão sistemática de literatura (RSL) (Capítulo 3) com o intuito de identificar os principais algoritmos de ML empregados na modelagem de CSMs, além das outras etapas que podem integrar o processo e as principais técnicas nelas empregadas. O resultado desta etapa promoverá o embasamento teórico necessário para a realização da etapa de modelagem, que possibilita a avaliação adequada da interação entre os algoritmos e demais etapas comuns ao processo de modelagem de CSMs.

Então, avaliando as métricas de performance de modelos partindo dos diferentes algoritmos e suas diferentes combinações com as demais etapas consideradas no processo de modelagem (*e.g.* como performa um CSM modelado a partir de um SVM sem FS *versus* com diferentes metodologias de FS), será proposto um *framework* para a modelagem de CSMs partindo dos algoritmos de ML, visando direcionar adequadamente este tipo de trabalho e, por fim, o *framework* será aplicado para que sua eficácia seja ou não validada.

1.4 Delimitação da Pesquisa

Nesta pesquisa será proposto um *framework* para a modelagem de CSMs, no qual serão incorporados conceitos de ML, pré-processamento de dados, *resampling*, FS e OHP. Ressalta-se que não é escopo e nem objetivo deste trabalho propor ou incrementar a formulação matemática de algoritmos de ML, e sim estabelecer qual a sequência de etapas que deve ser seguida para a otimização de seus resultados em CS, dependendo do algoritmo de classificação escolhido. Para os fins desta pesquisa, foram considerados os algoritmos mais utilizados e apropriados segundo a RSL. Mais particularmente, quanto aos classificadores, LR, DT, SVM, RF e *ensembles de boosting* (AdaBoost) e *stacking*; um *filter* (*Correlation-based* FS ou CFS) e *wrapper* (*Sequential Forward Selection* ou SFS) para FS;

1.5 Estrutura do Trabalho

Este trabalho encontra-se estruturado em 6 Capítulos. No Capítulo 2 encontra-se a fundamentação teórica na qual este trabalho se baseia, apresentando os conceitos gerais acerca de CS, algoritmos de classificação, ML, pré-processamento de dados, *resampling*, FS e OHP. Já no Capítulo 3 encontra-se a RSL sobre aplicação de ML em CS. No Capítulo 4 são descritos os modelos elaborados, o Capítulo 5 descreve a elaboração dos *frameworks* seguindo os resultados obtidos e, por fim, o Capítulo 6 traz as conclusões da pesquisa.

2. FUNDAMENTAÇÃO TEÓRICA

Neste capítulo serão apresentados os conceitos principais sobre o risco de crédito e sua administração, CSMs e algoritmos de ML. Ressalta-se que o embasamento teórico presente neste capítulo constitui a base na qual foi construída esta dissertação e que suporta os resultados e conclusões obtidos com este trabalho.

2.1 Risco de Crédito

A busca pela gestão das instituições que compõe a indústria financeira em aumentar os seus resultados, como é comum da dinâmica entre risco e recompensa, é acompanhada por um aumento de risco. Instituições financeiras enfrentam numerosos riscos e a forma com que eles são administrados é essencial na sua performance (LEO; SHARMA; MADDULETY, 2019). Alguns dos riscos tradicionais da indústria financeira são: risco de mercado, risco de crédito, risco extrapatrimonial, riscos operacionais e de tecnologia, risco de câmbio, risco de liquidez e risco de insolvência. Um desdobramento completo dos riscos financeiros e suas respectivas classificações encontra-se na Figura 2.1.

Conforme Saunders *et al.* (2006), em razão destes riscos e do papel das instituições financeiras na economia, trata-se de um assunto sujeito à atenção de órgãos reguladores. Estes órgãos exigem que uma instituição tenha capital o suficiente para lidar com grande parte dos riscos inerentes às suas atividades operacionais e as instituições ativamente os monitoram, medem e administram. O risco de crédito é tradicionalmente a maior ameaça para a indústria e, conseqüentemente, o que requer mais capital (LEO; SHARMA; MADDULETY, 2019; APOSTOLIK *et al.*, 2009).

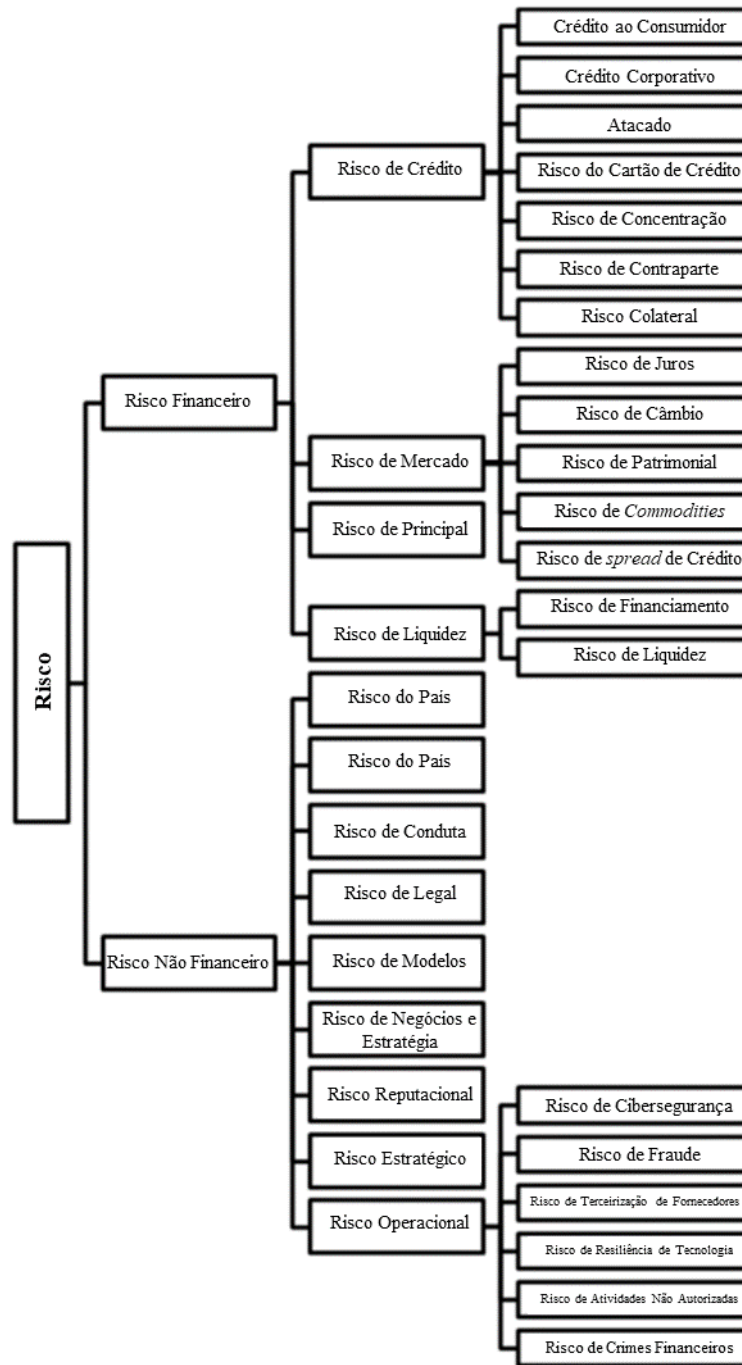


Figura 2.1 – Taxonomia dos riscos da indústria financeira
 Fonte: Adaptado de Leo, Sharma e Maddulety (2019)

O Comitê de Supervisão Bancária de Basileia define o risco de crédito como a probabilidade de um tomador ou contraparte falhar no cumprimento de suas obrigações, considerando as condições acordadas (ORLOVA, 2020). Mais especificamente, conforme Apostolik *et al.* (2009), se trata da potencial perda que um credor sofre caso um tomador falhe no cumprimento de suas obrigações contratuais e não pague o montante emprestado e os juros

acumulados. De todos os riscos financeiros, configura o mais pesquisado e avaliado, especialmente para aplicações de ML (LEO; SHARMA; MADDULETY, 2019).

2.2 Administração do Risco de Crédito

Na área de finanças, segundo Leo, Sharma e Maddulety (2019), a administração do risco de crédito continua um importante e desafiador tópico de pesquisa, sendo que os primeiros esforços se iniciaram no século passado. Em busca de ampliar as operações de crédito e, conseqüentemente, a lucratividade, sem um aumento equivalente do risco que as acompanha, o objetivo de um sistema de administração do risco de crédito é maximizar a taxa de retorno ajustada ao risco de uma instituição financeira por meio da previsão e manutenção de sua exposição ao risco dentro de parâmetros considerados aceitáveis (ORLOVA, 2020; CHOPRA; BHILARE, 2018). Se trata, essencialmente, da prática de mitigar as perdas por meio do entendimento da adequação do capital de uma instituição de crédito e a perda decorrente do risco de crédito em um dado momento.

Um dos primeiros passos na busca desta adequação de capital é a previsão da chance de tomadores não cumprirem suas obrigações contratuais e inadimplirem os empréstimos efetuados. Para isso, existe CS, utilizado por instituições financeiras para determinar a probabilidade de que um potencial devedor falhe no adimplemento de suas obrigações contratuais no processo de concessão de crédito (HUANG *et al.*, 2004). CS conta com modelos de auxílio à tomada de decisão e técnicas que informam credores antes da escolha ou não pela concessão de crédito (KHEMAKHEM; BEN SAID; BOUJELBENE, 2018).

Estes modelos e técnicas, por sua vez, encontram-se sujeitos a melhorias viabilizadas por tecnologias emergentes e requisitadas em decorrência da crescente competitividade do mercado. Isto não só colabora para grande atenção profissional acerca do assunto como estimula pesquisas para otimizar a prática de CS, processo que tem sido acelerado também pela implementação de novos acordos internacionais que regulamentam o assunto, como os acordos de Basileia, *Solvency II*, *Sarbanes-Oxley* e IFRS (MARTENS *et al.*, 2010).

2.2.1 Credit Scoring

CS é o processo de afeição do risco associado com a concessão de crédito para um cliente, que é avaliado individualmente por meio de variáveis ou atributos provindos de dados demográficos e financeiros (BOU-HAMAD, 2017; ZHOU *et al.*, 2021). O resultado geralmente

segue um sistema de pontos para a distinção da credibilidade dos tomadores, no qual perfis com maior pontuação significam maior credibilidade, correspondendo às instâncias que mais se assemelham aos perfis que anteriormente pagaram os seus respectivos créditos, conforme acordado contratualmente (PŁAWIAK; ABDAR; ACHARYA, 2019).

Existem diferentes abordagens para este processo, sobretudo: *application scoring*, *behavioral scoring* e *dynamic scoring*. *Application scoring* é a primeira importante abordagem de CS (BAESENS; RÖSCH; SCHEULE, 2015), sendo o seu propósito estimar o risco de inadimplência de um tomador no momento da solicitação da concessão de crédito e fazer com que a instituição financeira tome a decisão ou não de aceitá-la (CHOI; KOO; PARK, 2015). Já *behavioral scoring* é uma abordagem que analisa o comportamento de clientes em operações de crédito já existentes, usualmente dentro de um horizonte de tempo pré-estabelecido, na qual dados recentes de um tomador já aprovado no processo de *application scoring* são observados para reavaliá-lo. Por fim, *dynamic scoring* possibilita uma avaliação de um tomador em qualquer momento no futuro, indo além dos horizontes de tempo comumente adotados em *behavioral scoring* e permitindo, através de um processo de difícil construção, segundo Baensens, Rösch e Scheule (2015), que se modele não somente se o tomador irá tornar-se inadimplente, mas também em que momento. Este trabalho dá enfoque ao *application scoring*.

Existem várias definições para CS, porém a maior parte delas menciona a ferramenta como um método de classificação, sendo que, essencialmente, o seu propósito é dividir os clientes entre dois tipos: bons pagadores e maus pagadores (THOMAS; EDELMAN; CROOK, 2002; WANG *et al.*, 2011; ZHANG *et al.*, 2011). Tal definição, em prática, apesar de incompleta, não é incorreta, pois apesar das regulamentações vigentes exigirem a estimativa de certos parâmetros referentes ao risco de crédito, um dos primeiros passos em sua modelagem é a classificação, especialmente para *application scoring* e na determinação da probabilidade de inadimplência do tomador ou *probability of default* (PD). Em razão disto, de acordo com Bou-Hamad (2017) e Wang *et al.* (2005), geralmente se aborda o problema de CS com técnicas de classificação a partir de dados históricos do consumidor, buscando avaliar a relação entre as características dos clientes e sua possível inadimplência para classificar novas operações como de baixo ou alto risco.

Na tomada de decisão referente ao crédito para o consumidor, credores normalmente fazem o uso de abordagens intuitivas ou analíticas (MALHOTRA; MALHOTRA, 2000). As abordagens intuitivas consistem em analistas de crédito usando seu próprio conhecimento e experiências ao examinar possíveis clientes, já as abordagens estatísticas fazem o uso de métodos embasados em modelos matemáticos e estatísticos na avaliação de sua credibilidade.

Em geral, como discutido, a maior parte destes métodos podem ser divididos em abordagens estatísticas e técnicas de otimização, ou abordagens de ML.

Os primeiros CSMs foram desenvolvidos através da metodologia discriminante proposta no trabalho de Fisher (1936), que introduziu a LDA, ferramenta inovadora para problemas de classificação binária. O trabalho de Durand (1941) foi pioneiro na distinção entre bons e maus pagadores por meio de métodos estatísticos. Identificando o desafio de CS como um problema de classificação e empregando a LDA para separar potenciais tomadores entre bons e ruins, deu-se um ponto de partida na modelagem de CSMs, incentivando os futuros desenvolvimentos e modificações que resultaram nas metodologias mais complexas e empregadas no processo de concessão de crédito nos dias de hoje, sendo que, apesar de existirem diferentes formas de se abordar o problema de CS, há um enfoque nos procedimentos de classificação (FERREIRA; LOUZADA; DINIZ, 2015; AGHAEIRAD; CHEN; RIBEIRO, 2017).

Do ponto de vista do aprendizado estatístico e *data mining*, enxergando CS como um problema de classificação binária (WANG; XU; ZHOU, 2015), supondo que: (x^i, y_i) , $i \in 1, 2, \dots, n$ representa um conjunto de dados para CS, aonde $x^i = x_{i1}, x_{i2}, \dots, x_{ip}$ são variáveis preditivas ou atributos de clientes e y_i são as classes (geralmente respostas binárias, aonde 0 significa o efetivo pagamento da obrigação e 1, o contrário, sinalizando seu inadimplemento), busca-se determinar qual a probabilidade condicional $p(y | x)$ de um certo cliente pertencer a uma das classes, dada as variáveis consideradas. A avaliação na concessão de crédito ao consumidor deve levar em conta atributos como: profissão, escolaridade, situação de moradia, renda mensal, idade, estado civil, número de dependentes, contas bancárias, seguro de vida e tempo de serviço no cargo atual (PŁAWIAK; ABDAR; ACHARYA, 2019). Com base em atributos como os de cima, se estabelece um limite de corte que, que então serve de referência para recusar ou não o crédito.

CSMs eram inicialmente desenvolvidos por especialistas em processos com grande consumo de tempo e recursos (MUNKHDALAI *et al.*, 2019). Mais tarde, conforme Wang, Xu e Zhou (2015), especialmente após a popularização dos cartões de crédito ao fim da década de 60, surgiram diversas metodologias cada vez mais sofisticadas para lidar com o desafio de classificação imposto pelo CS. Além disso, o aumento significativo de crédito ao consumidor e eventos de crise financeira global, especialmente a crise das hipotecas nos Estados Unidos em 2008-2009 (NIKLIS; DOUMPOS; ZOPOUNIDIS, 2014), resultaram em um crescente enfoque em regulamentação, tendo o processo de administração do risco de crédito observado um

aumento do interesse de ambas as comunidades acadêmicas e empresariais, sendo que, por uma perspectiva de metodologia, a pesquisa na área é fortemente voltada para CS (LEO; SHARMA; MADDULETY, 2019). Assim, acentuou-se a importância dos CSMs e, em razão disto, vários métodos continuam sendo desenvolvidos para lidar com o problema corretamente (LI; CHEN, 2021; GIRI *et al.*, 2021).

Segundo Trivedi (2020), CSMs devem ser capazes de capturar comportamentos associados à probabilidade de inadimplência de um devedor no futuro. Por meio dos modelos preditivos, em *application scoring*, são então elaborados os *credit scorecards*, que desempenham o papel central em CS, para a avaliação da métrica PD. Comparando as características de tomadores em operações passadas, os *credit scorecards* dividem as novas operações em baixo e alto risco para que então seja determinado um *credit score* (CHOI; KOO; PARK, 2015). Para isso, algoritmos de classificação têm distinta importância. Enquanto LR e LDA, relativamente simples e de fácil interpretação, permanecem o padrão na indústria para a construção de *credit scorecards* (PŁAWIAK; ABDAR; ACHARYA, 2019; SHI; ZHANG; QIU, 2013; DUMITRESCU, 2022), ML, por sua vez, tem sido motivo de interesse na academia em numerosos novos estudos voltados para CS (WANG *et al.*, 2020), sendo que sua evolução tem tido grande impacto sobre as metodologias habituais, possibilitando o uso de grandes quantidades de dados históricos para a modelagem (LESSMANN *et al.*, 2015) e também gerando ótimos resultados, demonstrando habilidade de superar as técnicas clássicas e tradicionais (YU *et al.*, 2008; CHOPRA; BHILARE, 2018). Estes resultados, por sua vez, não podem ser ignorados, uma vez que melhorias em frações percentuais para um CSM podem implicar em significativas economias (BAESENS *et al.*, 2003; WEST, 2000; HUANG *et al.*, 2007).

Um grande estímulo para a digitalização de serviços, aliado ao aumento dos requerimentos de órgãos reguladores e o acúmulo de dados coletados, que é característico da era da *Big Data*, também reforça o crescente interesse pela adoção de metodologias de ML (LEO; SHARMA; MADDULETY, 2019; HURLEY; ADEBAYO, 2016). Além do desejo de aprimorar a precisão analítica dos modelos, a capacidade de automação e análise de um variado e crescente volume de dados comum a indústria financeira (KHEMAKHEM; BEN SAID; BOUJELBENE, 2018) também agem como grandes incentivos para o uso de algoritmos de *data mining* e ML no ramo (VAN LIEBERGEN, 2017; ABEDINI *et al.*, 2016). Por estas e outras razões, de acordo com Khemakhem, Ben Said e Boujelbene (2018), ML tem encontrado aplicações em diversas áreas, desempenhando papel cada vez maior no auxílio à tomada de decisão em serviços financeiros que se tornam progressivamente mais complexos.

2.2.2 Regulamentação

Desde a segunda metade do século XX, quando o sistema financeiro internacional era assegurado por certa estabilidade, lançavam-se as bases do sistema internacional de regulamentação financeira. Um dos princípios arquitetados para manter a estabilidade do sistema financeiro é o da obrigatoriedade da regulamentação de seus riscos, no qual o risco de crédito destaca-se como um dos mais significantes (ORLOVA, 2020). Com o passar dos anos, em resposta aos avanços na indústria financeira, um crescente número de falências de instituições e, em grande parte, aos períodos de crise econômica, acordos internacionais para a regulamentação surgiram e se desenvolveram para supervisionar e garantir a estabilidade da indústria. Estes acordos intensificaram o enfoque sob o risco de crédito e a modelagem a seu respeito (WU *et al.*, 2021), buscando estimular o desenvolvimento de modelos de portfólios mais apropriados e, em especial, reforçar os *frameworks* institucionais internos para a construção de CSMs (BREIMAN, 1996; HASTIE; TIBSHIRANI; FRIEDMAN, 2011). Este trabalho se enfoca sob os Acordos de Basiléia, três dos principais acordos que regulamentam a indústria financeira pelo mundo.

O Comitê de Supervisão Bancária de Basiléia representa uma instituição internacional que define as regras que governam as atividades da indústria financeira em países economicamente desenvolvidos, formando as condições básicas que caracterizam o sistema financeiro internacional.

Os acordos de Basiléia foram elaborados pelo Comitê de Supervisão Bancária de Basiléia, que foi fundado em 1974 pelos bancos centrais dos países integrantes do G10 e hoje é integrado por 27 membros que se encontram regularmente no Banco para Compensações Internacionais, a mais antiga instituição financeira internacional, em Basiléia, na Suíça (BAESENS; RÖSCH; SCHEULE, 2015). Conforme o comitê, instituições financeiras devem administrar o risco de crédito inerente aos seus portfólios, assim como considerar a sua relação com os demais riscos. Segundo Orlova (2020), o gerenciamento efetivo do risco de crédito é um componente crítico para uma abordagem compreensiva da administração de risco e essencial para o sucesso em longo prazo de qualquer instituição de crédito.

A base para as exigências dos acordos de Basiléia é o requerimento de capital o suficiente para se assegurar da estabilidade do sistema financeiro. A adequação de capital é um dos principais critérios para a estabilidade de uma instituição financeira (ORLOVA, 2020), de forma que o primeiro dos acordos, o Acordo de Basiléia I, introduzido em 1988, surgiu justamente com o objetivo de estabelecer os requerimentos para determinação do mínimo

capital regulatório, buscando assegurar que bancos sempre tivessem crédito o suficiente para devolução aos depositantes (BAESENS; RÖSCH; SCHEULE, 2015).

Para isso, o Acordo de Basiléia I introduziu a razão de Cooke, equivalente à razão entre o capital disponível e o valor dos ativos ponderados pelo risco ou *risk-weighted assets* (RWA), fixando um limite inferior de 8%, valor que, segundo Baesens, Rösch e Scheule (2015) especula-se se tratar de uma média na indústria na época do acordo. Priorizando fortemente o risco de crédito, o acordo também introduziu ponderações de risco fixas para cada classe de exposição, sendo 0% o peso para exposições em caixa, 50% para hipotecas e 100% para exposições comerciais. Desta forma, considerando-se, por exemplo, uma hipoteca de R\$ 100,00, aplicando-se o critério de ponderação equivalente de 50%, chega-se ao valor do RWA, neste caso igual a R\$ 50,00. Finalmente, para se chegar ao valor do capital requerido, é utilizada uma fórmula sobre o valor do RWA. Utilizando o valor de 8% mencionado, o capital requerido para lidar com a potencial perda desta hipoteca seria de R\$ 4,00, ou seja, a hipoteca de R\$ 100,00 deve ser financiada por ao menos R\$ 4,00 de patrimônio para cobrir o risco de crédito. Apesar de trata-se de um grande passo na direção de uma melhor gestão de riscos na indústria financeira, este acordo ainda continha alguns problemas, com destaque para a falta de atenção aos outros tipos de risco e ao risco individual do tomador, sendo que as ponderações levavam em conta exclusivamente o tipo de exposição e não o contratante ou as características do produto adquirido (BAESENS; RÖSCH; SCHEULE, 2015).

Para lidar com as falhas do primeiro acordo, o Acordo de Basiléia II foi introduzido, encorajando instituições financeiras a calcularem o seu mínimo capital regulatório. O acordo consiste em 3 pilares, conforme apresentado pela Figura 2.2.

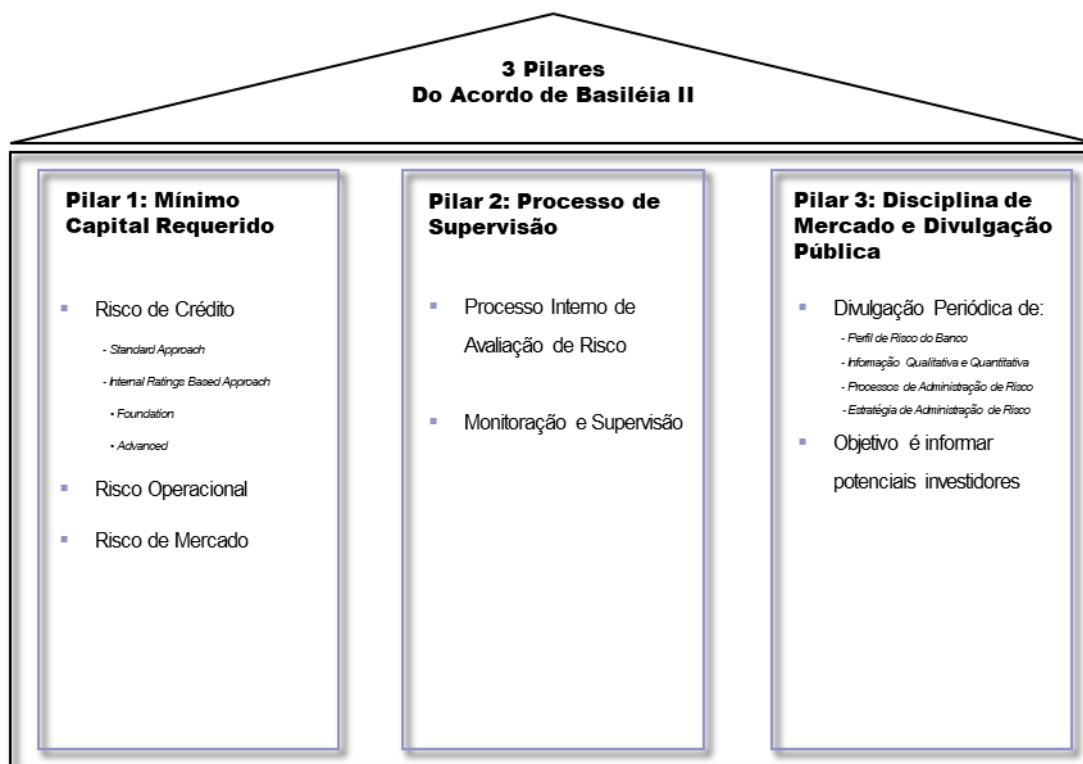


Figura 2.2 – Pilares do Acordo de Basileia II
Fonte: Adaptado de Baesens, Rösch e Scheule (2015)

Conforme os termos do Acordo de Basileia II, para a determinação do mínimo capital regulatório devem ser estabelecidos procedimentos de cálculo para os riscos de crédito, operacional e de mercado (ORLOVA, 2020). Para aferir o risco de crédito, uma instituição financeira deve então seguir uma das seguintes abordagens, sendo que todas se resumem a construção de modelos quantitativos para mensurar o risco (BAESENS; RÖSCH; SCHEULE, 2015):

- *Standardized Approach (SA)*;
- *Internal Ratings-Based Approach (IRB)*;
 - *Basic Internal Rating (Foundational IRB ou FIRB)*;
 - *Advanced Internal Ratings-Based Approach (Advanced IRB ou AIRB)*.

Enquanto a abordagem SA depende fortemente de ECAIs, sofrendo com inconsistências entre diferentes empresas, pela abordagem IRB, o Acordo de Basileia II permite que as instituições financeiras mensurem o risco de crédito independentemente. Para isso, existem três parâmetros chave e que devem ser calculados:

- *Probability of Default (PD)*: métrica que estima o percentual de devedores inadimplentes;

- *Exposure at Default* (EAD): métrica para estimar o valor total da perda para a instituição decorrente dos devedores inadimplentes;
- *Loss Given Default* (LGD): métrica utilizada para calcular o percentual de exposição ou valor monetário perdido pela instituição em decorrência da inadimplência dos devedores.

O produto destes parâmetros corresponde à perda esperada ou *Expected Loss* (EL) (Eq. 2.1) e estes são colocados em um modelo Merton/Vasicek para chegar ao valor do mínimo capital requerido (ORLOVA, 2020; MARTENS *et al.*, 2010; LEO; SHARMA; MADDULETY, 2019).

$$EL = PD \times EAD \times LGD \quad (2.1)$$

Supondo um EAD de R\$ 1.000,00, sendo LGD de 20%, isto significa que 20% de R\$ 1.000,00 serão perdidos. Sendo PD, que corresponde a probabilidade de ocorrência desta perda, igual a 1%, a EL, perda esperada pelo banco, é de R\$ 2,00. A abordagem IRB ainda se desdobra em duas subcategorias: FIRB, aonde, PD é estimada pela instituição financeira, enquanto LGD e EAD são fornecidos por um órgão regulador; e AIRB, na qual todos os parâmetros são estimados internamente. Tipicamente, não se permite a abordagem FIRB para operações de crédito ao consumidor, devendo a instituição escolher entre AS ou, preferencialmente, AIRB (BAESENS; RÖSCH; SCHEULE, 2015). Conforme Martens *et al.* (2010), ao permitir o uso de CSMs internos na determinação do mínimo capital regulatório, o Acordo de Basiléia II incentiva instituições a refinar os modelos já existentes, uma vez que previsões mais precisas implicam em menos requisição de capital. Isto configura mais um motivo para o crescente interesse pelo tópico na indústria financeira, especialmente em torno de técnicas de *data mining* e tecnologias de informação que suportem a implementação de tais *frameworks*.

Segundo Orlova (2020), como condições para a aplicação da abordagem IRB, também existem requerimentos quantitativos e qualitativos da instituição e sua gestão como um todo, inclusive sobre seus modelos internos de CS. Assim, todos os modelos desenvolvidos no Pilar 1 para o cálculo dos parâmetros requisitados precisam ser subsequentemente avaliados por supervisores (Pilar 2). Por fim, após a aprovação, os modelos são informados para o mercado (Pilar 3), sendo que bancos periodicamente apresentam informações sobre sua estratégia de administração de risco para informar investidores e convencê-los de sua eficiência gerencial, o que em troca lhes ajuda na captação de fundos (BAESENS; RÖSCH; SCHEULE, 2015).

Já o Acordo de Basiléia III, introduzido como resposta direta à crise financeira mundial, contribui com o acordo que o precedeu, trazendo, em 2010, um conjunto de novas propostas para fortalecer os padrões do capital global. Conforme Baesens, Rösch e Scheule (2015), o

ponto principal deste acordo é fortalecer os componentes com maior capacidade para absorção de perdas, diminuindo a dependência dos modelos internos ou de ECAIs. O acordo, no entanto, só teve efeito em 2013, ainda sendo implementado efetivamente e, em comparação com o Acordo de Basileia III, tendo menor impacto sobre os CSMs.

2.3 Metodologias Convencionais de *Credit Scoring*

As tarefas relacionadas a CS são atribuídas aos analistas e especialistas de crédito. Responsáveis pela determinação da habilidade de tomadores pagarem seus empréstimos a tempo, analistas e especialistas de crédito ao consumidor são muito procurados pela indústria financeira (CHOPRA; BHILARE, 2018). As diferentes formas com que eles o fazem podem ser divididas, essencialmente, em metodologias intuitivas e analíticas.

Enquanto antigamente vigoravam métodos intuitivos, no qual analistas de crédito observavam o histórico e perfil de clientes, concedendo ou não o crédito embasado por suas próprias experiências, a expansão dos mercados logo desenvolveu uma necessidade por métodos científicos ou analíticos (MALHOTRA; MALHOTRA, 2000; BOU-HAMAD, 2017). Atualmente, com a enorme competitividade do mercado e a vigência de acordos bastante exigentes quanto a administração do risco de crédito na indústria financeira, abordagens analíticas não são só mais atrativas como necessárias.

Hoje, no processo de CS, analistas tem o auxílio de *credit scores* provindos de diferentes ECAIs, enquanto especialistas e profissionais da área de *data science* e inteligência computacional estudam e desenvolvem CSMs próprios para as grandes instituições. Antes da introdução de ML ao CS, a classificação estatística tradicional foi extensivamente utilizada na modelagem de CSMs (AGHAEIRAD; CHEN; RIBEIRO, 2017), sendo que até hoje destacam-se aplicações de LR e LDA. Técnicas de ML, contudo, têm encontrado crescente influência em aplicações corporativas, demonstrando, conforme Leo, Sharma e Maddulety (2019) grande capacidade para a melhoria de performance em relação às metodologias tradicionais.

2.3.1 Abordagem Intuitiva

Tradicionalmente, na abordagem intuitiva, analistas de crédito exploram os dados de crédito de consumidores por meio de cinco perspectivas conhecidas como 5Cs (Figura 3.3). São elas: caráter do consumidor, capital, colateral, capacidade e condições econômicas (THOMAS

et al., 2002; SIAMI; GHOLAMIAN; BASIRI, 2014). No *framework* da metodologia dos 5Cs, são levados em conta, no que diz respeito ao consumidor:

- Caráter: diz respeito ao seu histórico de crédito;
- Capital: equivale, essencialmente, ao valor de seus ativos;
- Capacidade: leva em conta a razão entre suas dívidas e renda;
- Colateral: ativos que podem ser alienados para compensar sua inadimplência;
- Condições: trata dos propósitos que motivam o consumidor a buscar a concessão de crédito, o montante e taxas de juros envolvidos.

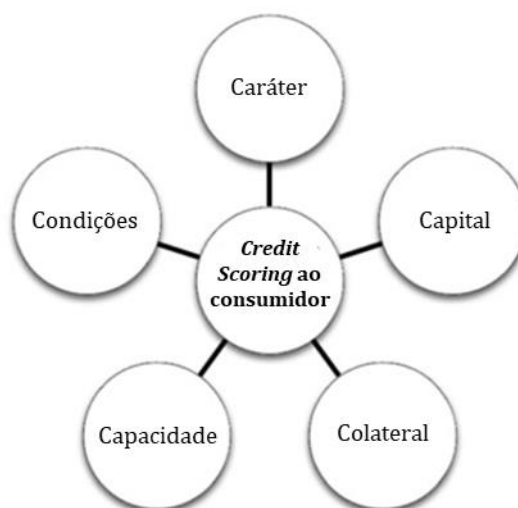


Figura 2.3 – Framework dos 5Cs na análise de crédito ao consumidor
Fonte: Adaptado de Thomas *et al.* (2002)

Conforme Zheng (2019), a afeição do risco do potencial cliente pelos analistas de crédito é feita manualmente, num processo tedioso e que consome muito tempo. Desta forma, conforme aumentam exponencialmente os dados disponíveis sobre o consumidor e o número de operações de concessão de crédito, tendo em mente os serviços bancários digitais e até mesmo plataformas de empréstimo *peer-to-peer*, a análise manual do risco de crédito torna-se inviável, dando lugar a novas metodologias quantitativas e automatizadas (WANG *et al.*, 2011; ZHENG, 2019). Além disso, apesar de considerar-se variáveis com grande correlação estatística com a capacidade de pagamento do consumidor, tratando-se de uma metodologia subjetiva e que, dependendo das experiências próprias de cada analista, resulta em inconsistências potencialmente danosas, não sendo incomum que um analista aprove um cliente que foi

recusado por outro (CROOK, 1996; ABDU; POINTON, 2011; MALHOTRA; MALHOTRA, 2000).

Em razão disso, segundo Malhotra e Malhotra (2000), abordagens não sistemáticas e inconsistentes incentivam cada vez mais o uso de sistemas de CS que diminuam o elemento da subjetividade da tomada de decisão, especialmente por meio de modelos matemáticos, estatísticos e de inteligência computacional.

2.3.2 Regressão Logística

A LR é um processo de modelagem estatística da probabilidade de uma variável discreta de saída (variável dependente ou classe) dada uma ou mais variáveis explanatórias de entrada (variáveis independentes ou atributos). Com aplicações em uma grande variedade de áreas (HONORIO; ORTIZ, 2015), inclusive no campo de ML, trata-se de uma ferramenta útil para problemas de classificação, na qual se busca determinar em que classe se encontra uma nova amostra (EDGAR; MANZ, 2017).

Como na regressão linear, a LR produz um modelo da relação entre múltiplas variáveis (SEUFERT, 2014) em um conjunto de dados. Essencialmente, usa-se uma função logística para a modelagem de uma variável binária de saída (TOLLES; MEURER, 2016), para que então seja prevista a sua probabilidade de assumir um valor no intervalo de 0 a 1.

Assim, a probabilidade da variável dependente Y ocupar determinada classe, pode ser estimada diretamente a partir de um conjunto de n variáveis independentes X_1, X_2, \dots, X_n e a LR pode ser usada tanto para a classificação quanto para a estimativa da probabilidade de classes. Por esta razão seria mais apropriado que se chamasse classificação logística (SEUFERT, 2014), uma vez que, apesar de seu nome, trata-se de um modelo de classificação e não regressão propriamente dita (SUBASI, 2020). O modelo é descrito conforme a Eq. 2.2:

$$P(Y = 1) = \frac{1}{1 + e^{-g(x)}} \quad (2.2)$$

Em que:

$$g(x) = B_0 + B_1 * X_1 + \dots + B_n * X_n \quad (2.3)$$

Sendo os coeficientes B_0, B_1, \dots, B_n da Eq. 2.3 estimados a partir do conjunto de dados na busca de uma combinação que maximize a probabilidade de a amostra ter sido observada.

Analisando o modelo nas Eqs. 2.2 e 2.3, é possível notar que quando $g(x)$ tende a $-\infty$, $P(Y=1)$ tende a 0 e quando $g(x)$ tende a $+\infty$, $P(Y=1)$ tende a 1 (HOSMER *et al.*, 1989). Desta forma, a curva de probabilidade apresenta um formato sigmoide (forma de S) (Figura 2.4).

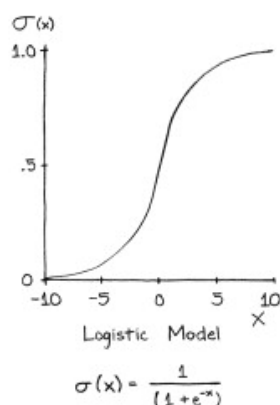


Figura 2.4 – Curva Sigmoide
Fonte: Adaptado de Seufert (2014)

Trata-se de um dos modelos estatísticos mais utilizados para algoritmos de classificação (ABDOU *et al.*, 2016). Em ML, conforme Gudivada *et al.* (2016), vetores ponderados dos atributos são mapeados na função sigmoide e interpretados como a probabilidade de uma amostra pertencer a determinada classe. O algoritmo, por sua vez, deve ajustar os coeficientes – o que pode ser feito de diferentes maneiras – para então corretamente classificar amostras de treinamento e finalmente possibilitar que o modelo seja aplicado em novas amostras.

Conforme Seufert (2014), é um método extremamente robusto e flexível para classificação preditiva dicotômica; ou seja, para a previsão de resultados ou estados binários. Em razão de sua relativa simplicidade em comparação a outros algoritmos, não fazendo suposições rigorosas (TAKAWIRA; MWAMBA, 2015), costuma ser a primeira opção para problemas de classificação (GUDIVADA *et al.*, 2016).

Além de sua maior praticidade, a LR também é altamente explicativa (BEQUÉ *et al.*, 2017), tendo significativas vantagens em termos de interpretabilidade (SHEN; WANG; SHEN, 2020), o que é comumente requerido pelas agências financeiras regulamentadoras, tornando-a *benchmark* na indústria do risco de crédito (DUMITRESCU, 2021).

2.4 Algoritmos de *Machine Learning*

A digitalização de serviços e o aumento dos requisitos regulatórios de relatórios têm levado à criação e/ou coleta de grandes quantidades de dados não estruturados com alta frequência. Uma variedade de desenvolvimentos tecnológicos contribuiu com novas formas de explorar e extrair informações de infraestruturas volumosas de dados diversos e não estruturados (LEO; SHARMA; MADDULETY, 2019). Em resposta a isso, um renovado

interesse em estatística, combinado com ciência da computação, resultou em diferentes técnicas para a melhoria de métodos de classificação e previsão (BOU-HAMAD, 2017). Portanto, o desejo de aprimorar as capacidades analíticas e a automação em várias áreas de negócios (incluindo o gerenciamento de riscos), combinado com o interesse da indústria financeira em tecnologias de inteligência artificial (IA) (VAN LIEBERGEN, 2017), despertou o interesse das instituições financeiras na adoção de soluções de ML para lidar com o aumento na quantidade de dados coletados, bem como a menor precisão das técnicas estatísticas ordinárias (SIAMI; GHOLAMIAN; BASIRI, 2014).

ML consiste em algoritmos computacionais que dependem menos de suposições sobre os dados e sua distribuição. Esses algoritmos aprendem e se aprimoram para identificar padrões, representando uma solução ideal para desafios complexos que também requerem adaptabilidade (SHALEV-SHWARTZ; BEN-DAVID, 2014). De acordo com Van Liebergen (2017), ML busca fazer previsões *out-of-sample*, aprendendo a partir das correlações passadas encontradas nas amostras, geralmente deixando de fornecer explicações detalhadas das relações analisadas, o que torna os modelos de ML mais difíceis de desenvolver e avaliar. Isso significa, por outro lado, que os modelos de ML são mais robustos e capazes de lidar com relações complexas e não lineares (o que é comumente o caso de dados de crédito), embora em detrimento de sua interpretabilidade (GALINDO; TAMAYO, 2000). Apesar destas limitações na identificação de causalidade, economistas têm adotado cada vez mais ML, em conjunto com outras ferramentas, para avaliar relações complexas (LEO; SHARMA; MADDULETY, 2019).

Em ML, algoritmos de classificação são comumente referidos como classificadores. Não é surpresa que, na literatura, tem se formado consenso de que classificadores de ML são capazes de prover modelos de maior qualidade para CS, muitas vezes proporcionando aumento significativo de desempenho em comparação com LR, que é padrão no setor (LESSMANN *et al.*, 2015). Os algoritmos de ML podem ser divididos em três categorias: supervisionados, não supervisionados ou semi-supervisionados. Considerando que a abordagem de classificação para pontuação de crédito envolve o uso de conjuntos de dados rotulados para treinar (e, portanto, supervisionar) os classificadores, a maioria dos CSMs são um ótimo exemplo de aprendizado supervisionado.

2.4.1 Modelos Individuais

Modelos de ML também podem ser exemplos de aprendizado individual ou conjunto. Quando um CSM contém um único algoritmo de ML dedicado à classificação, diz-se que o modelo é individual, enquanto modelos que combinam vários algoritmos para uma solução conjunta são um exemplo de *ensemble learning*.

Embora algoritmos de ML sejam, em geral, mais precisos do que os métodos convencionais, e CSMs *ensemble* geralmente demonstrem melhor performance do que classificadores individuais (HSIEH; HUNG, 2010; YU *et al.*, 2008), várias meta-análises e trabalhos comparativos mostram que ainda não se pode afirmar que uma específica técnica de ML tem melhor desempenho na modelagem de CSMs (BOU-HAMAD, 2017). No entanto, também se sabe que os melhores modelos são limitados por vários fatores, como estrutura dos dados, FS, escolha pontos de *cutoff*, detalhes do problema e objetivos da classificação (HAND; HENLEY, 1997; YU; WANG; LAI, 2008).

2.4.1.1 Árvores de Decisão

Uma das técnicas mais utilizadas no campo de *data mining* e na criação de classificadores é a do algoritmo das Árvores de Decisão ou *Decision Trees* (DT) (KUMAR; VERMA, 2012). DT trata de um poderoso método empregado frequentemente nas mais diversas áreas, como ML, processamento de imagens e identificação de padrões (STEIN *et al.*, 2005).

Trata-se de um modelo que reúne uma série de testes básicos de forma eficiente e coesa para comparar atributos numéricos com um valor limiar em cada teste (DAMANIK *et al.*, 2019). Normalmente, as regras conceituais são de elaboração muito mais fácil do que os valores numéricos das conexões entre os nós ou *nodes* (BARROS *et al.*, 2012; GUPTA, 2014). A estrutura de uma DT encontra-se na Figura 2.5.

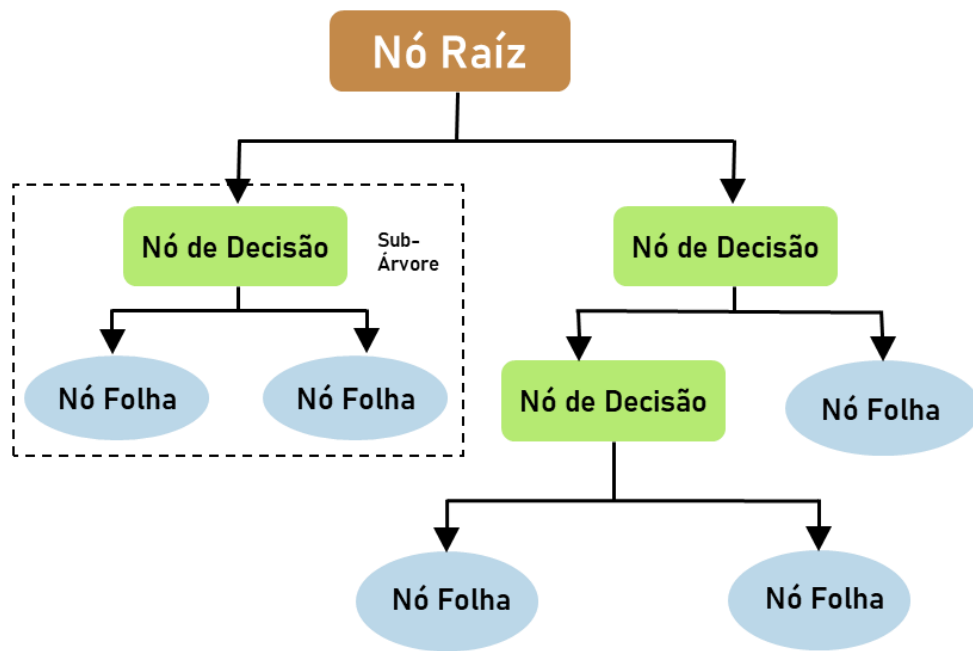


Figura 2.5 – Exemplo de uma DT
Fonte: Adaptado de Janikow (1998)

DT é comumente utilizado em modelos de classificação em *data mining* (GAVANKAR; SAWARKAR, 2017). Cada nó representa um atributo em uma categoria que será classificada e os nós de decisão se dividem de acordo com os valores possíveis (SWAIN; HAUSLA, 1977; DEY, 2016). Pela sua simplicidade e precisão, DT encontra implementação em campos distintos. Na Figura 3.6 encontra-se um exemplo de uma DT aplicada ao problema de CS.

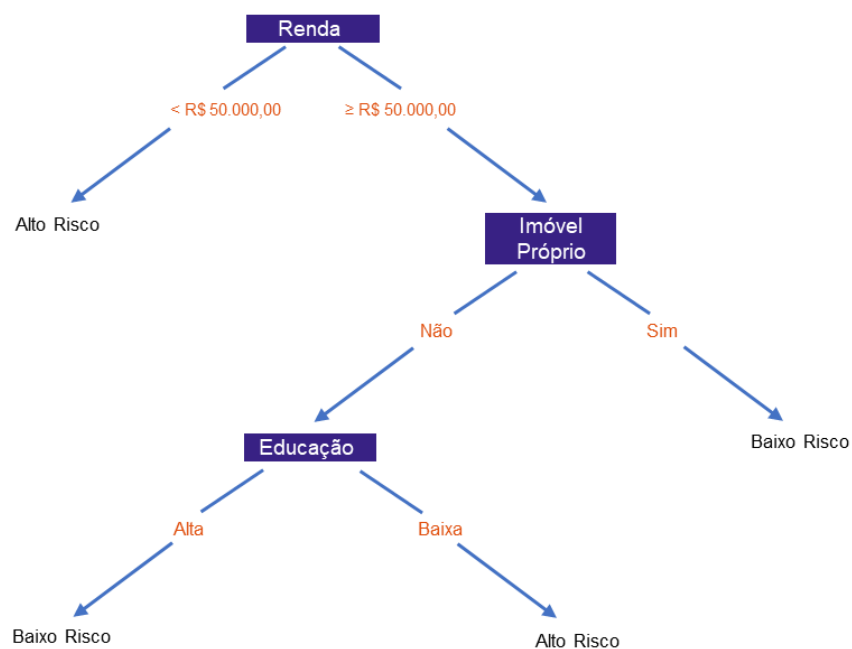


Figura 2.6 – DT aplicada em contexto de CS
Fonte: Adaptado de Teles *et al.* (2019)

2.4.1.2 Support Vector Machines

Proposto por Vapnik (1999), o SVM é um dos métodos de ML mais populares dos últimos anos (CAO *et al.*, 2020). Desenvolvido inicialmente para problemas de classificação, o método busca o limite que separa os dados de treinamento em suas respectivas classes com o mínimo de erro possível. Para isso, é definido o hiperplano com base em vetores construídos pelas observações que estão localizadas nos limites e os dados que se encontram em lados distintos do hiperplano são alocados em classes diferentes (ØSTERGÅRD *et al.*, 2018).

Conforme Jeng (2006), o SVM tenta encontrar a margem máxima entre as duas classes por meio da combinação linear dos vetores, solucionando o problema por meio de uma programação quadrática com restrições lineares. Com a introdução do parâmetro \mathcal{E} , que representa a função de perda, este algoritmo também pode ser estendido à solução de problemas não lineares. A formulação matemática para o SVM, conforme Li *et al.* (2010), encontra-se disposta na Eq. 2.4:

$$\hat{y} = \hat{f}(X) = w\phi(X) + b \quad (2.4)$$

Sendo $X \in \mathfrak{R}^k$ o input k-dimensional, ϕ a transformação não-linear de \mathfrak{R}^k para \mathfrak{R}^h , com $h > k$, e $\hat{f}(X)$ o hiperplano em \mathfrak{R}^h . Já w e b são coeficientes obtidos pela minimização da função de risco descrita pela Eq. 2.5.

$$R_{svm}(C) = C \sum_{i=1}^n C(\hat{f}(X_i), y_i) + \frac{1}{2} \|w\|^2 \quad (2.5)$$

Sendo:

$$C(\hat{f}(X_i), y_i) = \begin{cases} |\hat{f}(X) - y| - \mathcal{E} & \text{se } |\hat{f}(X) - y| \geq \mathcal{E} \\ 0 & \text{Caso Contrário} \end{cases} \quad (2.6)$$

Solucionado o problema de minimização, a função de previsão final do método SVM é obtida pela Eq. 2.7 (VAPNIK, 1999).

$$f(X, a_i, a_i^*) = \sum_{i=1}^n (a_i - a_i^*)K(X, X_i) + b \quad (2.7)$$

No qual X é o novo input do modelo, $K(X, X_i)$ é a função que representa a distância entre X e X_i e a_i, a_i^* e b são os coeficientes obtidos pela minimização da função de risco (LI *et al.* 2010).

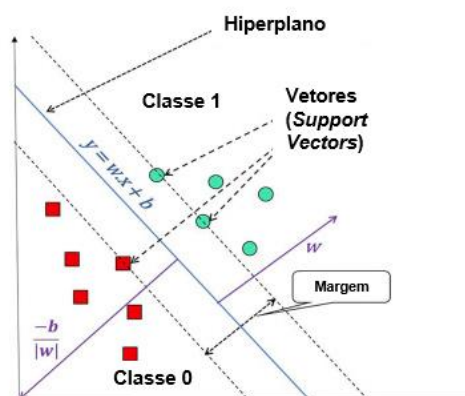


Figura 2.7 – Ilustração do SVM
 Fonte: Adaptado de Rani *et al.* (2002)

2.4.2 Ensemble Learning

Ensemble learning, ou o aprendizado conjunto, é um tipo de paradigma de ML em que, de formas distintas, são combinados múltiplos modelos, chamados aqui de *base learners*, construídos para o desenvolvimento de uma abordagem híbrida na solução de um problema (POLIKAR, 2006; SAIA; CARTA, 2016; FENU, 2018). Segundo Strobl *et al.* (2008), constitui uma popular ferramenta preditiva para lidar com estruturas de dados complexas com alta precisão, sabendo que, conforme Dietterich (1997), na maior parte dos casos, modelos *ensemble* produzem previsões com performance superior aos *base learners*.

Em *ensemble learning*, as decisões de múltiplos classificadores diferentes são combinadas para formar uma saída agregada final (FLOREZ-LOPEZ; RAMON-JERONIMO, 2015), e quanto mais diversos e precisos forem esses classificadores, mais robusto pode ser o modelo *ensemble*, alcançando resultados superiores ao qualquer um de seus membros individuais, o que se encontra demonstrado na literatura (BREIMAN, 1996; DIETTERICH, 1997). Em razão do sucesso de metodologias *ensemble* em diferentes problemas de classificação, estes algoritmos logo encontraram aplicações em CS, sendo diversos modelos de *ensemble learning* e com diferentes *base learners* propostos recentemente com este intuito (WANG; XU; ZHOU, 2015).

O trabalho de Paass e Kindermann (1998) foi pioneiro na aplicação de algoritmos *ensemble* no contexto de CS. Apesar desta abordagem ser inicialmente pouco utilizada na área de gestão de riscos financeiros (AGHAEIRAD; CHEN; RIBEIRO, 2017), desde então, pesquisadores têm experimentado diferentes combinações de modelos na criação de CSMs com sucesso, tendo os resultados apontado para melhorias na precisão dos modelos agregados em suas respectivas aplicações (LI; CHEN, 2022; CHOPRA; BHILARE, 2018). Dumitrescu *et al.*

(2022), por exemplo, reconhece a superioridade dos modelos *ensemble* em relação a tradicional abordagem estatística da LR, afirmando que estes lidam melhor com efeitos não lineares.

Desta forma, diversos pesquisadores e profissionais encorajam o uso de modelos *ensemble* sofisticados e, nos dias de hoje, a combinação de algoritmos de ML tem recebido atenção cada vez maior em CS (DUMITRESCU *et al.*, 2022; AGHAEIRAD; CHEN; RIBEIRO, 2017). Contudo, por diversas razões, destacando-se sua difícil interpretação e modelagem, trata-se ainda de modelos novos e com pequena aplicação prática.

Modelos *ensemble* podem ser homogêneos ou heterogêneos, dependendo se há ou não diferentes *base learners* envolvidos (LESSMANN *et al.*, 2015). Um conjunto homogêneo consiste em diferentes modelos do mesmo algoritmo de classificação. Por exemplo, o *Random Forest* (RF), como veremos, é um conjunto homogêneo muito popular de DTs. Por outro lado, um modelo de conjunto heterogêneo é composto por diferentes classificadores, buscando melhorar os resultados ao ter bons modelos compensando o desempenho inferior de outros classificadores distintos também agregados ao conjunto (SANTANA *et al.*, 2010).

Existem também diferentes métodos de agregação de modelos. Os *base learners* podem ser efetivamente combinados por meio de *bagging*, *boosting* ou *stacking*. A técnica de *bagging*, alcunha para *bootstrap aggregation* (YETURU, 2020), foi proposta por Breiman (1996), e representa o processo de treinamento de um modelo com diferentes subconjuntos de dados.

O algoritmo de *bagging* busca otimizar um modelo criando uma sequência de múltiplos classificadores. Considerando que cada um dos classificadores é treinado com um subconjunto diferente de dados, a previsão final é feita por uma regra de votação que seleciona os melhores modelos de acordo com os resultados variados (SAWARN *et al.*, 2020). Por exemplo, para a classificação do risco de crédito baseada no input x , gerar K subconjuntos (obtidos a partir dos dados de treinamento) possibilita que um algoritmo de *bagging* treine separadamente diferentes modelos $y_k(x)$, em que $k = 1, 2, \dots, K$ (LIN; HU; TSAI, 2012). A resposta final é dada então pela Eq. 2.8:

$$y(x) = \frac{1}{K} \sum_{k=1}^K y_k(x) \quad (2.8)$$

Já algoritmos de *boosting*, segundo Misra e Li (2020), buscam melhor desempenho preditivo de forma distinta, combinando classificadores fracos em um modelo mais robusto. Na agregação por meio de algoritmos de *boosting*, todo o conjunto de dados é ajustado a um modelo e treinado iterativamente, de forma com que cada novo modelo tente compensar as fraquezas do anterior.

Cada um dos classificadores mais fracos $y_k(x)$, no qual $k = 1, 2, \dots, K$, é treinado utilizando ponderações dos dados originais. Em cada iteração, o algoritmo de *boosting* ajusta o peso dos coeficientes de maneira com que as amostras classificadas incorretamente recebam maior peso, subseqüentemente influenciando mais o classificador seguinte. Após K iterações, um modelo *ensemble* robusto é obtido conforme exposto pela Eq. 2.9:

$$y(x) = \text{sign} \left(\sum_{k=1}^K \alpha_k y_k(x) \right) \quad (2.9)$$

Sendo α_k a medida ponderada para a precisão do classificador mais fraco (FREUND; SCHAPIRE, 1996). Existem diferentes tipos de algoritmo de *boosting*, como *Adaptive Boosting* (AdaBoost), *Gradient Boosting* (GBM), e *Extreme Gradient Boosting* (XGBoost). Contudo, em linhas gerais, todos eles demonstram a abordagem descrita, ilustrada pela Figura 2.8:

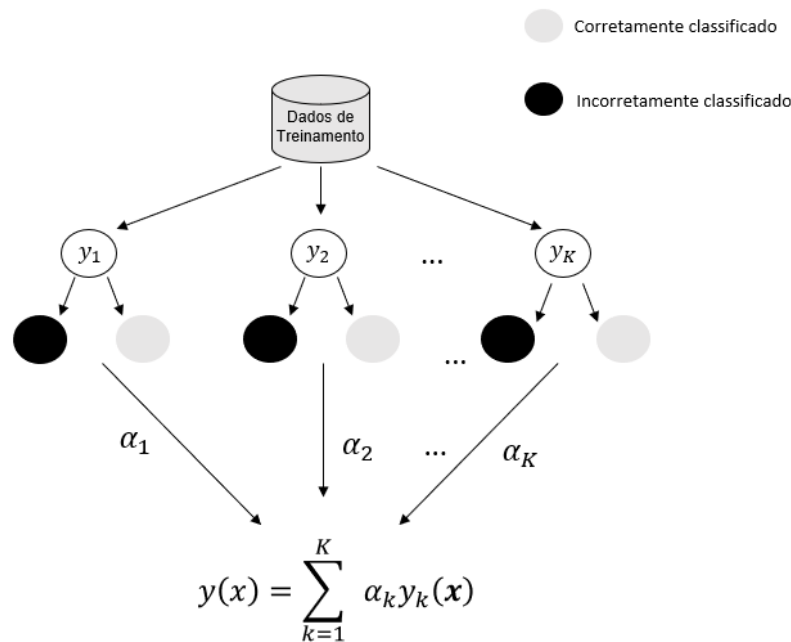


Figura 2.8 – Abordagem de algoritmos de *boosting*
Fonte: Autoria própria (2023)

Ambas as abordagens de *boosting* e *bagging* são comumente encontradas em *ensembles* homogêneos. Suas principais diferenças quanto suas principais características, por sua vez, encontram-se no Quadro 3.1:

Quadro 2.1 – Comparação entre algoritmos de agregação *bagging* e *boosting*

Aspecto	<i>Bagging</i>	<i>Boosting</i>
Dados de treinamento	Múltiplos subconjuntos	Totalidade dos dados
Complexidade dos <i>base learners</i>	Complexos	Fracos
Redução de variância	Sim	Não
Redução de <i>bias</i>	Não	Sim
Controle de complexidade	Próprios <i>base learners</i>	Tamanho do <i>ensemble</i>

Fonte: Autoria própria (2023)

Por fim, o método de agregação *stacking* foi introduzido nos trabalhos de Wolpert (1992), e consiste em fazer com que o modelo final aprenda com a previsão de múltiplos *base learners* para o mesmo conjunto de dados. Dado que alguns classificadores podem rotular corretamente uma amostra que é classificada incorretamente por outros, o *stacking* consiste em um algoritmo treinado para aprender com as diversas metodologias aplicadas nos distintos classificadores individuais, sendo por isso mais usado em *ensembles* heterogêneos. Além disso, modelos heterogêneos também podem ser combinados, resultando em classificadores híbridos.

2.4.2.1 *Random Forest*

No estudo de Breiman (2001), foi apresentado o método RF, que consiste em combinar um grande número de DTs de forma aleatória e não-correlacionada. O objetivo é obter uma previsão calculando a média dos votos de todas as DTs (LI *et al.*, 2020). O RF é um dos algoritmos de ML mais precisos e rápidos quando aplicado a conjuntos de dados extensos. Além disso, conforme Crisci *et al.* (2012), permite uma interpretação fácil das variáveis e dos resultados.

Trata-se de um algoritmo *ensemble* extremamente versátil e popular que, em essência, não passa de uma agregação de DTs por meio de *bagging*. Segundo Dai *et al.* (2018), é realizada uma combinação das DTs $\{h(x, \theta_k), k = 1, 2, \dots\}$, no qual cada DT recebe um vetor aleatório $\{\theta_k\}$. Para o vetor X e *output* Y o erro médio quadrático associado a cada DT $h(x)$ é definido pela Eq. 2.10 (DAI *et al.*, 2018).

$$E_{X,Y}(Y - h(X))^2 \quad (2.10)$$

O procedimento para ajustar a RF é simples, o primeiro passo é ajustar todas as m DTs independentes, em seguida, a cada interação é obtida uma amostra de tamanho N e selecionado aleatoriamente k atributos a fim de gerar novas DTs de forma não correlacionadas (DE LA

FUENTE; SMITH, 2017). As previsões individuais de cada m -ésima DT são guiadas por um conjunto de regras *if-then-else*, caracterizadas por Θ (HASTIE *et al.*, 2008). Para Breiman (2001), a previsão da RF é calculada pela média das k DTs $h(x)$ pertencentes a RF, sendo dada pela Eq. 2.11, onde M é o número de DTs e $T(X; \theta_m)$ o modelo que representa uma única DT, com X o vetor de dados de treino e θ_m o conjunto de regras destas DTs.

$$f(X) = \frac{1}{M} \sum_{m=1}^M T(X; \theta_m) \quad (2.11)$$

2.4.2.2 Stacking

Stacking constitui um modelo *ensemble* de ML caracterizado também pela combinação de múltiplos outros algoritmos. Proposto por Wolpert (1992), ao contrário dos algoritmos de *boosting* e *bagging*, combinam-se algoritmos distintos e treinados com os mesmos dados.

Considerando os diferentes *base learners* L_1, \dots, L_n treinados numa base de dados S , que consiste em amostras $s_i = (x_i, y_i)$, nos quais x_i são os vetores dos atributos e y_i corresponde à classe, é gerado um conjunto de treinamento para o meta-algoritmo generalizado. O intuito é o de que caso certos modelos classifiquem corretamente uma amostra e outros não, o treinamento com esta diversidade de metodologias pode possibilitar que o modelo final aprenda com as tentativas de previsão dos algoritmos de base.

2.5 Balanceamento de Classes

Uma das maiores adversidades dos CSMs e de grandes conjuntos de dados para classificação em geral é o desequilíbrio entre classes. Não surpreendentemente, em grandes conjuntos de dados de crédito reais de instituições de empréstimo para consumidores, o número de clientes não inadimplentes é consideravelmente maior do que o número de inadimplentes (isto é, ao menos, um dos maiores objetivos da gestão do risco de crédito). Este desequilíbrio entre as classes frequentemente interfere negativamente nos resultados da classificação (CHEN, 2021).

Conjuntos de dados desbalanceados correm o risco de causar *overfitting*, quando a previsão de um modelo se inclina em direção à classe majoritária (ONEL, 2018). O tratamento do desequilíbrio de classes pode ser feito de diferentes maneiras. Shajalal, Petr Hajek e Abedin (2023) discutem vários métodos para lidar com o desequilíbrio de classes. As soluções podem, principalmente, ocorrer no nível dos dados (*data-level*), fazendo nova amostragem dos dados

para reduzir a proporção de desequilíbrio, o que é chamado de *resampling*, ou dividindo o conjunto de dados em subconjuntos balanceados (HAMDOUN; RGUIBI, 2019; SARIEV; GERMANO, 2020).

Li *et al.* (2021) e Boz *et al.* (2018) também abordam o desequilíbrio de classes por meio do próprio CSM, no nível do algoritmo, por meio da aprendizagem sensível ao custo ou *cost-sensitive learning*. Em um algoritmo de aprendizagem sensível ao custo, são considerados também os custos da classificação incorreta. Consequentemente, um CSM sensível ao custo deve priorizar a redução das taxas do Erro Tipo II (conhecidas no setor como taxas de risco de crédito), quando amostras de consumidores inadimplentes são classificadas incorretamente como não inadimplentes (CHEN *et al.*, 2009). Enquanto o Erro Tipo I (risco comercial) leva à perda de negócios ao classificar incorretamente os consumidores não inadimplentes, o Erro Tipo II envolve perda de capital, expondo o banco a um risco muito maior (SIAMI; GHOLAMIAN; BASIRI, 2014).

No nível dos dados, técnicas e algoritmos de *resampling* significam lidar com o desequilíbrio de classes antes da etapa de classificação, reduzindo instâncias da classe majoritária, aumentando instâncias pertencentes à classe minoritária ou ambos. Essas abordagens são conhecidas, respectivamente, como *undersampling*, *oversampling*, ou métodos híbridos. A etapa de *resampling* pode ser feita aleatoriamente (por meio de *Random Oversampling* - ROS ou *Random Undersampling* - RUS), adicionando ou removendo instâncias sem nenhum critério, ou por meio de algoritmos capazes de fazê-lo através da criação de novas amostras sintéticas, ou que eliminam amostras entendidas como de menor importância. A maioria desses algoritmos funciona analisando a classe de uma amostra e seus vizinhos mais próximos.

Marqués, García e Sánchez (2013a) conduziram pesquisas interessantes neste sentido, avaliando diferentes técnicas de *resampling* e seus impactos no desempenho de classificadores. Assim como no trabalho de Papouskova e Hajek (2019) e Khemaken, Ben Said e Boujelbene (2018), os autores concluíram que o balanceamento dos conjuntos de dados pode melhorar significativamente o desempenho preditivo, adicionando também que, na maioria dos casos, técnicas de *oversampling* obtiveram melhores resultados do que as de *undersampling*.

2.5.1 SMOTE E Tomek's links

Um método algorítmico muito popular de *oversampling* é a *Synthetic Minority Oversampling Technique* (SMOTE). SMOTE trata de um algoritmo utilizado para incrementar a classe minoritária de uma base de dados desbalanceada.

Este método estatístico seleciona amostras de cada classe e seus vizinhos mais próximos para, então, criar amostras sintéticas – e não cópias – para a classe minoritária (KHEMAKHEM; BEN SAID; BOUJELBENE, 2018) por meio da combinação dos atributos de suas amostras selecionadas e os seus pontos vizinhos (GHORBANI; GHOUSSI, 2020). Vale apontar que, assim, a variedade dentro das amostras é mantida, ao contrário de ROS.

Já o algoritmo Tomek's links, empregado como forma de *undersampling*, busca eliminar amostras da classe majoritária. Um *tomek link* refere-se ocorrência de amostras de diferentes classes que, contudo, são vizinhas mais próximas. Por meio deste algoritmo, portanto, se identificam as instâncias em que isto ocorre para, então, se eliminar a amostra pertencente à classe majoritária (BACH; WERNER; PALT, 2019).

2.5.2 One-class classification

Em CS, quando há pouca informação histórica sobre consumidores inadimplentes, mais difícil é para instituições de crédito estimarem confiavelmente a métrica PD. Este caso é conhecido como *low default portfolio* (LDP). Apesar desta pesquisa ter como enfoque a abordagem binária de classificação e trabalhar com algoritmos de ML supervisionado (restando lidar com LDPs por meio de técnicas de *resampling*), uma solução interessante quando o desequilíbrio entre as classes deteriora são as metodologias de *one-class classification* (OCC).

Algoritmos de OCC constituem um ramo de ML utilizado para identificar objetos ou instâncias pertencentes a uma única classe. Quando o desbalanceamento de classes é tão severo que discriminar entre classes torna-se um desafio ainda maior, uma possível alternativa é OCC, existindo uma variedade de técnicas (especialmente de aprendizado não supervisionado ou semi-supervisionado), tais como detecção de anomalias, *clustering* ou *deep learning*.

Quanto ao OCC no contexto de CS, Kennedy, Namee e Delany (2013) analisaram a adequação de algoritmos semi-supervisionados de OCC como solução alternativa para LDPs, concluindo, por mim, que somente na ausência quase completa de histórico de inadimplência é que devem ser usados algoritmos de OCC no lugar dos modelos supervisionados para classificação binária.

2.6 Otimização de Hiper Parâmetros

De acordo com Wu *et al.* (2019), algoritmos de ML comumente envolvem problemas de otimização. A maioria destes algoritmos possui numerosos hiper parâmetros que devem ser ajustados e definidos antes do treinamento do modelo. Diferentes hiper parâmetros podem afetar significativamente a forma como um modelo se ajusta aos dados e, conseqüentemente, o desempenho resultante da classificação.

Na literatura, ajustar adequadamente hiper parâmetros ainda é um trabalho em andamento (ZHENG, 2019). Essa tarefa pode ser realizada manualmente ou por meio de buscas automáticas. Embora seja possível ajustar manualmente os parâmetros com base em *benchmarks* ou na própria experiência do modelador, métodos de busca automática contam com algoritmos de OHP.

De acordo com Feurer e Hutter (2019), as técnicas de OHP surgiram na década de 1990, quando ficou estabelecido que diferentes configurações de hiper parâmetros funcionavam melhor para conjuntos de dados diferentes. Atualmente, é amplamente aceito que hiper parâmetros ajustados melhoram o desempenho em relação às configurações padrão fornecidas pelas bibliotecas comuns de ML, geralmente ao custo de modelos mais complexos.

Um exemplo simples de ajuste automatizado de parâmetros é o algoritmo *grid search*. Trata-se de um algoritmo fácil de se entender e implementar (DUFOUR; NEVES, 2019). Essencialmente, *grid search* funciona treinando o modelo de ML com diferentes combinações possíveis de hiper parâmetros e, em seguida, encontrando a melhor combinação entre as tentativas por meio da avaliação dos resultados do modelo de acordo com uma métrica de desempenho pré-definida (WU *et al.*, 2019). Uma simples *grid Search*, com algumas combinações de hiper parâmetros, pode já trazer resultados satisfatórios. Algoritmos de OHP, portanto, atraem substancial interesse comercial, considerando sua capacidade de aperfeiçoamento da performance do modelo com um custo reduzido de esforço humano, além de também aumentarem a reprodutibilidade de experimentos e a imparcialidade de estudos científicos (FEURER; HUTTER, 2019). Zheng (2019) acrescenta que uma função de desempenho do modelo estimada com base em experiências históricas de treinamento pode reduzir o tempo de OHP, o que deve ser explorado mais a fundo.

2.7 Feature Selection

Na era do *Big Data*, as instituições de crédito possuem um excesso de informações sobre seus clientes, e utilizá-las em sua totalidade no treinamento de um CSM pode ser extremamente ineficiente. Não apenas um grande volume de dados é um problema em si, mas dados irrelevantes também podem causar dificuldades computacionais no processo de modelagem, exigindo maior esforço e investimento (ZHOU *et al.*, 2021; HU *et al.*, 2018).

Para treinar eficientemente um modelo que lida com uma quantidade excessiva de dados ou dados irrelevantes, existem várias metodologias para redução da dimensionalidade que envolvem a filtragem ou remoção de variáveis (ALA'RAJ; ABBOD, 2016). Consequentemente, as metodologias desenvolvidas para abordar esse problema visam avaliar e priorizar os fatores que têm maior impacto na variável dependente. No campo de *data mining*, esse processo é conhecido como FS ou seleção de variáveis, e tem desempenhado um papel cada vez maior nos setores de gerenciamento de risco da indústria financeira (WANG; XU; ZHOU, 2015; ZHOU *et al.*, 2021; LI; CHEN, 2022).

Nem todos os atributos em um conjunto de dados têm a mesma importância. Desta forma, selecionar adequadamente os atributos com maior impacto no resultado da classificação é uma parte importante da modelagem de risco de crédito (BOU-HAMAD, 2017). FS desempenha, portanto, um papel valioso na busca por combinações ótimas de atributos para a redução de dimensionalidade (ZHOU *et al.*, 2021).

Metodologias de FS se dividem em três categorias: métodos de filtro ou *filter*, métodos *wrapper* e métodos intrínsecos ou *embedded* (WAH *et al.*, 2018). Esses métodos diferem em termos de sua integração com o classificador, medidas de importância, complexidade computacional e capacidade de detectar redundâncias e interações entre características (PUDJIHARTONO, 2022).

Primeiramente, métodos *filter* classificam a relevância de cada característica individualmente, removendo aquelas com menor pontuação. Isto pode ser feito por meio de diferentes abordagens, como testes estatísticos, medidas de informação ou distância (LIU *et al.*, 2018). Estes métodos são mais rápidos, escaláveis e independentes do classificador, podendo se dividir ainda em abordagens univariadas ou multivariadas (YONGJUN *et al.*, 2012; YUSTA, 2009).

Por serem independentes do classificador, métodos *filter* também se encontram livres do viés do algoritmo de classificação, reduzindo *overfitting* (quando um modelo se ajusta demasiadamente aos dados de treinamento, mas é incapaz de classificar novos dados

precisamente). Isso também significa, no entanto, que a interação entre a seleção de variáveis e os algoritmos de classificação não é considerada, generalizando o conjunto de variáveis restantes, que não é ajustado especificamente para nenhum classificador em particular (ZHANG *et al.*, 2013). Um popular *filter* para FS é o método CFS, onde escolhem-se os atributos de maior correlação com a variável dependente.

Métodos *wrapper*, por sua vez, funcionam de maneira semelhante aos *filters*, com a exceção de que usam um algoritmo de classificação predefinido ao invés de uma medida independente para avaliação do subconjunto. Em razão disso, frequentemente *wrappers* resultam em melhor performance, muitas vezes às custas de maior intensidade computacional (WAH *et al.*, 2018).

Nestas abordagens, um algoritmo de classificação específico é treinado e avaliado usando diferentes subconjuntos (PUDJIHARTONO, 2022). Assim, *wrappers* utilizam o desempenho do classificador escolhido para identificar o subconjunto que lhe oferece melhor desempenho, o que não necessariamente significa que o mesmo subconjunto será ótimo para outros classificadores. Dentre eles encontram-se Algoritmo Genético ou *Genetic Algorithm* (GA), *Recursive Feature Elimination* (RFE) e os métodos sequenciais, especialmente *Sequential Backward Selection* (SBS) e SFS, que são métodos iterativos onde, respectivamente, se removem e adicionam atributos de forma progressiva.

Por fim, os métodos intrínsecos ou *embedded* também envolvem algoritmos de classificação, integrando o processo de FS e o de aprendizado (LIU *et al.*, 2018). Neste caso, durante a fase de treinamento, ajusta-se os parâmetros do classificador para que sejam determinados pesos apropriados para cada característica, a fim de obter os melhores resultados. Combinando as qualidades dos métodos *filter* e *wrapper*, métodos intrínsecos são uma solução intermediária para ambos (GUO *et al.*, 2019). Um exemplo popular de método intrínseco é o algoritmo *Least Absolute Shrinkage and Selection Operator* (LASSO). Combinações de diferentes tipos de métodos podem resultar ainda em métodos híbridos ou combinados.

2.8 Interpretabilidade

Em ambientes de reais de concessão de crédito ao consumidor, a compreensão e explicabilidade de um CSM são fundamentais, muitas vezes sendo obrigatórias (FLOREZ-LOPEZ; RAMON-JERONIMO, 2015) e requeridas nos Pilares 2 e 3 do Acordo de Basileia II. Além de detectar os clientes inadimplentes entre os tomadores, as instituições financeiras também têm interesse em determinar quais características dos solicitantes são mais

significativas (WANG; XU; ZHOU, 2015). Portanto, determinar quais variáveis em um modelo são melhores preditoras é de vital importância para sua interpretabilidade.

A interpretabilidade do modelo está se tornando cada vez mais importante, levando ao desenvolvimento do campo de *explainable AI* (XAI). Ferramentas de XAI adquiriram grande importância à medida que os modelos de ML começaram a ser empregados em aplicações mais críticas. De acordo com Bou-Hamad (2017), alguns praticantes e analistas de risco de crédito consideram, inclusive, que entender a relação entre os preditores é ainda mais importante do que a performance da previsão. A interpretabilidade é, portanto, fundamental em CS, e as instituições financeiras devem ser capazes de justificar os motivos para a recusa de crédito (FLOREZ-LOPEZ; RAMON-JERONIMO, 2015).

Alguns algoritmos de classificação mais simples, como DTs, regressão linear, LDA e LR são intrinsecamente interpretáveis (destacam-se os modelos estatísticos), pois seu funcionamento interno é relativamente direto (reiterando ser este um dos principais motivos pelo qual LR continua sendo uma referência na indústria). Infelizmente, esse não é o caso dos modelos de ML complexos que frequentemente apresentam melhores resultados. Leo, Sharma e Maddulety (2019), assim como Dumitrescu *et al.* (2021), argumentam que uma das maiores desvantagens do ML é justamente sua difícil interpretação.

Esta é uma das principais razões pelas quais muitas abordagens avançadas não foram adotadas e implementadas, considerando que sua falta de interpretabilidade as torna não prontamente transferíveis do domínio acadêmico para a aplicações práticas na indústria financeira (MARTENS *et al.*, 2010). Portanto, em termos de desempenho, a rigorosa demanda regulatória por interpretabilidade de modelos no caso de CSMs pode ser um problema, considerando o inevitável *trade-off* entre performance e interpretabilidade observado na maioria dos casos. Este desafio levou ao desenvolvimento de diferentes métodos para interpretar modelos de ML complexos.

Diferentes autores reconhecem a necessidade de equilibrar desempenho e interpretabilidade, buscando extrair regras compreensíveis de classificadores mais robustos. Considerando o *ensemble learning*, essa necessidade é ainda maior, dada a potencial vantagem que seu desempenho preditivo pode representar para a indústria como um todo (HSIEH; HUNG, 2010). Medidas de importância de variáveis, como aquelas discutidas no Subcapítulo 2.7, ao tratar de FS, podem ser uma ferramenta útil para a interpretação do modelo. No entanto, alguns autores criticam essas propostas, argumentando que estimar a importância das variáveis é o suficiente para o entendimento adequado dos atributos (BREIMAN, 2001; DE BOCK; VAN DEN POEL, 2012; KRUPPA *et al.*, 2013).

A interpretabilidade de um modelo também deve permitir a integração entre o modelo quantitativo e a expertise. Se especialistas decidirem que uma regra não é totalmente adequada, eles devem ser capazes de refiná-la de acordo com a experiência (TOMCZAK; ZIĘBA, 2015). Nesse sentido, agora existem abordagens agnósticas, ou seja, independentes do modelo, projetadas para funcionar com qualquer modelo de ML. Técnicas como *Shapley Additive Explanations* (SHAP), *Local Interpretable Model-agnostic Explanations* (LIME) e *Anchors* perturbam os dados de entrada e analisam as mudanças na saída do modelo para entender quais recursos têm o maior impacto na decisão final. Esses métodos não foram amplamente adotados e, até recentemente, segundo Tsai *et al.* (2014), poucos autores se concentraram em atingir um equilíbrio entre performance preditiva e interpretabilidade.

3. REVISÃO SISTEMÁTICA DE LITERATURA

Este capítulo está inserido na primeira etapa do método de pesquisa empregado nesta dissertação, ou seja, a fase de contextualização do problema.

3.1 O Método de Revisão Sistemática de Literatura

A RSL pode ser entendida como sendo a metodologia de geração do embasamento teórico-científico necessário para compreensão de um tema por meio da coleta, compreensão, síntese e avaliação de um conjunto de trabalhos científicos (LEVY; ELLIS, 2006; XIAO; WATSON, 2019). De acordo com Sousa Junior *et al.* (2019), a RSL auxilia o pesquisador na condução de todas as etapas metodológicas, tais como na definição do problema, seleção do método, coleta de dados, e análise, contribuindo para mitigação de falhas nas conclusões do trabalho e permitindo uma maior clareza sobre os procedimentos de pesquisa empregados.

A RSL não deve ser considerada como uma RL tradicional, mas sim como um projeto de pesquisa que visa explorar de forma clara uma questão específica utilizando a literatura existente. A RSL se difere das demais metodologias de RL pelo seu princípio exato, a qual o pesquisador deve pré-especificar e deixar claro aos leitores seus critérios de inclusão/exclusão de estudos (DENYER; TRANFIELD, 2009). Oliveira *et al.* (2016) apontam como uma das grandes vantagens da RSL, sobre os métodos tradicionais de RL, é que ela permite evitar a tendenciosidade e distorções nos dados. O principal intuito da RSL é promover o avanço e desenvolvimento de novos estudos, uma vez que para ultrapassar a fronteira do conhecimento deve-se saber onde está fronteira realmente se encontra.

3.2 Aplicação do Método de Revisão Sistemática de Literatura

Com o propósito de investigar sistematicamente o conhecimento em aplicações de ML em CS, identificando as fronteiras e lacunas da literatura existente; e prover novas diretrizes para trabalhos futuros, esta RSL adotará os mesmos procedimentos adotados nos trabalhos de Oliveira *et al.* (2016) e Sousa Junior *et al.* (2019), os quais são conduzidos em quatro fases principais: planejamento; pesquisa/triagem; análise/síntese; e apresentação de resultados.

3.2.1 Planejamento

A motivação principal deste estudo surge da falta de trabalhos que reúnam e discutam de forma abrangente e sistemática o conhecimento sobre ML em CS, possibilitando que os resultados obtidos possam auxiliar em todas as etapas de projetos com contexto similar, bem como uma visão estado da arte, compilando o que já foi estudado e, principalmente, oferecendo oportunidades de novos trabalhos.

Deste modo, os objetivos desta RSL são:

- Identificar e explorar os principais aspectos relacionados a estudos de ML em CS;
- Analisar e sumarizar os resultados encontrados;
- Discutir os resultados, delimitar o estado da arte, lacunas e tendências a serem exploradas.

De modo a atingir este propósito, a formulação das questões de pesquisa (QPs) é uma etapa fundamental na extração e análise do conhecimento intrínseco do tema. Em respeito a isto, diversos *frameworks* para estruturação de QPs foram descritos na literatura, tais como: o método PICO, PICOS ou PICOC (*Patient* ou *Population* ou *Problem*, *Intervention*, *Comparison*, *Outcomes*, e *Context*) (KLODA; BORUFF; CAVALCANTE, 2020), SPICE (*Setting*, *Perspective*, *Intervention*, *Comparison*, e *Evaluation*) (BOOTH, 2006), SPIDER (*Sample*, *Phenomenon of interest*, *Design*, *Evaluation*, e *Research type*) (METHLEY *et al.*, 2014), ECLIPSE (*Expectation*, *Client*, *Location*, *Impact*, *Professionals*, e *Service*) (WILDRIDGE; BELL, 2002), entre outros.

Esta pesquisa utilizará o *framework* PICOC e envolve a combinação de um problema (*problem*) que necessita de uma intervenção (*intervention*) para, em comparação a determinada coisa (*comparison*), produzir melhores resultados (*outcomes*) em determinado contexto (*context*). Neste sentido, as QPs desta pesquisa foram divididas em cinco elementos delineados pelo método PICOC:

- *Problem*: Na modelagem a partir de algoritmos de classificação...
- *Intervention*: ...pode a abordagem de ML,
- *Comparison*: em comparação às metodologias estatísticas tradicionais,
- *Outcomes*: resultar em uma melhoria de performance...
- *Context*: ... no contexto de CS?

Dado estes objetivos, as QPs foram formuladas em atendimento ao escopo do estudo e de modo a identificar as lacunas existentes na literatura, e incluem:

QP1) Quais são os principais algoritmos de classificação em CS? (*problem*)

- QP2) Como é feita a modelagem em projetos CS? (*context*)
- QP3) Quais são os algoritmos de ML mais utilizados em CS? (*intervention*)
- QP4) O quão importante e quais são as estratégias para pré-processamento de dados? (*intervention*)
- QP5) O quão importante e quais são as estratégias para *resampling*? (*intervention*)
- QP6) O quão importante e quais são as estratégias para FS? (*intervention*)
- QP7) O quão importante e quais são as estratégias para OHP? (*intervention*)
- QP8) A modelagem de CSMs por meio de algoritmos de ML é superior às metodologias tradicionais? (*comparison*)
- QP9) Como os resultados são mensurados/validados? (*outcomes*)
- QP10) Quais os melhores algoritmos de ML para CS? (*outcomes*)

No propósito de responder as QPs, as informações a serem extraídas dos artigos foram divididas em três categorias. A primeira categoria investiga as metodologias de modelagem de CSMs, que inclui as bases de dados e algoritmos de classificação utilizados, como eles são treinados, as etapas de pré-processamento de dados, *resampling*, FS e OHP. A segunda categoria desenvolve questões ligadas à avaliação e aplicação prática dos modelos, abordando as métricas de performance utilizadas e sua adequação com as normas vigentes para aplicação de CSMs na indústria. Por fim, a terceira e última categoria abrange o estado da arte, envolvendo uma breve análise dos dados bibliométricos da literatura.

3.2.2 Pesquisa e Triagem dos Artigos

A pesquisa exploratória sobre o tema foi realizada nas bases de dados científicas *Scopus*, *ScienceDirect*, *Taylor and Francis (T&F)*, *Institute of Electrical and Electronics Engineers (IEEE) Xplore* e *Web of Science (WoS)*, uma vez que estas bases são consideradas as principais fontes de dados de citações (MONGEON; PAUL-HUS, 2016) e possuem uma boa representatividade sobre as publicações acerca do tema. Sendo assim, utilizando-se de Lógica Booleana (*AND/OR*), foram pesquisados nos mecanismos de busca das bases de dados os termos relacionadas a ML (*Machine Learning*) e CS (*Credit Scoring*, *Credit Score*, *Credit Analysis*, *Credit Risk* e *Default Risk*). A meta-busca retornou 48 artigos da *Scopus*, 20 da *ScienceDirect*, 98 da T&F, 33 da *IEEE Xplore* e 49 da *WoS*, gerando um total de 248 trabalhos, reduzidos diretamente para 206 ao se eliminarem as instâncias repetidas decorrentes da sobreposição entre as bases.

A fase inicial de seleção dos artigos a serem baixados foi dada pela leitura dos títulos/resumos dos artigos em busca daqueles que atendam aos seguintes critérios de inclusão: (a) possuir no título/resumo/palavras-chave os termos de busca para ML em CS; (b) publicados em periódicos com revisão aos pares; (c) escritos na língua inglesa; e (d) possuir o artigo completo disponível. Após a aplicação dos critérios de inclusão, 105 trabalhos foram excluídos da busca por descumprirem pelo menos um dos critérios definidos anteriormente. No total, 101 foram baixados para etapa de avaliação de qualidade. A Figura 3.1 apresenta os procedimentos de pesquisa/triagem.

Procedimentos de Pesquisa/Triagem						
Seleção de bases de dados	Scopus	ScienceDirect	T&F	IEEE	WoS	Total
Pesquisa por artigos	48	20	98	33	49	248
Triagem	36	20	97	33	20	206
Avaliação de qualidade	30	18	34	4	15	101

Figura 3.1 – Processo de busca/triagem
Fonte: Autoria própria (2023)

A avaliação de qualidade consiste na leitura do título, resumo, objetivos, métodos, resultados e discussões. Nesta etapa foram excluídos da análise 14 artigos que não aplicaram algoritmos de classificação ou o fizeram sem a finalidade de CS. Seguindo a avaliação de qualidade, 87 artigos foram selecionados para leitura completa, extração de informações e análise dos dados.

3.2.3 Análise e Síntese

A síntese das informações extraídas dos 87 artigos foi realizada com o auxílio do software *Microsoft Excel*®. Para responder as QPs os artigos foram avaliados em 17 itens, e correspondem a: algoritmos de classificação utilizados; número de CSMs testados; número de bases de dados avaliadas; origem das bases de dados; número de amostras; número de atributos; estratégias de amostragem para treinamento, teste e avaliação; estratégias de pré-processamento de dados; estratégias de *resampling*; estratégias de FS; estratégias de OHP; indicadores de performance e método de validação; interpretabilidade; e principais autores, periódicos, ano de publicação e palavras-chave adotados.

Após a síntese, a análise dos resultados foi realizada em consonância com as QPs da pesquisa, sendo considerada a base das conclusões que suportam o desenvolvimento do corpo

de evidências sobre as principais perspectivas sobre as aplicações de ML em CS discutidas nesta RSL.

3.2.4 Apresentação dos Resultados

Os resultados foram resumidos em gráficos e tabelas para permitir a discussão sobre este tópico, revelando as percepções mais contundentes desta RSL. Com base nesta análise da interação entre ML e o processo de CS, é possível refletir e identificar as lacunas e percepções que cercam o desenvolvimento do tema na comunidade científica.

Após as análises e sínteses, os principais resultados estão resumidos nos Quadros 3.1 – 3.5 na sequência, na qual serão discutidas as QPs e o estado da arte sobre aplicações de ML em CS e as possibilidades de pesquisas no futuro.

3.3 Resultados e discussões da Revisão Sistemática de Literatura

Esta seção corresponde ao último passo da RSL, tendo como objetivo apresentar os dados obtidos a partir dos artigos analisados e responder as QPs propostas. Em primeiro lugar, como pode ser observado na Figura 3.2, ML tornou-se um assunto popular na área de gestão de riscos financeiros, verificando-se um significativo aumento da produção científica sobre o assunto nos últimos anos.

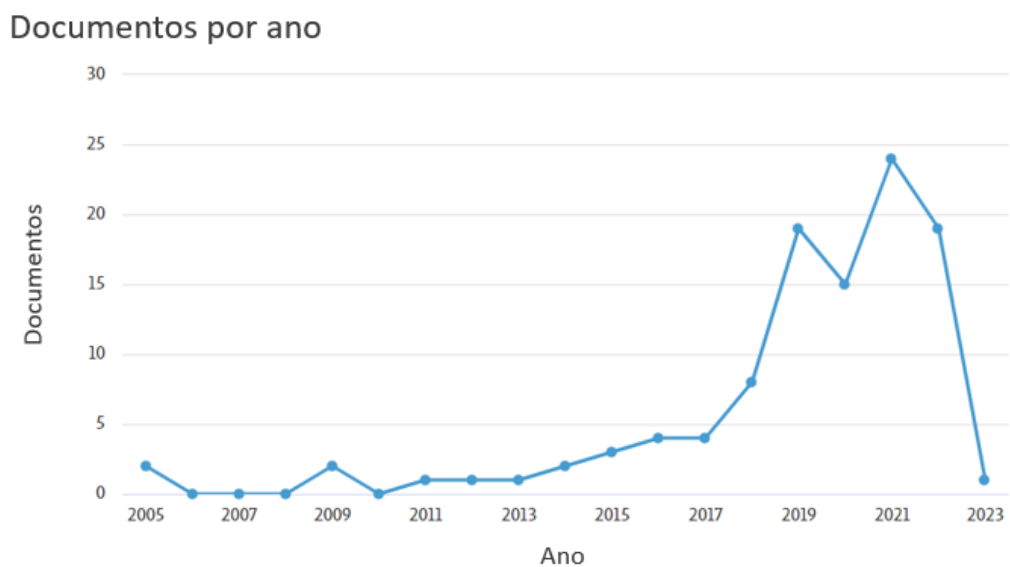


Figura 3.2 – Produção científica anual de acordo com a base Scopus em 2022
Fonte: Scopus

Quanto ao estado da arte das pesquisas acerca do tema, as Figuras 3.3-3.6 trazem dados interessantes acerca das publicações pertinentes, organizadas durante uma análise bibliométrica dos artigos. Em primeiro lugar, na Figura 3.3, encontram-se as principais fontes de pesquisas sobre a modelagem para classificação do risco de crédito, destacando-se revistas voltadas para ciências da computação, sistemas de informação e pesquisa operacional.

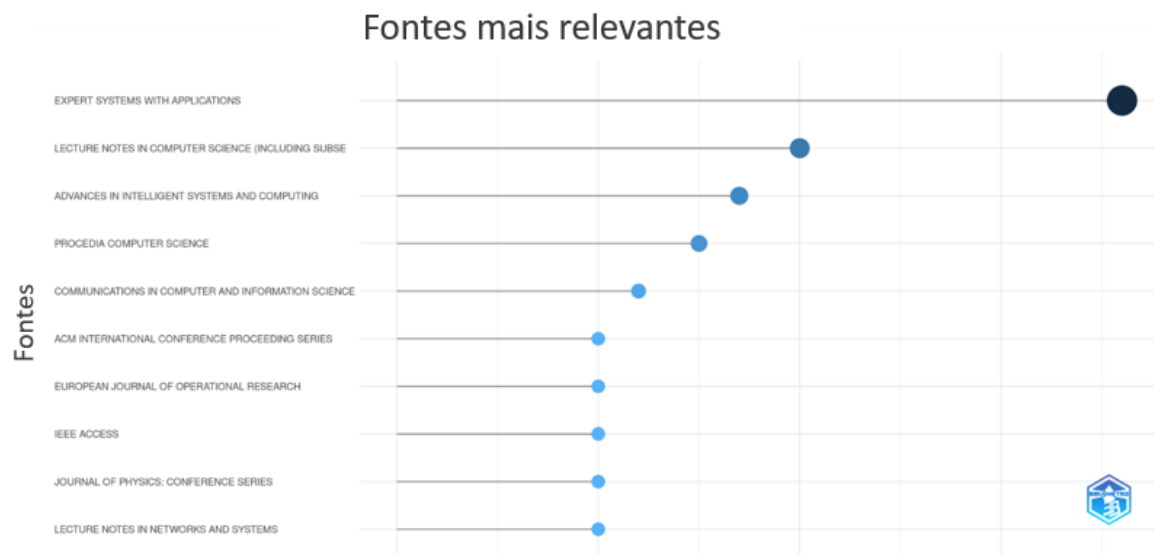


Figura 3.3 – Top 10 fontes de pesquisas no tema
Fonte: Scopus

Quanto aos países com a maior produção científica na área, apresentados na Figura 3.4, destacam-se a China e os Estados Unidos. Vale apontar que, apesar de significativamente menor do que os dois líderes, o Brasil se encontra entre os 5 países de destaque no que diz respeito ao escopo deste trabalho.

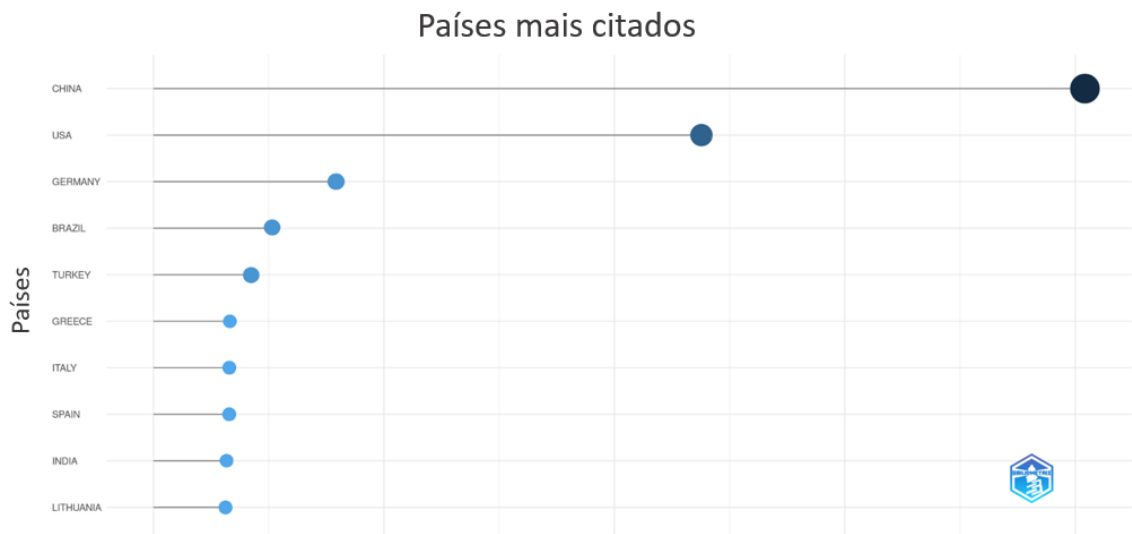


Figura 3.4 – Top 10 países em produção científica no tema
Fonte: Scopus (2023)

Em seguida, a Figura 3.5 apresenta a rede de co-citações dos principais autores. Analisando a figura, é possível observar não só os principais autores, como *clusters* que os dividem conforme as citações entre eles, permitindo que sejam identificados também os principais grupos de pesquisa sobre o assunto.

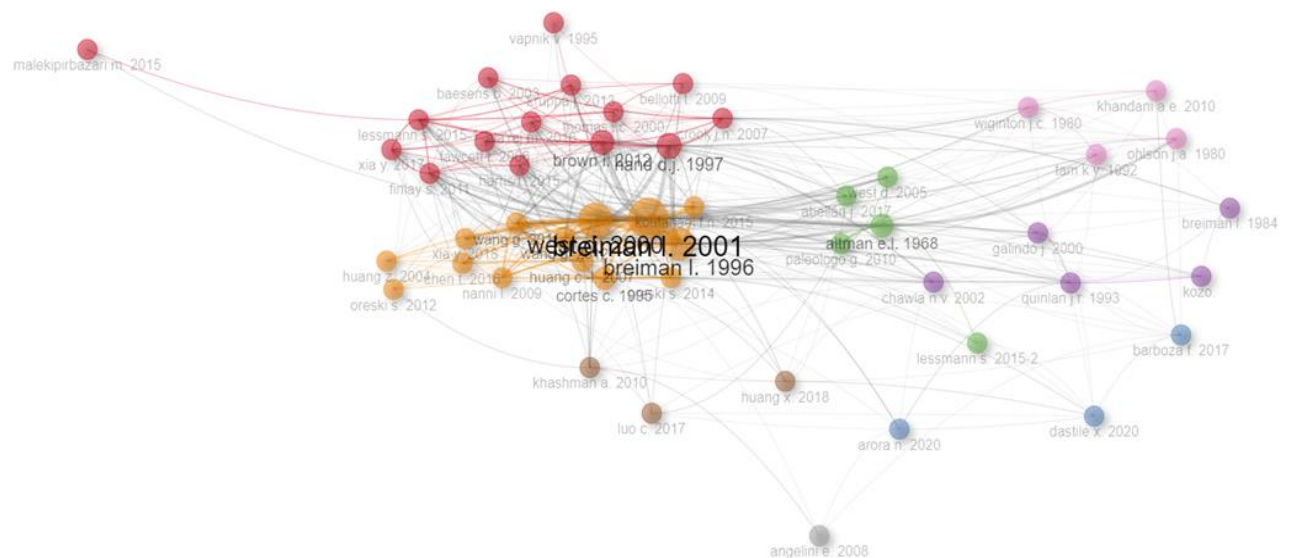


Figura 3.5 – Rede de co-citações dos principais autores
Fonte: Scopus

apresentando as tecnologias/métodos mais frequentes de cada tópico estudado e auxiliando na resposta das QPs 1 a 7.

Em primeiro lugar, a primeira divisão entre os artigos refere-se à natureza dos conjuntos de dados empregados na modelagem, considerando que são intrinsecamente relacionados ao campo de aplicação. Como mencionado, o foco deste artigo é o risco de crédito ao consumidor. Portanto, a maioria dos dados coletados se refere a artigos que empregaram ML no processo de modelagem do risco de crédito ao consumidor. No entanto, 25,97% dos artigos consultados (um total de 20 artigos) foram totalmente ou parcialmente dedicados ao risco de crédito corporativo. Os classificadores utilizados nesses artigos para conjuntos de dados corporativos são apresentados no Quadro 3.1.

Quadro 3.1 – Classificadores utilizados em artigos que empregaram dados de crédito corporativo

Autor	Tipo de Risco	No vel ?	Algoritmo	Acrônimo	Tipo	Individual	Ensembles Homogêneos	Ensembles Heterogêneos
Wang, Wang e Lai (2005)	Consumidor/ Corporativo	✓	<i>Unilateral-weighted Fuzzy SVM</i> <i>Bilateral-weighted Fuzzy SVM</i>	U-FSVM B-FSVM	Híbrido	SVM	-	-
Niklis, Doumpos e Zopounidis (2014)	Corporativo		-	-	-	LR/SVM-L/ASVM SVM-RBF	-	-
Petropoulos, Chatzis e Xanthopoulos (2016)	Corporativo	✓	<i>Student's-t Hidden Markov Model</i>	SHMM	Estatístico	LR/ChAID/FNN/SVM	RF	-
Bussmann <i>et al.</i> (2021)	Corporativo		-	-	-	LR	XGBoost	-
Aghaeirad, Chen e Ribeiro (2017)	Consumidor/ Corporativo		-	-	-	FNN/SOM + FNN	-	-
Fenerich <i>et al.</i> (2020)	Corporativo		-	-	-	DT/SVM/BN ET	-	-
Li <i>et al.</i> (2021)	Corporativo	✓	<i>Generative Adversarial Nets combined with XGBoost</i>	G-XGBoost	Híbrido	-	XGBoost	-

Bou-Hamad (2017)	Corporativo	✓	<i>Bayesian Model Averaging for Logistic Regression</i>	BMALR	Estatístico	-	RF-CART/RF-RF-CI	-
Martens <i>et al.</i> (2010)	Consumidor/Corporativo	✓	<i>Ant Colony Optimization Rule-based Algorithm</i>	AntMiner+	Rule-based	C4.5/SVM	-	MV
Corazza, March e Di Tollo (2021)	Corporativo	✓	<i>Adaptive Elman Approach</i>	-	NN	-	-	-
Zhou <i>et al.</i> (2021)	Consumidor/Corporativo		-	-	-	LR/CART/NN/SVM	-	-
Coenen, Verbeke e Guns (2022)	Corporativo		-	-	-	LR/SVM/SVR	-	-
Chen, Härdle e Moro (2011)	Corporativo		-	-	-	SVM	-	-
Moscatelli <i>et al.</i> (2020)	Corporativo		-	-	-	LDA/LR/PLR	RF/GBDT	-
Liang <i>et al.</i> (2018)	Consumidor/Corporativo	✓	<i>Unanimous Voting Ensemble</i>	UV	<i>Ensemble heterogêneo (Base Learners: SVM-L, Bag MLP, Boost CART)</i>	CART/MLP/SVM-L/NB	Bag SVM-L/Bag KNN/Bag MLP/Bag CART/Bag NB/AdaBoost SVM-L/AdaBoost KNN/AdaBoost MLP/AdaBoost CART/AdaBoost NB	<i>Stacking (Base Learners: SVM-L, Bag MLP, Boost CART)/MV (Base Learners: SVM-L, Bag MLP, Boost CART)</i>
Sariev e Germano (2020)	Consumidor/Corporativo	✓	<i>Bayesian MCMC Regularization Artificial Neural Network</i>	Bayesian MCMC NN	NN	NN/Classical Regularization NN/Bayesian Regularization NN/SVM/KN	-	-
Sun <i>et al.</i> (2015)	Consumidor/Corporativo	✓	<i>Imbalance Oriented Multiple-Classifer Ensemble</i>	IOMCE	<i>SVM Homogenous Ensemble</i> <i>MDA Homogenous Ensemble</i>	MDA/SVM-RBF	-	-

Zhang <i>et al.</i> (2022)	Consumidor /Corporativo		-		-	KNN	RF	-
Huang <i>et al.</i> (2011)	Corporativo		-		-	LR/SVM/MS VM/BNET/K NN	-	-
Huang e Huang (2010)	Corporativo		-		-	LR/SVM/MS VM/BNET/K NN	-	-
Huang, Wang e Wang (2018)	Consumidor/ Corporativo		-	-	-	SVM	-	-

*Abreviações utilizadas: ASVM = *Additive SVM*; BNET = *Bayesian Network*; CART = *Classification And Regression Tree*; ChAID = *Chi-square Automatic Interaction Detection*; CI = *Conditional Inference Trees*; FNN = *Feedforward NN*; GBDT = *Gradient Boosted DT*; KNN = *k-Nearest Neighbors*; MDA = *Multiple Discriminant Analysis*; MLP = *Multilayer Perceptron*; MSVM = *Multi-class SVM*; MV = *Majority Vote*; NB = *Naïve Bayes*; PLR = *Penalized LR*; SOM = *Self-organizing Maps*; SVM-L = *SVM Linear*; SVR = *Support Vector Regression*.

Fonte: Artigos consultados na RSL

Métodos *novel* correspondem àqueles que contam com algoritmos avançados desenvolvidos recentemente, na maior parte das vezes propostos pelos autores. Mais importante são os modelos *baseline* testados e listados pelos artigos, que são mais fáceis de implementar e reproduzir, e encontra-se divididos em individual, *ensembles* homogêneos e *ensembles* heterogêneos. O crescente interesse em ML já se evidencia aqui, considerando os diversos algoritmos de ML encontrados. Para CS no contexto corporativo, parece que modelos *ensemble* ainda são uma tendência minoritária, aparecendo em cerca de 20% dos artigos. Dos 66 diferentes classificadores *baseline*, a maior parte trata de modelos individuais (78,79%), especialmente no caso do SVM (WANG; WANG; LAI, 2005; NIKLIS; DOUMPOS; ZOPOUNIDIS, 2014; PETROPOULOS; CHATZIS; XANTHOPOULOS, 2016; FENERICH *et al.*, 2020; ZHOU *et al.*, 2021; COENEN; VERBEKE; GUNS, 2022; CHEN; HÄRDLE; MORO, 2011; LIANG *et al.*, 2018; SARIEV; GERMANO, 2020; SUN *et al.*, 2015; ZHANG *et al.*, 2022; HUANG *et al.*, 2011; HUANG; HUANG, 2010; MARTENS *et al.*, 2010), NN (PETROPOULOS; CHATZIS; XANTHOPOULOS, 2016; AGHAEIRAD; CHEN; RIBEIRO, 2017; ZHOU *et al.*, 2021; LIANG *et al.*, 2018; SARIEV; GERMANO, 2020), DT (PETROPOULOS; CHATZIS; XANTHOPOULOS, 2016; FENERICH *et al.*, 2020; ZHOU *et al.*, 2021; LIANG *et al.*, 2018; MARTENS *et al.*, 2010) e LR (NIKLIS; DOUMPOS;

ZOPOUNIDIS, 2014; PETROPOULOS; CHATZIS; XANTHOPOULOS, 2016; BUSSMANN *et al.*, 2021; ZHOU *et al.*, 2021; COENEN; VERBEKE; GUNS, 2022; MOSCATELLI *et al.*, 2020; HUANG *et al.*, 2011; HUANG; HUANG, 2010), que costumam servir de *benchmark* para outros modelos.

Além da LR tradicional, Moscatelli *et al.* (2020) também experimenta com *Penalized LR* (PLR). No caso do SVM, não foram encontradas muitas especificações dos modelos. Foram encontrados também exemplos de SVM-L (NIKLIS; DOUMPOS; ZOPOUNIDIS, 2014; LIANG *et al.*, 2018), o popular SVM-RBF (NIKLIS; DOUMPOS; ZOPOUNIDIS, 2014; SUN *et al.*, 2015), assim como MSVM (HUANG; HUANG, 2010; HUANG *et al.*, 2011) e ASVM (NIKLIS; DOUMPOS; ZOPOUNIDIS, 2014). Coenen, Verbeke and Guns (2022) ainda experimentam o algoritmo SVR. Diferentes tipos de DTs foram encontradas também, como *Chi-square Automatic Interaction Detection* (ChAID) (PETROPOULOS; CHATZIS; XANTHOPOULOS, 2016); *Classification And Regression Trees* (CART) (ZHOU *et al.*, 2021; LIANG *et al.*, 2018) e C4.5 (MARTENS *et al.*, 2010). Quanto a NN (ZHOU *et al.*, 2021; SARIEV; GERMANO, 2020), a maior parte eram *feedforward* (FNNs) (PETROPOULOS; CHATZIS; XANTHOPOULOS, 2016; AGHAEIRAD; CHEN; RIBEIRO, 2017), como no caso do tradicional *Multi-Layer Perceptron* (MLP) (LIANG *et al.*, 2018).

3.3.1.1 Classificadores

Restringindo-nos agora aos trabalhos com dados sobre a concessão de crédito ao consumidor, assim como no caso das bases corporativas, muitos algoritmos *novel* foram encontrados, e encontram-se dispostos preliminarmente no Quadro 3.2.

Quadro 3.2 – Classificadores novel utilizados em artigos que empregaram dados de crédito ao consumidor

Autor	Algoritmo	Acrônimo	Tipo
Wu <i>et al.</i> (2021)	<i>Deep Multiple Kernel Classifier</i>	DMKC	Híbrido
Chen e Xiang (2017)	-	Group LASSO	Estatístico
Florez-Lopez e Ramon-Jeronimo (2015)	<i>Correlated-adjusted Decision Forest</i>	CADF	<i>Ensemble</i> homogêneo de DTs
Dumitrescu <i>et al.</i> (2022)	<i>Penalized Logistic Tree Regression</i>	PLTR	Estatístico/ <i>Rule-Based</i>

Plawiak, Abdar e Acharya (2019)	<i>Deep Genetic Cascade Ensemble of Classifiers</i>	DGCEC	<i>Ensemble homogêneo de SVMs</i>
Giri <i>et al.</i> (2021)	<i>Locally and Globally tuned Biogeography Based Optimization</i>	LGBBO	<i>Rule Miner</i>
Shi, Zhang e Qiu (2013)	<i>F-score Feature Weighted SVM Random Forest Feature Weighted SVM</i>	FS-FWSVM RF-FWSVM	<i>Feature Weighted SVMs</i>
Wang, Xu e Zhou (2015)	<i>Lasso-logistic Regression Ensemble</i>	LLRE	<i>Ensemble homogêneo (Weighted Voting com os seguintes base learners: LASSO/LR)</i>
Siami, Gholamian e Basiri (2014)	<i>Locally Linear Neuro-fuzzy Model with Model Tree Learning</i>	LOLIMOT	<i>Ensemble homogêneo</i>
Martens <i>et al.</i> (2010)	AntMiner+	Ant Colony Optimization	<i>Rule-based</i>
Li e Chen (2021)	-	NN-Entropy	Estatístico
Choi, Koo e Park (2015)	<i>Fused Lasso with AIC model selection Fused Lasso with BIC model selection</i>	-	Estatístico Estatístico
Mancisidor <i>et al.</i> (2020)	<i>Semi-supervised Gaussian mixture parametrized MLP</i>	-	NN semi-supervisionada
Li <i>et al.</i> (2021)	<i>Multilayer LightGBM</i>	ML-LightGBM	<i>Ensemble homogêneo (boosting)</i>
Liang <i>et al.</i> (2018)	<i>Unanimous Voting Ensemble</i>	UV	<i>Ensemble heterogêneo (base learners: SVM-L/Bag MLP/Boost CART)</i>
Sariev e Germano (2020)	<i>Bayesian MCMC Regulatization Artificial Neural Network</i>	Bayesian MCMC NN	NN
Pan (2021)	<i>Continuous Colaborative Filtering Neural Network Discrete Colaborative Filtering Neural Network</i>	CCF-NN DCF-NN	NN
Sun <i>et al.</i> (2015)	<i>Imbalance Oriented Multiple-Classififer Ensemble</i>	IOMCE	<i>Ensemble homogêneo de SVMs Ensemble homogêneo de MDA</i>
Li <i>et al.</i> (2022)	<i>One-class Classification Driven Dynamical Ensemble Learning</i>	OCDEL	Algoritmo ensemble de OOC
Zhang <i>et al.</i> (2022)	-	-	0-1 programming
Bai, Zhen e Shen (2022)	<i>Gradient Boosting Survival Tree</i>	GBST	<i>Ensemble não paramétrico</i>
Wang, Wang e Lai (2005)	<i>Unilateral-weighted Fuzzy SVM Bilateral-weighted Fuzzy SVM</i>	U-FSVM B-FSVM	SVM híbrido
Chen <i>et al.</i> (2020)	<i>Generalized Shapley Choquet integral-based Ensemble Approach</i>	GSCI-based Ensemble	<i>Ensemble heterogêneo (base learners: Bagging, RF, AdaBoost, GBM, ET)</i>

Zhang <i>et al.</i> (2018)	<i>Multiple Instance Learning Algorithms</i>	RBF-MIL-LDA RBF-MIL-LR RBF-MIL-DT RBF-MIL-SVM CKNN EMDD MIL Boost MissSVM miSVM	<i>Multiple Instance Learning</i>
Tripathi <i>et al.</i> (2020)	<i>Evolutionary Extreme Learning Machine</i>	EELM	FNN
Chopra e Bilare (2018)	<i>Diverse Ensemble Creation by Oppositional Relabeling of Artificial Training Examples</i>	DECORATE	<i>Ensemble heterogêneo</i>
Weng, Huang e Cheng-Kui (2021)	<i>Clustering Based DT</i> <i>Clustering Based MLP</i> <i>Clustering Based NB</i> <i>Clustering Based SVM</i>	CBDT CBMLP CBNB CBSVM	<i>Ensemble híbrido e não supervisionado</i>
Liu, Fu e Lin (2010)	-	gr-GA-SVM	SVM híbrido
Armaki <i>et al.</i> (2017)	<i>Hybrid Meta-Learner Models</i>	-	40 algoritmos <i>ensemble</i> híbridos e heterogêneos de diferentes modelos de <i>deep learning</i> e <i>clustering</i>

*Abreviações utilizadas: CKNN = *Citation-KNN*; EMDD = *Emotion Mode Diversity Density*; ET = *Extra Trees*; LASSO = *Least Absolute Shrinkage Selection Operator*; MCMC = *Markov Chain Monte-Carlo*.

Fonte: Artigos consultados na RSL

De acordo com Lessmann *et al.* (2015), esse comum *setup* de pesquisa geralmente foca na construção e teste de novos e complexos algoritmos, testando apenas alguns métodos de referência, na melhor das hipóteses. Essa informação dos autores faz sentido, considerando que os artigos do Quadro 3.2 são responsáveis por aproximadamente apenas um terço (33,01%) de todos os modelos. Essa configuração também tem uma limitação, e os resultados dos classificadores novos podem ser excessivamente otimistas porque os desenvolvedores são mais adeptos a uma abordagem específica, ou o novo método pode ter sido ajustado mais intensivamente do que suas referências (HAND, 2006; THOMAS, 2010). Na verdade, na maioria desses casos, o classificador proposto teve um desempenho melhor do que a maioria, senão a totalidade, dos métodos usados como *benchmarks*.

Os métodos *baseline* e de referência estão listados abaixo nos Quadros 3.3-3.5. Como o objetivo é reunir sistematicamente informações sobre o assunto, é melhor analisar estes métodos, considerando que são responsáveis pela maioria dos modelos, bem como mais fáceis de entender e reproduzir.

Quadro 3.3 – Modelos estatísticos *baseline* em artigos sobre crédito ao consumidor

Estatístico (14,33%)	
Regressões (37)	LR (WU <i>et al.</i> , 2021; FLOREZ-LOPEZ; RAMON-JERONIMO, 2015; DUMITRESCU <i>et al.</i> , 2022; MUNKHDALAI <i>et al.</i> , 2019; ANTONAKIS; SFAKIANAKIS, 2009; KHEMAKEM; BEM SAID; BOUJELBENE, 2018; ZHENG, 2019; BAO; LIANJU; YUE, 2019; LI; CHEN, 2022; ZHOU <i>et al.</i> , 2021; MALHOTRA; MALHOTRA, 2000; BOZ <i>et al.</i> , 2018; LI <i>et al.</i> , 2021; DE CASTRO VIEIRA <i>et al.</i> , 2019; PAPOUSKOVA; HAJEK, 2019; DJEUNDJE <i>et al.</i> , 2021; MOSCATO; PICARIELLO; SPERLÍ, 2021; BOUGHACI; ALKHAWALDEH, 2017; WAAD; GHAZI; MOHAMED, 2013; PAN, 2021; LI <i>et al.</i> , 2022; ZHANG <i>et al.</i> , 2022; MARQUÉS; GARCÍA; SÁNCHEZ, 2013a; KENNEDY; NAMEE; DELANY, 2013; ARIZA-GARZÓN, 2020; ZHANG <i>et al.</i> , 2018; WANG <i>et al.</i> , 2019; FITZPATRICK; MUES, 2016; ZIEMBA; RADOMSKA-ZALAS; BECKER, 2020; CHOPRA; BILARE, 2018; HAMDOUN; RGUIBI, 2019; TORKEVAR; GAME, 2019; ARMAKI <i>et al.</i> , 2017) Non-Linear LR (DUMITRESCU <i>et al.</i> , 2022) Non-Linear LR+ALasso (DUMITRESCU <i>et al.</i> , 2022) LASSO-LR (WANG; XU; ZHOU, 2015) LR+RPart (ALEKSANDROVA, 2021)
DA (6) (DE CASTRO VIEIRA <i>et al.</i> , 2019)	LDA (FLOREZ-LOPEZ; RAMON-JERONIMO, 2015; ANTONAKIS; SFAKIANAKIS, 2009; ZHANG <i>et al.</i> , 2022; KENNEDY; NAMEE; DELANY, 2013; ZHANG <i>et al.</i> , 2018) MDA (SUN <i>et al.</i> , 2015)
n.a (1)	GAM (FITZPATRICK; MUES, 2016)

*Abreviações utilizadas: GAM = *Generalized Additive Model*; PART = *Partial DT*.

Fonte: Artigos consultados na RSL

Quadro 3.4 – Modelos de ML individuais *baseline* encontrados em artigos sobre crédito ao consumidor

Individual (56,68%)	
Outros rule based (12)	OneR (GIRI <i>et al.</i> , 2021; BOUGHACI; ALKHAWALDEH, 2017) JRip (GIRI <i>et al.</i> , 2021; BERKA, 2016) Decision Table (GIRI <i>et al.</i> , 2021; ZIEMBA; RADOMSKA-ZALAS; BECKER, 2020; ZIEMBA <i>et al.</i> , 2021) Conjunctive Rule (GIRI <i>et al.</i> , 2021) KEX1 (BERKA, 2016) KEX2 (BERKA, 2016) PRISM (BERKA, 2016) Ridor (BERKA, 2016)
DT (39) (CHOPRA; BILARE, 2018; ANTONAKIS; SFAKIANAKIS, 2009; SIAMI; GHOLAMIAN; BASIRI, 2014; BOZ <i>et al.</i> , 2018; AMPOUNTOLAS <i>et al.</i> , 2021; LI <i>et al.</i> , 2021; DE CASTRO VIEIRA <i>et al.</i> , 2019; ARIZA-GARZÓN, 2020; ZHANG <i>et al.</i> , 2018; WANG <i>et al.</i> , 2019; KOKATE; CHETTY, 2021; WENG; HUANG; CHENG-KUI, 2021; ARMAKI <i>et al.</i> , 2017)	J48 (WU <i>et al.</i> , 2021; GIRI <i>et al.</i> , 2021) ChAID (FLOREZ-LOPEZ; RAMON-JERONIMO, 2015) C4.5 (FLOREZ-LOPEZ; RAMON-JERONIMO, 2015; BAO; LIANJU; YUE, 2019; MARTENS <i>et al.</i> , 2010; PAPOUSKOVA; HAJEK, 2019; PAN, 2021; ZIEMBA; RADOMSKA-ZALAS; BECKER, 2020; CHOPRA; BILARE, 2018; ZIEMBA <i>et al.</i> , 2021) C5.0 (TRIVEDI, 2020) Assistant (FLOREZ-LOPEZ; RAMON-JERONIMO, 2015) CART (ZHOU <i>et al.</i> , 2021; LIANG <i>et al.</i> , 2018; ZHANG <i>et al.</i> , 2022; CHOPRA; BILARE, 2018) CART-Univariate (FLOREZ-LOPEZ; RAMON-JERONIMO, 2015; WANG; XU; ZHOU, 2015) CART-Oblique (FLOREZ-LOPEZ; RAMON-JERONIMO, 2015) PART (GIRI <i>et al.</i> , 2021; BERKA, 2016) DTCS (BOZ <i>et al.</i> , 2018) CDT (PAPOUSKOVA; HAJEK, 2019; CHOPRA; BILARE, 2018) Heoffding DT (PAPOUSKOVA; HAJEK, 2019)
NN (52) (FLOREZ-LOPEZ; RAMON-JERONIMO, 2015; DUMITRESCU <i>et al.</i> , 2022; ANTONAKIS; SFAKIANAKIS, 2009; KHEMAKEM; BEM SAID; BOUJELBENE, 2018; ZHENG, 2019; BAO; LIANJU; YUE, 2019; SIAMI; GHOLAMIAN; BASIRI, 2014; LI; CHEN, 2022; ZHOU <i>et al.</i> , 2021; PAPOUSKOVA;	FNN (AGHAEIRAD; CHEN; RIBEIRO, 2017) MLP (AMPOUNTOLAS <i>et al.</i> , 2021; MANCISIDOR <i>et al.</i> , 2020; MOSCATO; PICARIELLO; SPERLÍ, 2021; LIANG <i>et al.</i> , 2018; WAAD; GHAZI; MOHAMED, 2013; CHOPRA; BILARE, 2018; PEARS; OETAMA, 2010; WENG; HUANG; CHENG-KUI, 2021; ALEKSANDROVA, 2021)

<p>HAJEK, 2019; DJEUNDJE <i>et al.</i>, 2021; SARIEV; GERMANO, 2020; PAN, 2021; LI <i>et al.</i>, 2022; ZHANG <i>et al.</i>, 2022; CHEN <i>et al.</i>, 2020; ARMAKI <i>et al.</i>, 2017; MARQUÉS; GARCÍA; SÁNCHEZ, 2013a)</p>	<p>Self-learning MLP (AMPOUNTOLAS <i>et al.</i>, 2021; MANCISIDOR <i>et al.</i>, 2020; ARMAKI <i>et al.</i>, 2017) Sigmoid MLP (MUNKHDALAI <i>et al.</i>, 2019) Deep Sigmoid MLP (MUNKHDALAI <i>et al.</i>, 2019) Deeper Sigmoid MLP (MUNKHDALAI <i>et al.</i>, 2019) Softmax MLP (MUNKHDALAI <i>et al.</i>, 2019) Deep Softmax MLP (MUNKHDALAI <i>et al.</i>, 2019) Deeper Softmax MLP (MUNKHDALAI <i>et al.</i>, 2019) SHNN (KHASHMAN, 2009) DHNN (KHASHMAN, 2009) Back Propagation FNN (KENNEDY; NAMEE; DELANY, 2013) 3-Layer Back Propagation Supervised NN (KHASHMAN, 2009) Backpropagation NN with Adaptive Learning (MALHOTRA; MALHOTRA, 2000) Backpropagation NN with Levenberg-Marquandt Approximation (MALHOTRA; MALHOTRA, 2000) Learning Vector Quantization Network (MALHOTRA; MALHOTRA, 2000) Classical Regularization NN (SARIEV; GERMANO, 2020) Bayesian Regularization NN (SARIEV; GERMANO, 2020) ELM (ZHANG <i>et al.</i>, 2022) DeepHit (BAI; ZHEN; SHEN, 2022) LSTM (WANG <i>et al.</i>, 2019) AM-LSTM (WANG <i>et al.</i>, 2019) BLSTM (WANG <i>et al.</i>, 2019) LSTM-Meanpool (WANG <i>et al.</i>, 2019)</p>
<p>SVM (40) (DUMITRESCU <i>et al.</i>, 2022; TRIVEDI, 2020; MUNKHDALAI <i>et al.</i>, 2019; KHEMAKEM; BEM SAID; BOUJELBENE, 2018; SHI; ZHANG; QIU, 2013; ZHENG, 2019; BAO; LIANJU; YUE, 2019; MARTENS <i>et al.</i>, 2010; ZHOU <i>et al.</i>, 2021; BOZ <i>et al.</i>, 2018; TELES <i>et al.</i>, 2021; MANCISIDOR <i>et al.</i>, 2020; PAPOUSKOVA; HAJEK, 2019; SARIEV; GERMANO, 2020; HUANG; WANG; WANG, 2018; WANG; WANG; LAI, 2005; CHEN <i>et al.</i>, 2020; ZHANG <i>et al.</i>, 2018; CHOPRA; BILARE, 2018; PEARS; OETAMA, 2010; TORKEVAR; GAME, 2019; WENG; HUANG; CHENG-KUI, 2021; LIU; FU; LIN, 2010; ARMAKI <i>et al.</i>, 2017)</p>	<p>SVM-L (FLOREZ-LOPEZ; RAMON-JERONIMO, 2015; DE CASTRO VIEIRA <i>et al.</i>, 2019; LIANG <i>et al.</i>, 2018; ZHANG <i>et al.</i>, 2022) SVM-RBF (PŁAWIAK; ABDAR; ACHARYA, 2019; DE CASTRO VIEIRA <i>et al.</i>, 2019; PAN, 2021; SUN <i>et al.</i>, 2015; ZHANG <i>et al.</i>, 2022) SVM-Poly (PŁAWIAK; ABDAR; ACHARYA, 2019) SVM-2Poly (FLOREZ-LOPEZ; RAMON-JERONIMO, 2015) SVM-Sigmoid (PŁAWIAK; ABDAR; ACHARYA, 2019) Self-learning SVM (MANCISIDOR <i>et al.</i>, 2020) S3VM (MANCISIDOR <i>et al.</i>, 2020) FQSSVM (LI <i>et al.</i>, 2022) Pegasos (MOSCATO; PICARIELLO; SPERLÍ, 2021)</p>
<p>NB (16) (TRIVEDI, 2020; ANTONAKIS; SFAKIANAKIS, 2009; BOZ <i>et al.</i>, 2018; LIANG <i>et al.</i>, 2018; WAAD; GHAZI; MOHAMED, 2013; PAN, 2021; ZHANG <i>et al.</i>, 2022; CHEN <i>et al.</i>, 2020; WANG <i>et al.</i>, 2019; ZIEMBA; RADOMSKA-ZALAS; BECKER, 2020; TORKEVAR; GAME, 2019; WENG; HUANG; CHENG-KUI, 2021; ARMAKI <i>et al.</i>, 2017; ZIEMBA <i>et al.</i>, 2021)</p>	<p>LBC (KENNEDY; NAMEE; DELANY, 2013) QBC (KENNEDY; NAMEE; DELANY, 2013)</p>
<p>n.a (15)</p>	<p>BNET (TRIVEDI, 2020; SIAMI; GHOLAMIAN; BASIRI, 2014; BOUGHACI; ALKHAWALDEH, 2017)</p> <p>KNN (FLOREZ-LOPEZ; RAMON-JERONIMO, 2015; ANTONAKIS; SFAKIANAKIS, 2009; BAO; LIANJU; YUE, 2019; AMPOUNTOLAS <i>et al.</i>, 2021; SARIEV; GERMANO, 2020; PAN, 2021; ZHANG <i>et al.</i>, 2022; KENNEDY; NAMEE; DELANY, 2013; WANG <i>et al.</i>, 2019; ZIEMBA; RADOMSKA-ZALAS; BECKER, 2020; ARMAKI <i>et al.</i>, 2017; ZIEMBA <i>et al.</i>, 2021)</p>

*Abreviações utilizadas: AM-LSTM = *Attention Mechanism LSTM*; BLSTM = *Bidirectional LSTM*; CDT = *Credal DT*; DHNN = *Double Hidden Layer NN*; DTCS = *Cost-sensitive DT*; FQSSVM = *Fuzzy Kernel-free SVM*; LBC = *Linear Bayes Normal*; LSTM = *Long Short-term Memory Network*; PEGASOS; QBC = *Quadratic Bayes Normal*; SHNN = *Single Hidden Layer NN*; SVM-Poly = *Polynomial SVM*.

Fonte: Artigos consultados na RSL

Quadro 3.5 – Modelos *ensemble baseline* de ML encontrados em artigos sobre crédito ao consumidor

Ensemble (28,99%)			
Homogêneo		Heterogêneo	
Decision Forest (37)	RF (FLOREZ-LOPEZ; RAMON-JERONIMO, 2015; DUMITRESCU <i>et al.</i> , 2022; TRIVEDI, 2020; MUNKHDALAI <i>et al.</i> , 2019; CHOPRA; BILARE, 2018; ZHENG, 2019; BAO; LIANJU; YUE, 2019; WANG; XU; ZHOU, 2015; BOZ <i>et al.</i> , 2018; AMPOUNTOLAS <i>et al.</i> , 2021; TELES <i>et al.</i> , 2021; LI <i>et al.</i> , 2021; DE CASTRO VIEIRA <i>et al.</i> , 2019; PAPOUSKOVA; HAJEK, 2019; MOSCATO; PICARIELLO; SPERLÍ, 2021; BOUGHACI; ALKHAWALDEH, 2017; WAAD; GHAZI; MOHAMED, 2013; PAN, 2021; ZHANG <i>et al.</i> , 2022; CHEN <i>et al.</i> , 2020; ARIZA-GARZÓN, 2020; WANG <i>et al.</i> , 2019; FITZPATRICK; MUES, 2016; ZIEMBA; RADOMSKA-ZALAS; BECKER, 2020; HAMDOUN; RGUIBI, 2019; TORKEVAR; GAME, 2019; WENG; HUANG; CHENG-KUI, 2021; ZIEMBA <i>et al.</i> , 2021; MARQUÉS; GARCÍA; SÁNCHEZ, 2013a) SVDF (FLOREZ-LOPEZ; RAMON-JERONIMO, 2015) WVDF (FLOREZ-LOPEZ; RAMON-JERONIMO, 2015) ET (AMPOUNTOLAS <i>et al.</i> , 2021; CHEN <i>et al.</i> , 2020) Forest PA (PAPOUSKOVA; HAJEK, 2019) CIF (BAI; ZHEN; SHEN, 2022) RSF (BAI; ZHEN; SHEN, 2022) Rotation Forest (CHOPRA; BILARE, 2018; ALEKSANDROVA, 2021) ORF (ZIEMBA <i>et al.</i> , 2021)	n.a. (10)	HWA (CHEN <i>et al.</i> , 2020)
			WA (CHEN <i>et al.</i> , 2020)
Boosting (30)	XGBoost (MUNKHDALAI <i>et al.</i> , 2019; ZHENG, 2019; DJEUNDJE <i>et al.</i> , 2021; BAI; ZHEN; SHEN, 2022; WANG <i>et al.</i> , 2019; ARIZA-GARZÓN, 2020; ALEKSANDROVA, 2021; MARQUÉS; GARCÍA; SÁNCHEZ, 2013a)	n.a. (10)	RS (CHOPRA; BILARE, 2018)
	GBM (CHOPRA; BILARE, 2018; CHEN <i>et al.</i> , 2020; KOKATE; CHETTY, 2021; ALEKSANDROVA, 2021; MARQUÉS; GARCÍA; SÁNCHEZ, 2013a) GBDT (FLOREZ-LOPEZ; RAMON-JERONIMO, 2015; BAO; LIANJU; YUE, 2019; LI <i>et al.</i> , 2022; PEARS; OETAMA, 2010) LightGBM (LI <i>et al.</i> , 2021) OD-LightGBM (LI <i>et al.</i> , 2022) GBMCI (BAI; ZHEN; SHEN, 2022) BRT (FITZPATRICK; MUES, 2016)		MV (MARTENS <i>et al.</i> , 2010; LIANG <i>et al.</i> , 2018; CHEN <i>et al.</i> , 2020; KOKATE; CHETTY, 2021)
	AdaBoost (WU <i>et al.</i> , 2021; CHOPRA; BILARE, 2018; BOZ <i>et al.</i> , 2018; DE CASTRO VIEIRA <i>et al.</i> , 2019; BOUGHACI; ALKHAWALDEH, 2017; LIANG <i>et al.</i> , 2018; CHEN <i>et al.</i> , 2020; CHOPRA; BILARE, 2018)		Stacking (LI; CHEN, 2022; LIANG <i>et al.</i> , 2018; ALEKSANDROVA, 2021)
	CoxBoost (BAI; ZHEN; SHEN, 2022)		
n.a. (9)	Bagging (CHOPRA; BILARE, 2018; DE CASTRO VIEIRA <i>et al.</i> , 2019; WU <i>et al.</i> , 2021; LIANG <i>et al.</i> , 2018; CHEN <i>et al.</i> , 2020; CHOPRA; BILARE, 2018)	Static Indirect (1)	HCES-Bag (HANDHIKA <i>et al.</i> , 2019)
	Stacking (WU <i>et al.</i> , 2021)	Static Direct (2)	SOM consensus model (BAO; LIANJU; YUE, 2019)
	RS (WU <i>et al.</i> , 2021)		
	RC (WU <i>et al.</i> , 2021)		

*Abreviações utilizadas: BRT = *Boosted Regression Trees*; CIF = *Canonical Interval Forest*; HCES = *Hill-Climbing Ensemble Selection*; HWA = *Hyper-parameter WA*; ORF = *Optimized RF*; RC = *Random Committee*; RS = *Random Subspace*; RSF = *Random Survival Forest*; SVDF = *Simple MV Decision Forest*; WA = *Weighted Averaging*; WVDF = *Weighted MV Decision Forest*.

Fonte: Artigos consultados na RSL

Representando o *benchmark* perfeito, tratando-se da primeira escolha dos setores de gestão de riscos financeiros, LR foi novamente o mais popular método estatístico. Similarmente aos CSMs para crédito corporativo, novamente os mais comuns classificadores de ML foram: NN, DT e SVM. Existe, contudo, uma preocupação com a popularidade de NNs, sendo que se trata dos mais difíceis de interpretar. Artigos sobre o crédito ao consumidor também demonstraram maior interesse com abordagens *ensemble*. Florez-Lopez e Ramon-Jeronimo (2015) fornecem provas empíricas de que metodologias *ensemble* são capazes de consistentemente superar a performance de modelos individuais em diferentes métricas de performance. Apesar de ainda em menor número do que modelos individuais, de todos os *ensembles*, a agregação por meio da metodologia *boosting* e RFs foram a vasta maioria. Todos os classificadores e seus respectivos tipos se encontram na Figura 3.7.

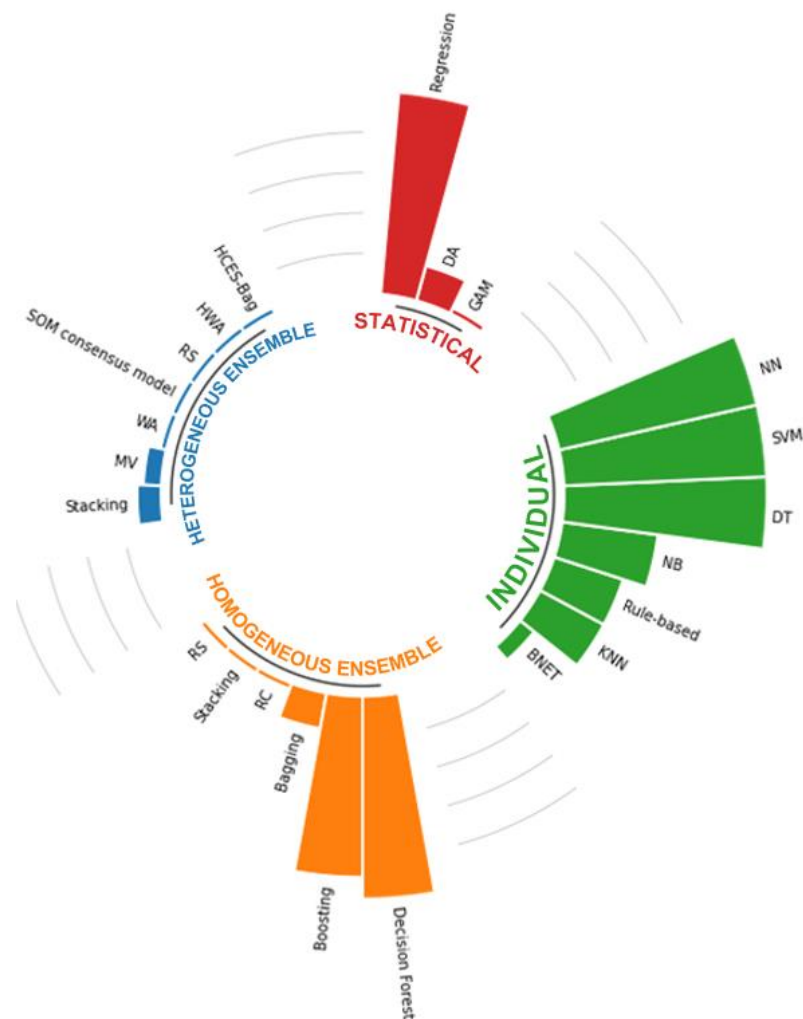


Figura 3.7 – Principais classificadores divididos por grupo
Fonte: Autoria própria (2023)

Considerando os requisitos regulatórios, e sabendo que algoritmos de *boosting* não são facilmente interpretados, RF parece ser a melhor abordagem *ensemble* para a construção de CSMs, tratando-se de um popular e relativamente simples algoritmo que não passa de uma combinação homogênea de DTs, agregadas por *bagging* e, por sua vez, mais facilmente interpretáveis.

3.3.1.2 Conjuntos de dados

Quanto aos conjuntos de dados ou *datasets* utilizados para o treinamento e avaliação dos CSMs propostos na literatura os resultados encontram-se na Figura 3.8.

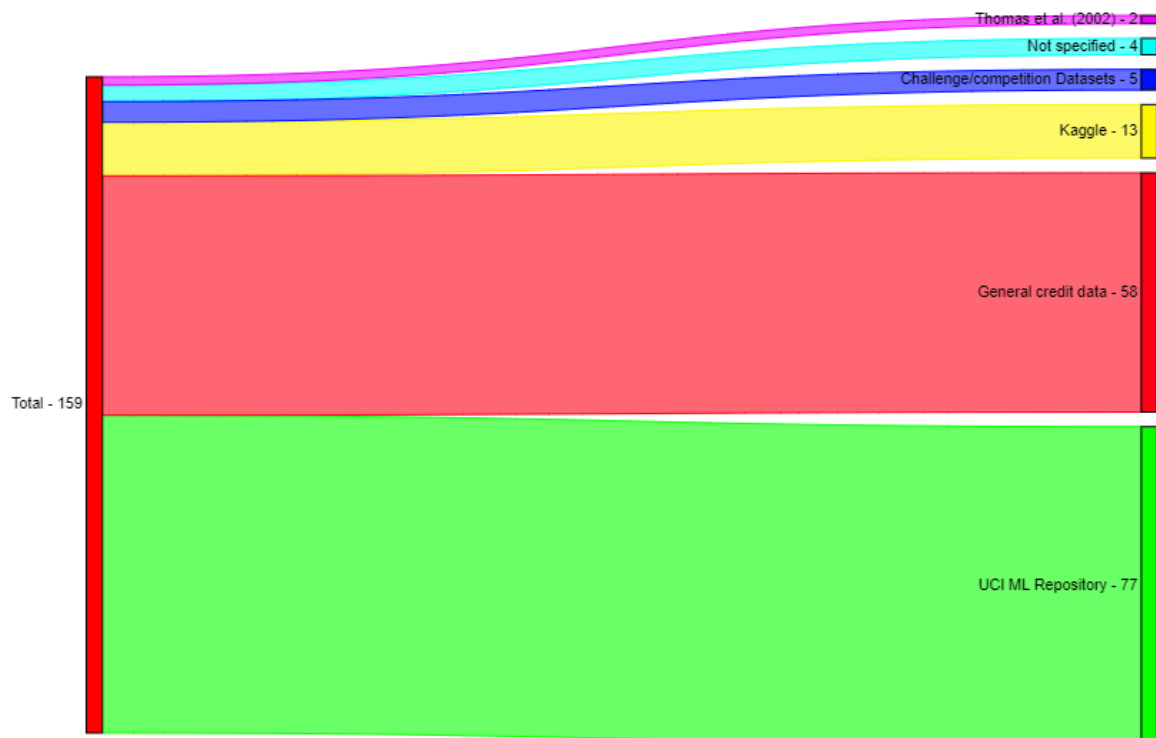


Figura 3.8 – Origem dos *datasets* empregados na literatura
Fonte: Autoria própria (2023)

De acordo com a Figura 3.8, apesar de muitos dos artigos analisados modelarem CSMs partindo de dados vindos diretamente de bancos e outras instituições privadas de concessão de crédito, o ramo de pesquisa abordado neste trabalho é beneficiado pelo fato de muitos autores optarem por usar os mesmos dados, geralmente disponíveis, pronta e gratuitamente, o que é essencial para o processo de *peer review*, além de fornecer incontáveis *benchmarks* diferentes.

De todos os 159 *datasets* relacionados, quase metade (48,43%) foram fornecidos pela *University of California Irvine* (UCI) e seu repositório de ML. Louzada, Ara e Fernandes (2016) mencionam um estudo que indica que cerca de 45% dos artigos sobre análise do risco de crédito observados levaram em conta os *datasets Australian, German* e/o *Japanese* (que serão explorados no Subcapítulo 4.1), provindos da UCI. Respectivamente, os *datasets German* (37,66% de todos *datasets* utilizados vindo do repositório de ML da UCI) (KENNEDY; NAMEE; DELANY, 2013; MARQUÉS; GARCÍA; SÁNCHEZ, 2013a; ZHANG *et al.*, 2022; SUN *et al.*, 2015; SARIEV; GERMANO, 2020; WAAD; GHAZI; MOHAMED, 2013; LIANG *et al.*, 2018; BOUGHACI; ALKHAWALDEH, 2017; HANDHIKA *et al.*, 2019; CHEN; XIANG, 2017; FLOREZ-LOPEZ; RAMON-JERONIMO, 2015; TRIVEDI, 2020; GIRI *et al.*, 2021; SHI; ZHANG; QIU, 2013; AGHAEIRAD; CHEN; RIBEIRO, 2017; BAO; LIANJU; YUE, 2019; SIAMI; GHOLAMIAN; BASIRI, 2014; ZHOU *et al.*, 2021; CHOI; KOO; PARK, 2015; CHEN *et al.*, 2020; TRIPATHI *et al.*, 2020; PEARS; OETAMA, 2010; WENG; HUANG; CHENG-KUI, 2021; LIU; FU; LIN, 2010; ARMAKI *et al.*, 2017; MARTENS *et al.*, 2010), *Australian* (33,77%) (KENNEDY; NAMEE; DELANY, 2013; MARQUÉS; GARCÍA; SÁNCHEZ, 2013a; ZHANG *et al.*, 2022; WAAD; GHAZI; MOHAMED, 2013; LIANG *et al.*, 2018; BOUGHACI; ALKHAWALDEH, 2017; DUMITRESCU *et al.*, 2022; PŁAWIAK; ABDAR; ACHARYA, 2019; ANTONAKIS; SFAKIANAKIS, 2009; GIRI *et al.*, 2021; SHI; ZHANG; QIU, 2013; AGHAEIRAD; CHEN; RIBEIRO, 2017; BAO; LIANJU; YUE, 2019; SIAMI; GHOLAMIAN; BASIRI, 2014; ZHOU *et al.*, 2021; CHEN *et al.*, 2020; TRIPATHI *et al.*, 2020; PEARS; OETAMA, 2010; WENG; HUANG; CHENG-KUI, 2021; TORKEVAR; GAME, 2019; LIU; FU; LIN, 2010; ARMAKI *et al.*, 2017; KHASHMAN, 2009) e *Japanese* (15,58%) (KENNEDY; NAMEE; DELANY, 2013; MARQUÉS; GARCÍA; SÁNCHEZ, 2013a; AGHAEIRAD; CHEN; RIBEIRO, 2017; BOUGHACI; ALKHAWALDEH, 2017; ZHOU *et al.*, 2021; WANG; WANG; LAI, 2005; CHEN *et al.*, 2020; TRIPATHI *et al.*, 2020; ARMAKI *et al.*, 2017) foram as escolhas mais comuns. A *Kaggle*, uma comunidade *online* para cientistas de dados, também foi uma escolha popular para a obtenção de dados, sendo também comumente mencionada.

Em média, foram utilizados 2,41 *datasets* por artigo. Lessmann *et al.* (2015), que encontraram uma média ainda menor, de 1,9 *datasets*, em seu estudo de *benchmark*, argumentam que o uso de múltiplos *datasets* é necessário para examinar a robustez de um CSM em relação às condições ambientais. Sabendo que alguns estudos que testaram um número excessivo de *datasets* elevam a média, este pequeno número se torna ainda mais problemático. De fato, mais da metade dos artigos utilizam apenas um conjunto de dados. Em relação ao

número de amostras e atributos por conjunto, observa-se uma média de 83.888 amostras e 92 atributos. No entanto, ambas as médias também são enganosas devido a alguns valores atípicos de dimensão muito elevada. Dados de crédito reais são tipicamente grandes e de alta dimensionalidade, e os *datasets* utilizados na comparação de classificadores devem ser semelhantes, garantindo a validade externa dos resultados empíricos (FINLAY, 2011; HAND, 2006). Isto também preocupa, pois, embora poucos estudos utilizem *datasets* reais com milhões de amostras e centenas de atributos, os *datasets* mais populares (UCI *Australian, German e Japanese*) contém apenas de 690 a 1000 amostras, e 14 a 20 atributos.

Metade dos artigos também listou o tamanho de seus subconjuntos de treinamento e de teste. Um modelo é, em média, treinado com 68,3% de todos os dados e testado com 28,2%. Lessmann *et al.* (2015) sugerem outra divisão, incluindo um subconjunto de avaliação, mas muito poucos autores adotaram esta metodologia (*e.g.* CHOPRA; BILARE, 2018; WANG; XU; ZHOU, 2015; KENNEDY; NAMEE; DELANY, 2013; FITZPATRICK; MUES, 2016). Malhotra e Malhotra (2000) afirmam que não existem diretrizes claras para dividir um *dataset* em subconjuntos de treinamento e teste. Enquanto alguns autores defendem uma divisão de 60-40, outros preferem 75-25. Houve casos de divisões aleatórias (SHI; ZHANG; QIU, 2013) e até abordagens de *bootstrapping* (FLOREZ-LOPEZ; RAMON-JERONIMO, 2015). Alguns dos artigos até usam tamanhos variados para cada *dataset* (KHASHMAN, 2009; AGHAEIRAD; CHEN; RIBEIRO, 2017).

Zhou *et al.* (2021) não apenas avaliam como os classificadores de referência se saem com diferentes dimensões de dados, mas também analisam esses classificadores para avaliar o impacto dos métodos FS. Enquanto Aghaeirad, Chen e Ribeiro (2017), por exemplo, relatam resultados melhores para subconjuntos de treinamento maiores, Khashman (2009) afirma que uma alta proporção de dados de treinamento em relação aos dados de teste não fornece dados significativos, o que significa que este nem sempre será o caso.

3.3.1.3 Pré-processamento de Dados

O pré-processamento de dados é uma etapa indispensável na classificação por meio de algoritmos de ML. Na fase de preparação, antes do desenvolvimento do modelo, é importante lidar corretamente com os dados brutos, para que possam ser adequadamente utilizados para treinar o classificador. Primeiramente, dados brutos podem conter muitas informações irrelevantes ou faltantes. O processo de lidar com partes ruidosas e faltantes dos dados brutos é conhecido como limpeza de dados (*data cleaning*). Felizmente, enquanto os dados faltantes

podem ser removidos ou substituídos diretamente, abordagens de ML supervisionado (por exemplo, NN ou KNN) e não supervisionadas (*e.g. clustering*) podem ser úteis para lidar com dados ruidosos, eliminando amostras mal rotuladas e/ou *outliers*, e até mesmo balanceando conjuntos de dados (conforme discutido no Subcapítulo 2.5 em relação ao desbalanceamento de classes).

Quando se trata de algoritmos de *data mining*, muitas vezes é necessário transformar os dados para torná-los adequados ao processo de aprendizado. A transformação de dados (*data transformation*) inclui redimensionamento, normalização (escalonamento dos valores dos dados dentro de uma faixa pré-definida), fusão de atributos e discretização (substituição de valores numéricos brutos por intervalos ou níveis conceituais). Por fim, técnicas de redução de dimensionalidade (*data reduction*) são essenciais quando se lida com grandes quantidades de dados. Estas técnicas visam reduzir os custos de armazenamento e análise de dados por meio de mecanismos de codificação e compressão – armazenando um modelo dos dados em vez de todos os dados – e por meio da seleção de um subconjunto de atributos (FS realizado no nível de dados).

O pré-processamento dos dados brutos também pode evitar problemas como *overfitting* e baixa generalização. Além disso, a interpretabilidade do modelo pode ser melhorada por meio da redução da dimensionalidade e remoção de ruídos. Portanto, para garantir melhores resultados, é importante considerar cuidadosamente as etapas de pré-processamento necessárias e aplicá-las de acordo com os requisitos do problema em questão.

Métodos de pré-processamento de dados foram abordados em um total de 35,82% de todos os artigos. No entanto, isso não significa que a maioria dos modelos não tenha sido preparada adequadamente, uma vez que alguns desses métodos são tão comuns que muitas vezes não são mencionados. Além disso, métodos de *resampling* e FS podem ser realizados no nível de dados. Entretanto, neste estudo, dada sua importância para o desenvolvimento do CSM, ambos são explorados separadamente. De acordo com a Figura 3.9, a maioria dos métodos foi para *data cleaning*, seguidos por métodos de *data transformation* e *data reduction*, respectivamente. A Tabela 3.6 aborda as etapas específicas adotadas.

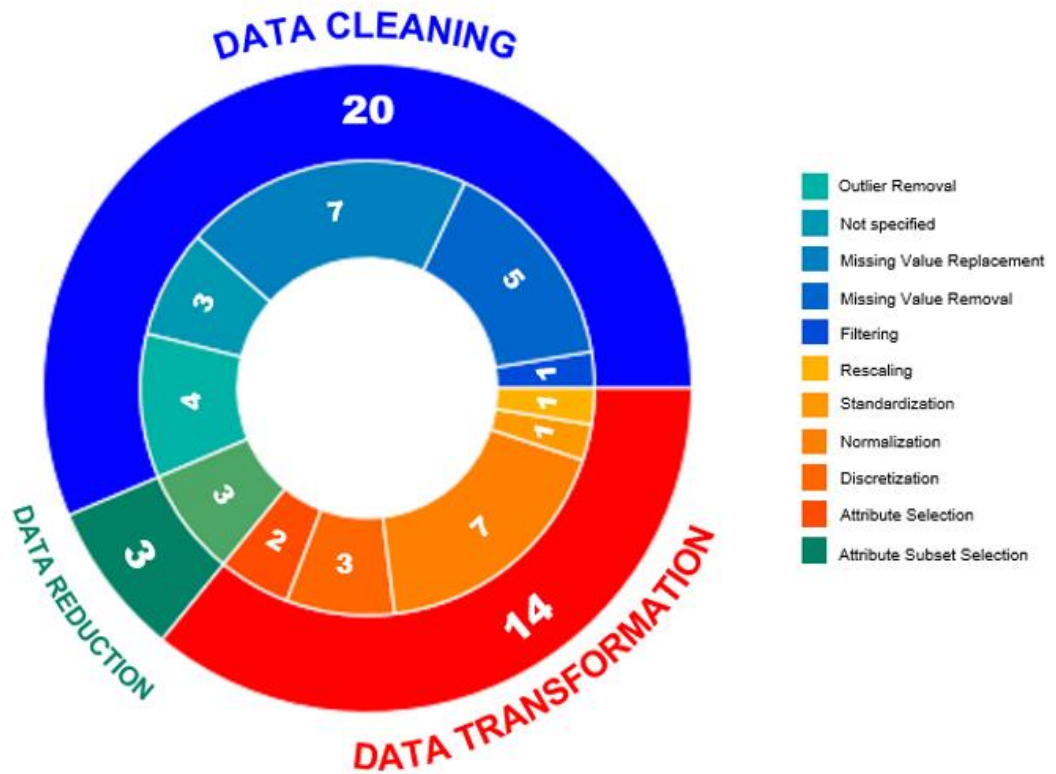


Figura 3.9 – Tipos de pré-processamento encontrados na literatura
Fonte: Autoria própria (2023)

Quadro 3.6 – Etapas de pré-processamento encontradas na literatura

	Etapa	Autor
DATA CLEANING	<i>Missing Value Removal</i>	Aleksandrova (2021); AMPOUNTOLAS <i>et al.</i> (2021); Armaki <i>et al.</i> (2017); Bao, Lianju e Yue (2019)
	<i>Missing Value Replacement</i>	Bao, Lianju e Yue (2019); Chen <i>et al.</i> (2020); Djeundje <i>et al.</i> (2021); Kokate e Chetty (2021); Li <i>et al.</i> (2022); Sariev e Germano (2020); Wang <i>et al.</i> (2019)
	Remoção de outliers	Aleksandrova (2021); Armaki <i>et al.</i> (2017); Bao, Lianju e Yue (2019); Munkhdalai <i>et al.</i> (2019)
	<i>Filtering</i>	Djeundje <i>et al.</i> (2021)
	Não especificado	Ariza-Garzón (2020); Pan (2021); Weng, Huang e Cheng-Kui (2021)
DATA TRANSFORMATION	<i>Attribute Selection</i>	Djeundje <i>et al.</i> (2021); Fenerich <i>et al.</i> (2020)
	Discretização	Sariev e Germano (2020); Ziemba <i>et al.</i> (2021)
	Normalização	Bao, Lianju e Yue (2019); Chen <i>et al.</i> (2020); Kennedy, Namee e Delany (2013); KHASHMAN (2009); Li <i>et al.</i> (2022); Pan (2021); Zhou <i>et al.</i> (2021)
	<i>Rescaling</i>	Łławiak, Abdar e Acharya (2019)
	<i>Standardization</i>	Łławiak, Abdar e Acharya (2019)
DATA REDUCTION	<i>Attribute Subset Selection</i>	Bao, Lianju and Yue (2019); Chen <i>et al.</i> (2020); Li e Chen (2021); AMPOUNTOLAS <i>et al.</i> (2021)

Fonte: Artigos consultados na RSL

No caso de *data cleaning*, a substituição de valores ausentes muitas vezes foi feita substituindo os dados faltantes pelo valor médio do atributo correspondente (e.g. WANG *et al.*, 2020; KOKATE; CHETTY, 2021). *Outliers* foram removidos de diferentes maneiras, incluindo o uso de KNN (ARMAKI *et al.*, 2017) e remoção de *outliers* por desvio padrão (e.g. MUNKHDALAI *et al.*, 2019). Djeundje *et al.* (2021) também filtraram dados com variância negligenciável. Quanto à transformação de dados, na maioria dos casos, os dados foram normalizados usando o método de normalização *min-max* (e.g. KENNEDY; NAMEE; DELANY, 2013; KHASHMAN, 2009; ZHOU *et al.*, 2021; CHEN *et al.*, 2020). A seleção de atributos foi feita pela fusão de variáveis pouco populadas (DJEUNDJE *et al.*, 2021) ou pela transformação de características em dados binários (FENERICH *et al.*, 2020). Exemplos de discretização incluem os critérios de discretização de Kononenko e Fayyad (ZIEMBA *et al.*, 2021).

Finalmente, a seleção de subconjuntos de atributos para *data reduction* foi feita removendo características altamente correlacionadas (CHEN *et al.*, 2020; AMPOUNTOLAS *et al.*, 2021), usando o algoritmo de aprendizado não supervisionado *k-Means Clustering* (BAO; LIANJU; YUE, 2019) e selecionando atributos de acordo com os níveis de avaliação de risco predefinidos de um banco (LI; CHEN, 2022). Essas técnicas específicas para *data reduction* são consideradas métodos de FS não supervisionados, portanto, dado o enfoque em aprendizado supervisionado, é preferível considerá-las como parte da etapa de pré-processamento de dados.

3.3.1.4 Métodos de *Feature Selection*

Quanto a FS, pouco mais de um terço (37,31%) da literatura que compõe esta RSL mencionam uma etapa de FS. Os métodos encontrados e seus respectivos artigos encontram-se no Quadro 3.7.

Quadro 3.7 – Métodos de FS encontrados na literatura

Autor	Método(s)
Zhang <i>et al.</i> (2022)	SFS/SBS/LASSO
Sun <i>et al.</i> (2015)	B&B/t-test/ <i>Factor Analysis</i>
Waad, Ghazi e Mohamed (2013)	MaxRel/mRMR/t-test/Kullback-Leibler/Relief/Info-Gain/Wrapper Bidirectional/SFS/SBS
Handhika <i>et al.</i> (2019)	MARS
Djeundje <i>et al.</i> (2021)	PCA
Papouskova e Hajek (2019)	ENORA
Moscatelli <i>et al.</i> (2020)	PCA/LDA/ISOMAP/Kernel PCA
Florez-Lopez e Ramon-Jeronimo (2015)	<i>Splitting rules</i>
Dumitrescu <i>et al.</i> (2022)	DTs/ALasso (embutidos no classificador)
Trivedi (2020)	Chi-square/Gain-Ratio/Info-Gain
Pławiak, Abdar e Acharya (2019)	GA

Munkhdalai <i>et al.</i> (2019)	TFFS/NAP
Bao, Lianju e Yue (2019)	RF
Siami, Gholamian e Basiri (2014)	Não especificado
Zhou <i>et al.</i> (2021)	LASSO/MARS
Li e Chen (2021)	RF
Fitzpatrick e Mues (2016)	Gain-Ratio/Info-Gain
Ziemba, Radomska-Zalas and Becker (2020)	SA/SU/SU-FCBF/CFS
Kokate and Chetty (2021)	Feature Importance
Fenerich <i>et al.</i> (2020)	Feature Importance/SFS
Weng, Huang e Cheng-Kui (2021)	Info-Gain
Liu, Fu e Lin (2010)	GA
Ziemba <i>et al.</i> (2021)	CFS/SU/FCBF/Significance Attribute/Relief

*Abreviações utilizadas: ALasso = *Adaptive LASSO*; B&B = *Branch & Bound*; MARS = *Multivariate Adaptive Regression Spline*; MaxRel = *Maximum Relevance*; mRMR = *Minimum Redundancy Maximum Relevance*; NAP = *RF based New Approach*; PCA = *Principal Component Analysis*; SU = *Symmetrical Uncertainty*; SU-FCBC = *SU-Fast Correlation-Based Filter*; TFSS = *Two-stage Filter FS*.

Fonte: Artigos consultados na RSL

Além de testar algoritmos de FS tradicionais, alguns dos artigos se concentram no desenvolvimento de novos algoritmos de FS. Neste contexto, Huang, Wang e Wang (2018) desenvolveram um método híbrido chamado *Bisection Method based on Tabu Search* (BMTS), observando um aumento geral no desempenho em todos os *datasets* testados. Sun *et al.* (2015) também introduziram o BBHFS (B&B Based *Hybrid Feature Selection*), que também apresentou resultados estáveis e satisfatórios. Além disso, Waad, Ghazi e Mohamed (2013) experimentaram uma complexa fusão de três estágios para FS usando programação quadrática. Uma abordagem híbrida adicional é proposta por Boughaci e Alkhalwaldeh (2017) com *Collaboration Learning and Classification Based on Agents* (AGENTS), uma combinação de métodos *filter*, *wrapper* e *Principal Component Analysis* (PCA), que fornece uma análise de dados mais rápida e flexível. Por fim, Zhang *et al.* (2022) tentam selecionar um subconjunto ótimo de características por meio de programação 0-1.

Os resultados podem variar amplamente quando se trata do impacto que FS tem sobre o desempenho preditivo. Em estudos comparativos, como o de Sun *et al.* (2015), evidências mostram que diferentes algoritmos de FS podem melhorar, ter pouco impacto ou até mesmo diminuir as métricas de desempenho. O consenso, no entanto, é que os algoritmos de seleção de características podem melhorar significativamente os resultados de classificação (*e.g.* HUANG; WANG; WANG, 2018, WAAD; GHAZI; MOHAMED, 2013; PAPOUSKPVA; HAJEK, 2019). Zhou *et al.* (2021) estudam quais classificadores são mais sensíveis aos métodos de FS. Interessantemente, Zhang *et al.* (2022) conduzem um teste de hipótese usando o coeficiente de Gini para avaliar o impacto do número de atributos, concluindo que o poder discriminatório de um subconjunto de variáveis é independente do número de características

nele (refutando a ideia de que quanto mais atributos, melhor). À luz disso, apenas pelo seu potencial de melhoria, FS se mostra uma etapa muito importante na modelagem de risco.

Desconsiderando métodos *novel*, de acordo com a Figura 3.10, simples algoritmos *filter* são muito populares, representando cerca de 53% (em 27 menções) dos métodos utilizados. Considerando FS por meio de métodos *filter* no contexto do desenvolvimento de CSMs, Radomska-Zalas e Becker (2020) conduziram um estudo comparativo de múltiplos algoritmos *filter* para FS, concluindo que *correlation-based* FS (CFS) foi a abordagem de melhor performance (apesar de ainda provavelmente incapaz de superar algoritmos *wrapper* e *embedded*).

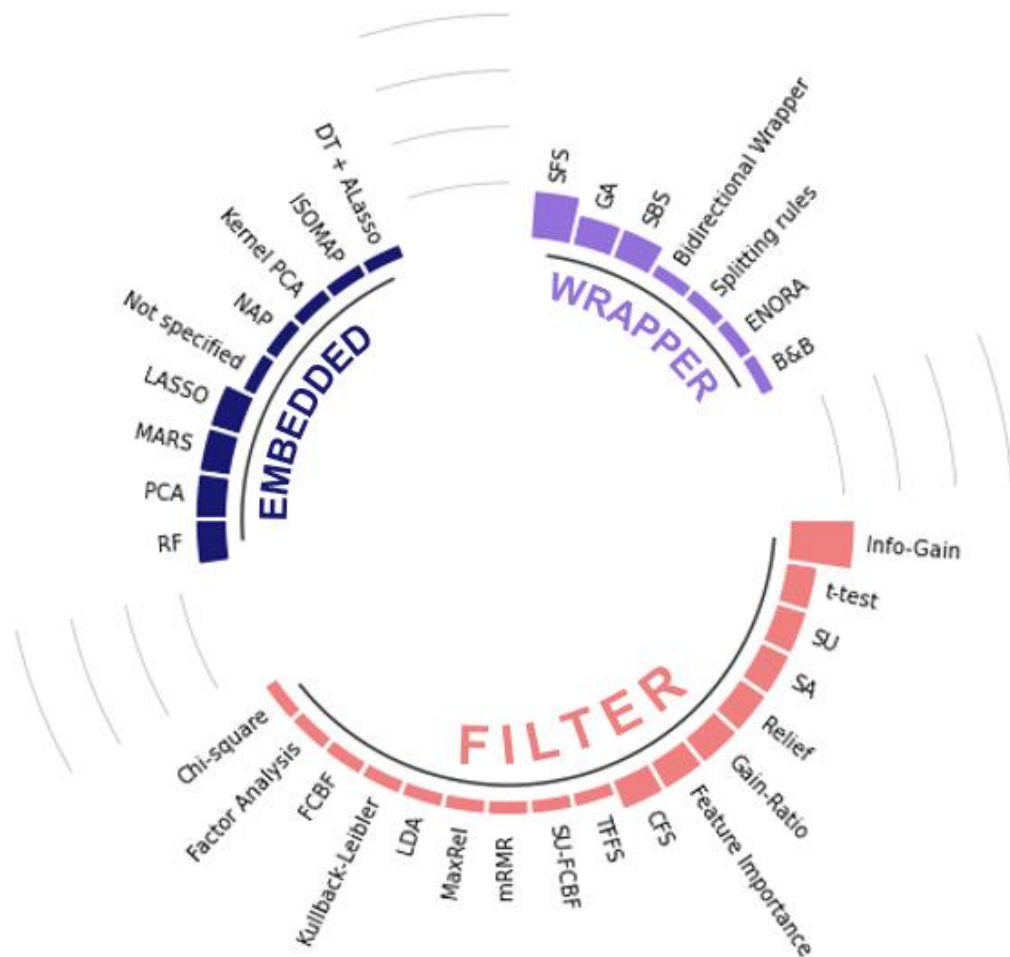


Figura 3.10 – Métodos de FS encontrados na literatura
Fonte: Autoria própria (2023)

3.3.1.5 Métodos para o Balanceamento de Classes

Vinte e um (21) dos artigos analisados (31,34% do total) buscam lidar com o problema do desbalanceamento de classes. Wang, Xu e Zhou (2015) e Sun *et al.* (2015) inclusive, propõem novas formas de fazê-lo, respectivamente, por meio do algoritmo *Imbalance Oriented Multiple-Classifer Ensemble* (IOMCE), e por técnicas de *clustering*.

Contudo, a maior parte dos autores utiliza diferentes e populares algoritmos já existentes para a classificação em *datasets* desbalanceados. Os algoritmos utilizados para lidar com o desbalanceamento de classes encontrados nesta RSL encontram-se no Quadro 3.8:

Quadro 3.8 – Métodos de balanceamento de classes encontrados na literatura

Autor	Algoritmo	Tipo
Kennedy, Namee e Delany (2013)	Não especificado	<i>Data-level</i>
Marqués, García e Sánchez (2013a)	ROS, RUS, OSS, NCL, SMOTE, WE e SL-SMOTE	<i>Data-level</i>
Marqués, García e Sánchez (2013a)	<i>Sampling Base Clustering</i> (SBC)	<i>Training set partition</i>
Sariev e Germano (2020)	-	<i>Training set partition</i>
Moscato, Picariello and Sperlí (2021)	ROS/RUS/IHT/SMOTE/ADASYN/SmoteToken/SmoteEnn	<i>Data-level</i>
Djeundje <i>et al.</i> (2021)	Não especificado	<i>Data-level</i>
Papouskova e Hajek (2019)	EUSBoost/SMOTEBagging	<i>Data-level</i>
De Castro Vieira <i>et al.</i> (2019)	Não especificado	<i>Data-level</i>
Mancisidor <i>et al.</i> (2020)	Não especificado	<i>Data-level</i>
AMPOUNTOLAS <i>et al.</i> (2021)	SMOTENC	<i>Data-level</i>
Boz <i>et al.</i> 2018	-	<i>Incorporated in the classifier</i>
Boz <i>et al.</i> 2018	-	<i>Training set partition</i>
Khemakem, Bem Said and Boujelbene (2018)	SMOTE/ROS	<i>Data-level</i>
Zhou <i>et al.</i> (2021)	SMOTE	<i>Data-level</i>
Ariza-Garzón (2020)	Not specified	<i>Data-level</i>
Kokate e Chetty (2021)	SMOTE	<i>Data-level</i>
Hamdoun e Rguibi (2019)	<i>Stratified sampling</i>	<i>Training set partition</i>
Fenerich <i>et al.</i> (2020)	RUS	<i>Data-level</i>
Ziemba <i>et al.</i> (2021)	RUS/SMOTE	<i>Data-level</i>
Marqués, García e Sánchez (2013a)	SMOTE	<i>Data-level</i>
Li <i>et al.</i> (2021)	-	Incorporado no classificador

Fonte: Artigos consultados na RSL

É possível observar que a maior parte destes algoritmos é aplicado no *data-level*, o que significa que podem ser incorporados no estágio de pré-processamento de dados. Como discutido, este tipo de abordagem envolve *oversampling* ou *undersampling*. A Figura 3.11 divide estes métodos de acordo, ilustrando quais são os mais utilizados.

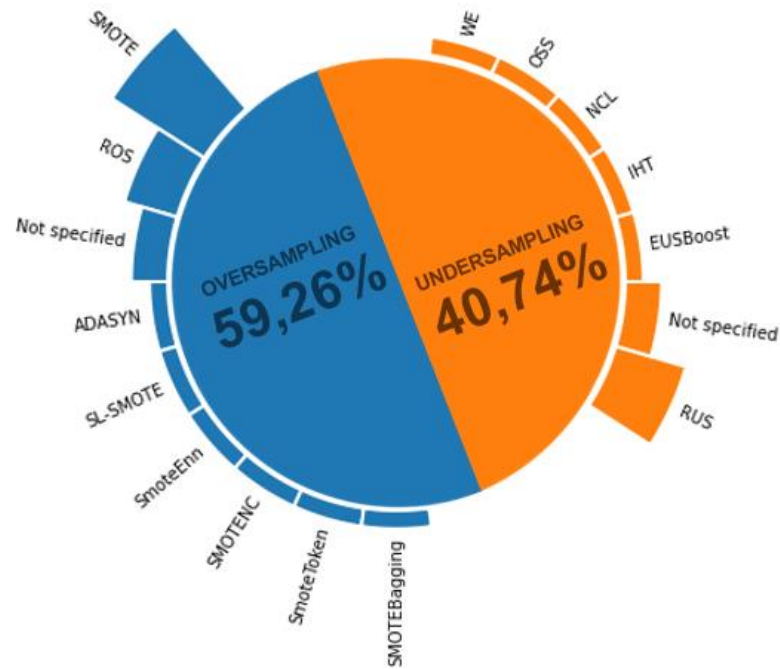


Figura 3.11 – Algoritmos de *resampling* encontrados na literatura
 Fonte: Autoria própria (2023)

Como esperado, sabendo que existem estudos evidenciando que *oversampling* costuma ter melhor performance do que *undersampling* (MARQUÉS; GARCÍA; SANCHÉZ, 2013a), a maior parte dos algoritmos de *resampling* encontrados eram algoritmos de *oversampling*. SMOTE (MOSCATO; PICARIELLO; SPERLÍ, 2021; KHEMAKEM; BEM SAID; BOUJELBENE, 2018; ZHOU *et al.*, 2021; KOKATE; CHETTY, 2021; ZIEMBA *et al.*, 2021; MARQUÉS; GARCÍA; SÁNCHEZ, 2013a) foi consideravelmente o algoritmo mais popular, especialmente contabilizando também suas variações (35,43% das menções) como: *Safe-Level* SMOTE (SL-SMOTE) (MARQUÉS; GARCÍA; SÁNCHEZ, 2013a), SmoteEnn (MOSCATO; PICARIELLO; SPERLÍ, 2021), SMOTENC (AMPOUNTOLAS *et al.*, 2021), SmoteToken (MOSCATO; PICARIELLO; SPERLÍ, 2021) e SMOTEBagging (PAPOUSKOVA; HAJEK, 2019). Também foram encontradas instâncias de ROS (MARQUÉS; GARCÍA; SÁNCHEZ, 2013a; MOSCATO; PICARIELLO; SPERLÍ, 2021; KHEMAKEM; BEM SAID; BOUJELBENE, 2018) e ADASYN (MOSCATO; PICARIELLO; SPERLÍ, 2021), ambos algoritmos de *oversampling*.

Técnicas de *undersampling* incluíram RUS (MARQUÉS; GARCÍA; SÁNCHEZ, 2013a; MOSCATO; PICARIELLO; SPERLÍ, 2021; FENERICH *et al.*, 2020; ZIEMBA *et al.*,

2021), *One Sided Selection technique* (OSS) (MARQUÉS; GARCÍA; SÁNCHEZ, 2013a), EUSBoost (PAPOUSKOVA; HAJEK, 2019), *Instance Hardness Threshold* (IHT) (MOSCATO; PICARIELLO; SPERLÍ, 2021), *Neighborhood Cleaning Rule* (NCL) (MARQUÉS; GARCÍA; SÁNCHEZ, 2013a) *Wilson's Editing algorithm* (WE) ou *Edited Nearest Neighbor* (ENN) (MARQUÉS; GARCÍA; SÁNCHEZ, 2013a).

Em geral, resultados para modelos pareados com etapas de *resampling* foram positivos, variando de pequenas melhoras (*e.g.* DJEUNDJE *et al.*, 2021; SUN *et al.*, 2015) até resultados substancialmente melhores (PAPOUSKOVA; HAJEK, 2019; DE CASTRO VIEIRA *et al.*, 2019; KHEMAKEM; BEM SAID; BOUJELBENE, 2018). Kennedy, Namee e Delany (2013) descrevem como a performance de um modelo gradualmente decresce conforme aumenta o desbalanceamento entre as classes. Usando testes *post-hoc* de Friedman e Nemenyi, Marqués, García e Sánchez (2013a) provam que métodos de *resampling* são efetivos para lidar com *datasets* desbalanceados. Papouskova e Hajek (2019), por fim, concluem que o balanceamento das classes tem que ser ponderado na modelagem da métrica PD.

3.3.1.6 Métodos de Otimização de Hiper Parâmetros

Vinte e um (21) artigos (31,43% do total) descrevem a utilização de alguma metodologia de OHP na modelagem do risco de crédito ao consumidor. Os métodos encontrados se encontram na Figura 3.12:

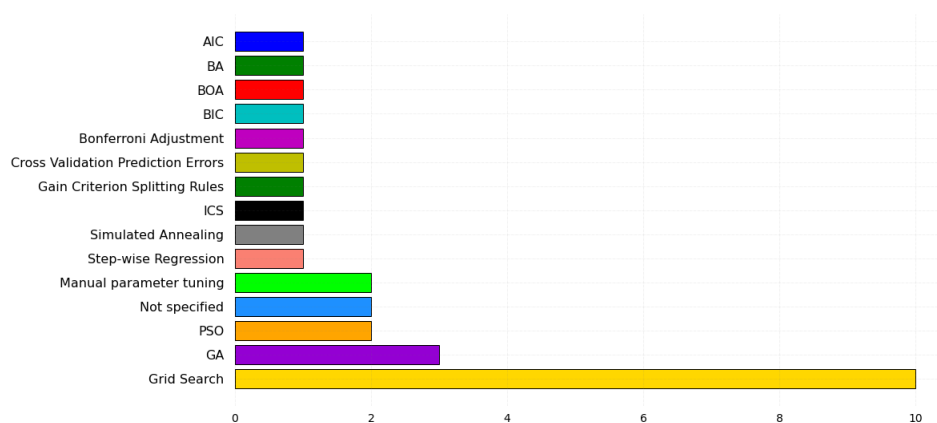


Figura 3.12 – Algoritmos de OHP encontrados na literatura
Fonte: Autoria própria (2023)

Apesar de grande parte dos autores provavelmente terem experimentado com alguns hiper parâmetros diferentes, poucos descrevem procedimentos de busca manual para aumento de performance (*e.g.* KENNEDY; NAMEE; DELANY, 2013; AGHAEIRAD; CHEN;

RIBEIRO, 2017). Similarmente aos métodos *filters* representando a maior parte das técnicas de FS, provavelmente em decorrência de sua relativa simplicidade é que *grid search* se destaca, sendo o mais comum método encontrado para OHP (WANG *et al.*, 2019; ZHENG, 2019; BAO; LIANJU; YUE, 2019; PEARS; OETAMA, 2010; ALEKSANDROVA, 2021), usualmente pareado com *10-fold cross-validation* (BAI; ZHEN; SHEN, 2022; DUMITRESCU *et al.*, 2022; MUNKHDALAI *et al.*, 2019; ZHANG *et al.*, 2018; FITZPATRICK; MUES, 2016).

Outros métodos encontrados foram: *Akaike Information Criterion* (AIC) (CHEN; XIANG, 2017), *Bat Algorithm* (BA) (TRIPATHI *et al.*, 2020), *Bayesian Hyper-parameter Optimization* (BOA) (WANG *et al.*, 2019), *Bayesian Information Criterion* (BIC) (CHEN; XIANG, 2017), *Bonferroni Adjustment* (FLOREZ-LOPEZ; RAMON-JERONIMO, 2015), *Gain Criterion* (FLOREZ-LOPEZ; RAMON-JERONIMO, 2015), *Improved Cuckoo Search* (ICS) (TRIPATHI *et al.*, 2020), *Simulated Annealing* (ZHENG, 2019), *Step-wise Regression* (FLOREZ-LOPEZ; RAMON-JERONIMO, 2015), *Particle Swarm Optimization* (PSO) (ARMAKI *et al.*, 2017; TRIPATHI *et al.*, 2020) e GA (ARIZA-GARZÓN, 2020; LIU; FU; LIN, 2010).

Uma observação interessante é que algoritmos sofisticados de OHP podem ser mais eficientes do que uma *grid search* (ZHENG, 2019). SA, por exemplo, o faz por meio de uma complexa análise combinatória. Todos os métodos de OHP encontrados na literatura e seus artigos correspondentes encontram-se no Quadro 3.9.

Quadro 3.9 – Métodos de OHP encontrados na literatura

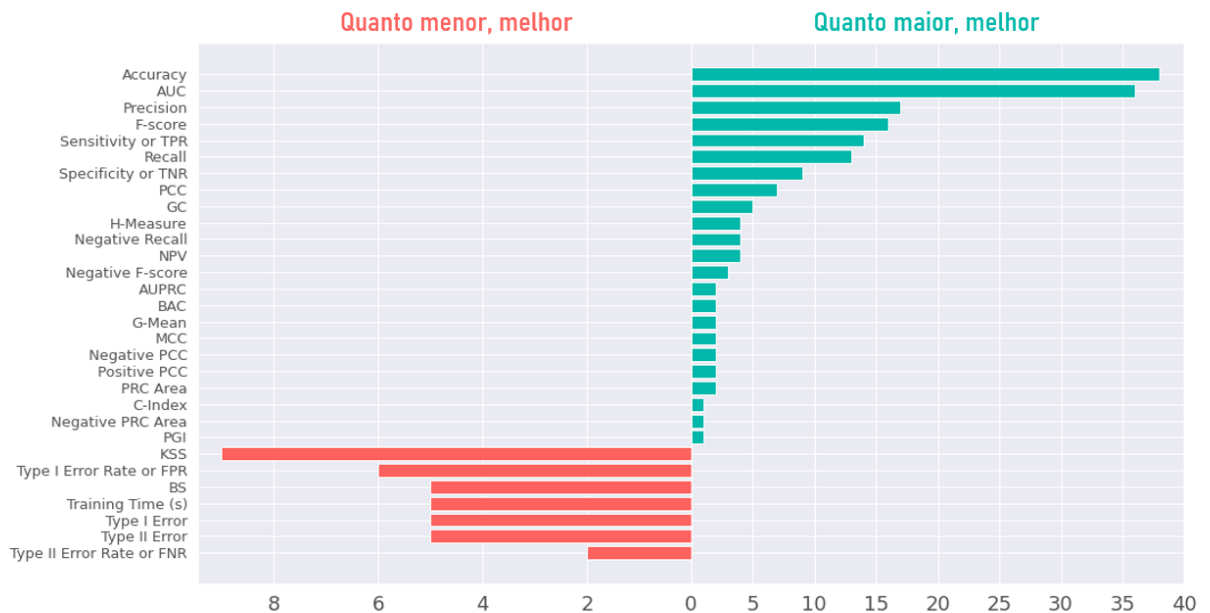
Autor	Método
Kennedy, Namee e Delany (2013)	Busca manual
Bai, Zhen e Shen (2022)	<i>Grid search</i> (c/ <i>5-fold cross-validation</i>)
De Castro Vieira <i>et al.</i> (2019)	Não especificado
Wang <i>et al.</i> (2019)	BOA/ <i>Grid search</i>
Chen e Xiang (2017)	AIC BIC <i>Cross-validation prediction errors</i>
Florez-Lopez e Ramon-Jeronimo (2015)	<i>Step-wise Regression/Bonferroni Adjustment/Gain Criterion Splitting Rules</i>
Dumitrescu <i>et al.</i> (2022)	<i>Grid search</i> (c/ <i>10-fold cross-validation</i>)
Plawiak, Abdar e Acharya (2019)	GA
Munkhdalai <i>et al.</i> (2019)	<i>Grid search</i> (c/ <i>10-fold cross-validation</i>)
Zheng (2019)	SA/ <i>Grid search</i>
Aghaeirad, Chen e Ribeiro (2017)	Busca manual
Bao, Lianju e Yue (2019)	<i>Grid search</i>
Li e Chen (2021)	Não especificado
Ariza-Garzón (2020)	GA
Zhang <i>et al.</i> (2018)	<i>Grid search</i> (c/ <i>cross-validation</i>)
Fitzpatrick e Mues (2016)	<i>Grid search</i> (c/ <i>10-fold cross-validation</i>)
Pears e Oetama (2010)	<i>Grid search</i>
Liu, Fu e Lin (2010)	GA
Armaki <i>et al.</i> (2017)	PSO
Aleksandrova (2021)	<i>Grid search</i>
Tripathi <i>et al.</i> (2020)	PSO/ICS/BA

Fonte: Artigos consultados na RSL

3.3.1.7 Métricas de Performance

Quando se trata de classificação, existem muitas diferentes métricas de performance, que podem ser utilizadas na aferição do desempenho de um modelo. As medidas mais comuns para avaliar a performance de CSMs na literatura referente ao risco de crédito ao consumidor encontram-se na Figura 3.13.

Para uma avaliação objetiva do desempenho de um modelo, devem ser escolhidas as métricas de performance adequadas (CHEN, 2021). Também é importante entendê-las e interpretá-las corretamente. O fato de que a precisão foi a métrica mais comumente encontrada é preocupante, pois, segundo Aslam *et al.* (2018), não leva em conta as repercussões do Erro Tipo II, que podem, neste contexto, ser devastadoras para as instituições de crédito.



* Abreviações utilizadas: AUC = *Area Under the Receiver Operating Characteristics Curve*; TPR = *True Positive Rate*; TNR = *True Negative Rate*; PCC = *Percentage of Correctly Classified Cases*; GC = *Gini Coefficient*; NPV = *Negative Predictive Value*; AUPRC = *Area Under the Precision-Recall Curve*; BAC = *Balanced Accuracy*; MCC = *Matthew's Correlation Coefficient*; PGI = *Partial Gini Index*; KSS = *Kolmogorov-Smirnov Statistic*; FPR = *False Positive Rate*; BS = *Brier Score*; FNR = *False Negative Rate*.

Figura 3.13 – Métricas de performance mais utilizadas na literatura

Fonte: Autoria própria (2023)

Na classificação binária, existem dois tipos de classe: positiva e negativa. Quando um classificador atribui uma classe a uma nova amostra, surgem quatro resultados possíveis. Se for uma amostra positiva, o classificador pode classificá-la corretamente como positiva (verdadeiro

positivo ou *true positive* - TP) ou classificá-la erroneamente como negativa (falso negativo ou *false negative* - FN). Se for uma amostra negativa, ela também será classificada corretamente (verdadeiro negativo ou *true negative* - TN) ou incorretamente (falso negativo ou *false negative* - FN). Estas quatro situações compõem a matriz de confusão (Quadro 3.10), que por si só é uma boa ferramenta para avaliação do modelo.

Quadro 3.10 – Matriz de Confusão

	Positivo (Y=1)	Negativo (Y = 0)
Classificado como positivo ($h_i = 1$)	TP	FP
Classificado como negativo ($h_i = 0$)	FN	TN

Fonte: Autoria própria (2023)

Todas as métricas de performance levam em consideração componentes da matriz de confusão (por isso os nomes “positivo” e “negativo” ao lado de algumas delas na Figura 3.13). As três medidas mais encontradas são bons exemplos disso.

Instâncias de FP e FN podem ser chamadas também, respectivamente, de Erro Tipo I e Erro Tipo II. Estes erros de classificação caracterizam, por si só, métricas de performance em sistemas de CS, representando, nesta ordem, o número de amostras em que um bom pagador foi equivocadamente classificado como mau, e um mau pagador foi equivocadamente classificado como bom. Enquanto o Erro Tipo I reduz a lucratividade de uma instituição financeira, o Erro Tipo II, ao viabilizar a concessão de crédito para clientes que se tornarão inadimplentes, ativamente causam perdas, motivo pelo qual pesquisas costumam preferi-lo ao Erro Tipo I na avaliação de CSMs.

A matriz de confusão, por sua vez, tem um papel ainda maior no cálculo das métricas de performance mais utilizadas para fins de CS, sendo que as circunstâncias previstas na matriz também ajudam na determinação de outras métricas mais sofisticadas. Em CSMs tradicionais, segundo Shen *et al.* (2020), *Area Under the Receiver Operating Characteristic Curve* (AUC) é uma forma efetiva para a avaliação de performance que, na prática, é muito utilizada com esta finalidade. Corresponde à área abaixo da *Receiver Operating Characteristic Curve* (ROC) (Figura 3.14) que, por sua vez, é uma curva baseada na taxa de falsos positivos (*false positive rate* ou FPR), definida na Eq. 3.1, plotada no eixo das abcissas e a taxa de verdadeiros positivos (*true positive rate* ou TPR) ou sensibilidade, definida pela Eq. 3.2, no eixo das ordenadas.

$$FPR = \frac{FP}{FP + TN} \quad (3.1)$$

$$TPR = \frac{TP}{TP + FN} \quad (3.2)$$

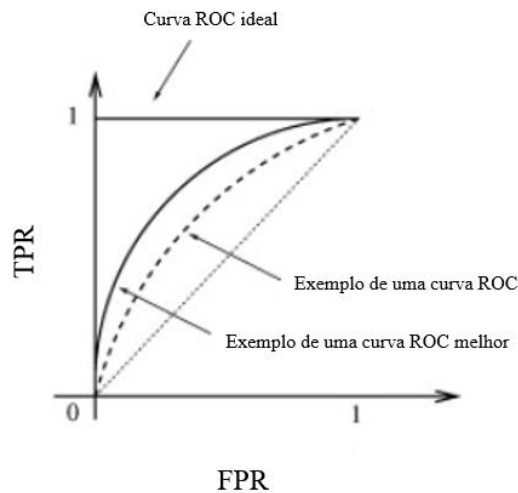


Figura 3.14– Exemplo de curvas ROC
 Fonte: Adaptado de Meyer-Baese e Schmid (2014)

A curva ROC ilustra graficamente o desempenho de um modelo de classificação binário na discriminação entre classes (TRUCCO *et al.*, 2019). Quanto mais ela se aproxima da curva ROC ideal, melhor é o desempenho do modelo. Portanto, quanto maior a área abaixo da curva (AUC), melhor é a performance do classificador (SHEN *et al.*, 2020), como ilustrado pelo Tabela 3.1.

Tabela 3.1 – Interpretações do valor de AUC

Valor de AUC	Interpretação
AUC = 0.5	Nenhuma discriminação (aleatório, <i>e.g.</i> jogar na moeda)
$0.5 < AUC \leq 0.6$	Discriminação ruim
$0.6 < AUC \leq 0.7$	Discriminação aceitável
$0.7 < AUC \leq 0.9$	Discriminação excelente
AUC > 0.9	Discriminação exemplar

Fonte: Adaptado de Yang e Berdine (2017)

A precisão de um modelo equivale ao percentual correto de previsões das amostras de uma classe. Pode ser calculada pelas Eqs. 3.3 e 3.4.

$$\text{Precisão (classe positiva)} = \frac{TP}{TP + FP} \quad (3.3)$$

$$\text{Precisão (classe negativa)} = \frac{TN}{TN + FN} \quad (3.4)$$

Para melhor ilustrar isto, como exemplo, quando um modelo preciso prevê que a classe de uma amostra será positiva, existe uma grande chance de a previsão estar correta, enquanto um modelo não preciso pode prever erroneamente muitas amostras como positivas.

Já a acurácia indica a proporção de exemplos ou instâncias corretamente classificados pelo modelo em relação ao total de exemplos avaliados. Calculada pela Eq. 3.5.

$$\text{Precisão (classe positiva)} = \frac{TP + TN}{TP + TN + FP + FN} \quad (3.5)$$

Precisão e acurácia, por si só, não são o suficiente para representar verdadeiramente o desempenho preditivo de um CSM, uma vez que não levam em conta os diferentes custos da classificação incorreta (Erro Tipo II, que é mais impactante, é equivalente a FN, ou seja, um cliente inadimplente é previsto como não inadimplente). Apesar de ser uma maneira muito utilizada e interessante de ilustrar o desempenho do modelo, a AUC também apresenta algumas deficiências além do argumento baseado nos diferentes custos de classificação, pois depende do classificador que está sendo avaliado. Os autores destacam que a distribuição dos custos de classificação deve depender do problema, e não do classificador. Uma das principais sugestões alternativas para a AUC é a *H-Measure* (LESSMANN *et al.*, 2015; MUNKHDALAI *et al.*, 2019; MANCISIDOR *et al.*, 2020; KENNEDY; NAMEE; DELANY, 2013; FITZPATRICK; MUES, 2016). A avaliação de desempenho também é frequentemente acompanhada por *cross-validation*, sendo *10-fold cross-validation* a técnica mais popular.

As 5 métricas de performance mais comuns para o melhor modelo dos três conjuntos de dados mais populares (*australian*, *german* e *japanese* do Repositório de ML da UCI) se encontram listadas nas Tabelas 3.2-3.4 (todas as métricas para todos os conjuntos de dados de crédito ao consumidor se encontram no Anexo A).

Tabela 3.2 – Resultados das 5 principais métricas na literatura para o *australian* UCI dataset

Autor	Melhor Modelo	Acurácia	AUC	TPR	Erro Tipo II ↓	F-score
Martens <i>et al.</i> (2007)****	SVM	0,857	-	-	-	-
Aghaeirad, Chen e Ribeiro (2017)	SOM+FNN	0,859	-	-	-	-
Abellan e Castellano (2017)*	Bag CDT	0,863	-	-	-	-
Wang <i>et al.</i> (2011)*	Bag DT	0,863	-	-	-	-
Xia <i>et al.</i> (2018)*	<i>Bagging+Stacking</i>	0,863	-	-	-	-
Peng <i>et al.</i> (2008)****	LibSVM	0,864	-	-	-	-
Luo, Cheng e Hsieh (2009)****	CLC	0,865	-	-	-	-
Chen e Li (2010)****	SVM (LDA FS)	0,865	-	-	-	-
Bao, Lianju e Yue (2019)	SVM	0,866	0,941	-	-	-
Hoffmann <i>et al.</i> (2007)****	NB	0,867	-	-	-	-
Zhang <i>et al.</i> (2022)	LR	0,868	-	-	-	-
Liu, Fu e Lin (2010)	gr-GA-SVM	0,868	-	-	-	-
Huang, Chen e Wang (2007)****	SVM (GA OHP)	0,869	-	-	-	-
West (2000)**	RBFN	0,871	-	-	-	-
Nanni e Lumini (2009)****	RS-LMNC	0,871	-	-	-	-
Tripathi <i>et al.</i> (2018)*	FC+WV	0,873	-	-	-	-
Shi and Xu (2016)****	SVDD-FSVM	0,873	-	-	-	-
Bequé e Lessmann (2017)**	Boost SVM-L	0,874	-	-	-	-
Guo <i>et al.</i> (2019)**	MSEC	0,874	0,94	-	-	-
Torkevar e Game (2019)	RF	0,875	-	-	-	-
Zhang <i>et al.</i> (2019)**	MHNGA	0,875	0,937	-	-	-
Ping e Yongheng (2011)****	SVM-RBF	0,875	-	-	-	-
Liang <i>et al.</i> (2018)	CART	0,88	-	-	-	-
Zhou <i>et al.</i> (2021)	SVM (MARS FS)	0,88	0,954	-	-	-
Xiao <i>et al.</i> (2016)*	ECSC	0,881	-	-	-	-
Xu <i>et al.</i> (2019)**	GFFS	0,881	0,935	-	-	-
Ala'raj & Abbod (2016)*****	GNG (MARS FS)	0,881	-	-	-	-
Xia <i>et al.</i> (2018)**	Bstacking	0,883	0,928	-	-	-
Ong, Huang e Tzeng (2005)****	<i>Genetic Programming</i>	0,883	-	-	-	-
Siami, Gholamian e Basiri (2014)	LOLIMOT	0,886	-	-	0,107	-
Vukovic <i>et al.</i> (2012)***	CBRPGA	0,886	-	-	-	-
Lahsasna, Ainon e Wah (2010)****	<i>Genetic Fuzzy Classifier</i>	0,886	-	-	-	-
Ala'raj e Abbod (2016)*	CSA	0,89	-	-	-	-
Qi <i>et al.</i> (2016)****	DeepSVM	0,89	-	-	-	-
Gozdzaczany e Rudziski (2016)***	<i>Fuzzy rule-based classifier with multi-objective evolutionary optimization</i>	0,891	-	-	-	-
Huang, Tzeng, Ong (2006)****	2SGP	0,892	-	-	-	-
Van Sang, Nam e Nham (2016)****	<i>Parallel RF</i>	0,894	-	-	-	-
Tripathi <i>et al.</i> (2020)	EELM-2 (BA OHP)	0,899	-	0,907	-	-
Tsai (2009)***	MLP (PCA FS)	0,899	-	-	-	-
Tsai (2008)****	MLP	0,902	-	-	-	-
Baesens <i>et al.</i> (2003)****	LS-SVM	0,904	-	-	-	-

Continuação da Tabela 3.2 – Resultados das 5 principais métricas na literatura para o *australian* UCI dataset

Autor	Melhor Modelo	Acurácia	AUC	TPR	Erro Tipo II ↓	F-score
Lin <i>et al.</i> (2008)****	SVM (PSO OHP; PSO FS)	0,91	-	-	-	-
Chen <i>et al.</i> (2020)	GSCI	0,912	0,914	0,945	-	0,909
Tsai (2014)***	MLP (SOM FS)	0,916	-	-	-	-
Tsai e Hung (2014)****	Hybrid NN	0,916	-	-	-	-
Zhang <i>et al.</i> (2010)***	VBDM	0,92	-	-	-	-
Edla <i>et al.</i> (2018)*	LWV (com FS)	0,926	-	-	-	-
Somol <i>et al.</i> (2005)****	<i>Gaussian Classifier</i>	0,926	-	-	-	-
Tripathi <i>et al.</i> (2019)*	LWV (com FS)	0,927	-	-	-	-
Marcano-Cedeno <i>et al.</i> (2011)****	AMMLP	0,928	-	-	-	-
Weng, Huang e Cheng-Kui (2021)	CBSVM (<i>Information Gain</i> OHP)	0,94	-	0,96	-	0,94
Tripathi <i>et al.</i> (2018)*	LWV (<i>Neighbourhood Rough Set</i> (NRS) FS)	0,954	-	-	-	-
Tsai and Wu (2008)****	ANN	0,973	-	-	-	-
Pławiak, Abdar e Acharya (2019)	DGCE SVM	0,974	-	-	-	-
Hsieh (2005)****	Hybrid SOM-KM-NN	0,98	-	-	-	-
Tsai and Hsu (2013)****	MC-LR	0,991	-	-	-	-
Armaki <i>et al.</i> (2017)	<i>Meta-learners</i> híbridos: (KNN-NN-SVMPSO)-(DL)-(SOM)/(KNN-NN-SVMPSO)-(DL)-(EM)	0,999	-	-	-	0,999
Marqués, García e Sánchez (2013a)	SVM	-	0,891	-	-	-
Zhang <i>et al.</i> (2022)	RF	-	0,931	-	-	-
Zhang <i>et al.</i> (2022)	LR	-	0,931	-	-	-
Waad, Ghazi e Mohamed (2013)	LR (<i>Three-stage</i> FS)	-	0,956	-	-	0,919
Waad, Ghazi e Mohamed (2013)	NB (<i>Three-stage</i> FS)	-	-	0,976	-	0,919
Liang <i>et al.</i> (2018)	Bag SVM-L	-	-	-	0,075	-
Liang <i>et al.</i> (2018)	SVM-L	-	-	-	0,075	-
Boughaci e Alkhawaldeh (2017)	RF (AGENTS FS)	-	0,977	-	-	-
Dumitrescu <i>et al.</i> (2022)	RF	-	0,934	-	-	-
Giri <i>et al.</i> (2021)	LGBBO	-	0,927	0,87	-	0,852
Aghaeirad, Chen e Ribeiro (2017)	FNN	-	-	0,871	-	-
Bao, Lianju e Yue (2019)	DT	-	-	-	0,017	-
Zhou <i>et al.</i> (2021)	CART (LASSO FS)	-	-	-	0,068	-
Tripathi <i>et al.</i> (2020)	EELM-1 (PSO OHP)	-	0,957	-	-	-
Pears e Oetama (2010)	GBDT	-	0,94	-	-	-
Bequé e Lessmann (2017)**	Boost LR-R	-	0,94	-	-	-
Média:		0,896	0,937	0,922	0,068	0,923

*Resultados adicionados de Chen *et al.* (2020); ** Resultados adicionados de Tripathi *et al.* (2020); *** Resultados adicionados de Pławiak, Abdar and Acharya (2019); **** Resultados adicionados de Armaki *et al.* (2017).

↓ Significa que quanto menor, melhor.

Fonte: Artigos consultados na RSL

Tabela 3.3 – Resultados das 5 principais métricas na literatura para o *german* UCI dataset

Autor	Melhor Modelo	Accuracy	AUC	F-score	Precision	Recall
Aghaeirad, Chen e Ribeiro (2017)	SOM+FNN	0,718	-	-	-	-
Martens <i>et al.</i> (2010)	C4.5	0,742	-	-	-	-
Chen e Xiang (2017)	Group Lasso (AIC OHP)	0,743	-	-	0,534	-
Baesens <i>et al.</i> (2003)***	LS-SVM	0,746	-	-	-	-
Nanni e Lumini (2009)***	RS-LMNC	0,747	-	-	-	-
Abellan e Castellano (2017)*	Bag CDT	0,748	-	-	-	-
Wang <i>et al.</i> (2011)*	Bag DT	0,749	-	-	-	-
Lahsasna, Ainon e Wah (2010)***	<i>Genetic Fuzzy Classifier</i>	0,75	-	-	-	-
Zhang <i>et al.</i> (2022)	SVM-RBF	0,753	-	-	-	-
Liu, Fu e Lin (2010)	gr-GA-SVM	0,755	-	-	-	-
Bequé e Lessmann (2017)**	Boost SVM-R	0,759	0,802	-	-	-
Hoffmann <i>et al.</i> (2007)***	NB	0,76	-	-	-	-
Van Sang, Nam e Nham (2016)***	<i>Parallel RF</i>	0,762	-	-	-	-
Ping e Yongheng (2011)***	SVM-RBF	0,766	-	-	-	-
Chen e Li (2010)***	SVM (LDA FS)	0,767	-	-	-	-
Siami, Gholamian e Basiri (2014)	LOLIMOT	0,768	-	-	-	-
Zhang <i>et al.</i> (2019)**	MHNGA	0,768	0,803	-	-	-
Tripathi <i>et al.</i> (2018)*	FC+WV	0,771	-	-	-	-
Ong, Huang e Tzeng (2005)***	<i>Genetic Programming</i>	0,773	-	-	-	-
Shi e Xu (2016)***	SVDD-FSVM	0,773	-	-	-	-
Florez-Lopez e Ramon-Jeronimo (2015)	<i>CADF Mixed</i>	0,774	-	-	-	-
Liang <i>et al.</i> (2018)	MV	0,776	-	-	-	-
Liang <i>et al.</i> (2018)	SVM-L	0,776	-	-	-	-
Ala'raj & Abbod (2016)*	CSA	0,777	-	-	-	-
Chen <i>et al.</i> (2020)	GSCI	0,778	0,704	0,582	-	-
Xiao <i>et al.</i> (2016)*	ECSC	0,778	-	-	-	-

Continuação da Tabela 3.3 – Resultados das 5 principais métricas na literatura para o *german* UCI dataset

Autor	Melhor Modelo	Accuracy	AUC	F-score	Precision	Recall
Huang, Chen e Wang (2007)***	SVM (GA OHP)	0,779	-	-	-	-
Xia <i>et al.</i> (2018)*	<i>Bagging+Stacking</i>	0,783	-	-	-	-
Guo <i>et al.</i> (2019)**	MSEC	0,783	0,806	-	-	-
Sun <i>et al.</i> (2015)	SVM-RBF	0,784	-	-	-	-
Xia <i>et al.</i> (2018)**	Bstacking	0,787	0,795	-	-	-
Tsai (2009)***	MLP (com FS)	0,788	-	-	-	-
Oreski, Oreski e Oreski (2012)***	HGA-NN	0,789	-	-	-	-
Tsai and Wu (2008)***	ANN	0,79	-	-	-	-
Ala'raj & Abbod (2016)****	GNG (MARS FS)	0,79	-	-	-	-
Tsai (2008)***	MLP	0,791	-	-	-	-
Saia e Carta (2016b)**	LDB	0,795	-	-	-	-
Huang, Tzeng e Ong (2006)***	2SGP	0,795	-	-	-	-
Saia e Carta (2016a)**	RF	0,8	0,73	-	-	-
Tripathi <i>et al.</i> (2020)	EELM-2 (BA OHP)	0,812	0,856	-	-	-
Bao, Lianju e Yue (2019)	<i>Cluster-based SVM</i>	0,813	-	-	-	0,844
Zhang <i>et al.</i> (2010)***	VBDTM	0,816	-	-	-	-
Lin <i>et al.</i> (2008)***	SVM (PSO OHP; PSO FS)	0,816	-	-	-	-
Qi <i>et al.</i> (2016)***	DeepSVM	0,837	-	-	-	-
Somol <i>et al.</i> (2005)***	<i>Gaussian Classifier</i>	0,838	-	-	-	-
Marcano-Cedeno <i>et al.</i> (2011)***	AMMLP	0,847	-	-	-	-
Luo, Cheng e Hsieh (2009)***	CLC	0,848	-	-	-	-
Tripathi <i>et al.</i> (2019)*	LWV (com FS)	0,852	-	-	-	-
Edla <i>et al.</i> (2018)*	LWV (com FS)	0,858	-	-	-	-
Tripathi <i>et al.</i> (2018)*	LWV (com FS)	0,865	-	-	-	-
West (2000)**	RBFN	0,871	-	-	-	-
Tsai e Hung (2014)***	Hybrid NN	0,875	-	-	-	-

Continuação da Tabela 3.3 – Resultados das 5 principais métricas na literatura para o *german* UCI dataset

Autor	Melhor Modelo	Accuracy	AUC	F-score	Precision	Recall
Xu <i>et al.</i> (2019)**	GFSS	0,876	0,813	-	-	-
Weng, Huang e Cheng-Kui (2021)	CBDT (<i>Information Gain</i> OHP)	0,89	-	0,92	0,93	-
Trivedi (2020)	RF (<i>Chi-square</i> FS)	0,931	-	0,931	-	-
Peng <i>et al.</i> (2008)***	LibSVM	0,94	-	-	-	-
Analide (2011)***	MLP (com FS)	0,972	-	-	-	-
Hsieh (2005)***	Hybrid SOM-KM-NN	0,985	-	-	-	-
Tsai e Hsu (2013)***	MC-LR	0,992	-	-	-	-
Zhou <i>et al.</i> (2021)	SVM (LASSO FS)	0,998	0,999	-	-	-
Armaki <i>et al.</i> (2017)	Hybrid Meta-Learner: (KNN-NN-SVMPSO)-(DL)-(DBSCAN)	0,998	-	0,997	-	-
Marqués, García e Sánchez (2013a)	LR (<i>Resampled</i> com SMOTE + WE)	-	0,753	-	-	-
Waad, Ghazi e Mohamed (2013)	LR (<i>Three-stage</i> FS)	-	0,769	-	0,681	-
Waad, Ghazi e Mohamed (2013)	NB (MaxRel FS)	-	-	0,568	-	-
Waad, Ghazi e Mohamed (2013)	RF (<i>Three-stage</i> FS)	-	-	-	-	0,59
Boughaci e Alkhalwaldeh (2017)	RF (AGENTS FS)	-	0,959	-	0,918	0,918
Handhika <i>et al.</i> (2019)	HCES-Bag (<i>base learners: GBC/DT/RF/ERT</i>) abordagem Top-T por Brier score	-	0,759	-	-	-
Handhika <i>et al.</i> (2019)	HCES-Bag (<i>base learners: GBC/RF/ERT</i>) abordagem Top-T por AUC	-	-	-	-	-
Chen and Xiang (2017)	Group Lasso (BIC OHP)	-	-	-	-	0,825
Florez-Lopez e Ramon-Jeronimo (2015)	GBT	-	0,798	-	-	-
Giri <i>et al.</i> (2021)	LGBBO	-	0,791	0,845	0,783	0,917
Bao, Lianju e Yue (2019)	<i>Cluster-based</i> LR	-	0,859	-	-	-
Bao, Lianju e Yue (2019)	ANN	-	-	-	0,68	-
Pears e Oetama (2010)	GBDT	-	0,811	-	-	-
Média:		0,812	0,812	0,807	0,754	0,819

* Resultados adicionados de Chen *et al.* (2020); ** Resultados adicionados de Tripathi *et al.* (2020); Resultados adicionados de Armaki *et al.* (2017).

Fonte: Artigos consultados na RSL

Tabela 3.4 – Resultados das 5 principais métricas na literatura para o *japanese* UCI *dataset*

Autor	Melhor Modelo	Accuracy	AUC	TPR	TNR	Erro Tipo I ↓
Aghaeirad, Chen e Ribeiro (2017)	SOM+FNN	0,848	-	0,877	0,819	-
Abellan e Castellano (2017)*	Bag CDT	0,865	-	-	-	-
Guo <i>et al.</i> (2019)**	MSEC	0,87	0,942	-	-	-
Zhang <i>et al.</i> (2019)**	MHNGA	0,872	0,939	-	-	-
Ala'raj e Abbod (2016)*	CSA	0,879	-	-	-	-
Tripathi <i>et al.</i> (2018)*	FC + WV	0,88	-	-	-	-
Tripathi <i>et al.</i> (2020)	EELM-2 (BA HPO)	0,884	0,946	0,932	-	-
Zhou <i>et al.</i> (2021)	SVM (MARS feature selection)	0,888	0,959	-	-	0,075
Tripathi <i>et al.</i> (2019)*	LWV (with feature selection)	0,891	-	-	-	-
Edla <i>et al.</i> (2018)*	FS+LWV	0,891	-	-	-	-
Xu <i>et al.</i> (2019)**	GFSS	0,894	0,933	-	-	-
Chen <i>et al.</i> (2020)	WA	0,896	0,898	-	-	-
Armaki <i>et al.</i> (2017)	Hybrid Meta-Learners: (KNN-NN-SVMPSO)-(DL)-(DBSCAN)/(KNN-NN-SVMPSO)-(DL)-(SOM)/(KNN-NN-SVMPSO)-(DL)-(FCM)	0,999	-	-	-	-
Marqués, García e Sánchez (2013a)	SVM (resampled)	-	0,888	-	-	-
Boughaci e Alkhawaldeh (2017)	RF (AGENTS FS)	-	0,982	-	-	-
Zhou <i>et al.</i> (2021)	CART	-	-	-	-	0,075
Zhou <i>et al.</i> (2021)	CART/CART (MARS FS)	-	-	-	-	0,075
Wang, Wang and Lai (2005)	B-FSVM (Polynomial Kernel, Logit Regression Member Generation)	-	-	-	-	-
Chen <i>et al.</i> (2020)	GSCI	-	-	0,922	-	-
Chen <i>et al.</i> (2020)	Gradient Boosting	-	-	-	0,887	-
Tripathi <i>et al.</i> (2020)	EELM-1 (PSO HPO)	-	-	-	0,865	-
Média:		0,889	0,936	0,91	0,857	0,075

* Resultados adicionados de Chen *et al.* (2020); **Resultados adicionados de Tripathi *et al.* (2020).

Fonte: Artigos consultados na RSL

Embora os valores médios pareçam indicar bons resultados, é difícil fazer afirmações assertivas com base nos dados coletados. Considerando as métricas de desempenho mais

populares (Figura 4.13) a maioria desses modelos é avaliada apenas pela acurácia e/ou AUC, o que pode levar a interpretações equivocadas. Isso não significa que a maioria dos estudos não avalie corretamente seus modelos. Na verdade, considerando o Anexo A, em média, os modelos são avaliados por 3,4 medidas diferentes. O principal problema é que essas medidas raramente são as mesmas. Lessmann *et al.* (2015) também criticam os autores por usarem indicadores conceitualmente semelhantes, portanto, a literatura poderia se beneficiar de um procedimento padronizado de avaliação.

Analisar os resultados da classificação por meio de poucos indicadores pode ser especialmente problemático quando se trata de risco de crédito, onde os conjuntos de dados reais geralmente estão desbalanceados e há diferentes custos de classificação. Por exemplo, se um modelo estiver *overfitted* e prever que todas as amostras pertencem à classe majoritária (dos não inadimplentes), os resultados podem parecer satisfatórios apenas pela acurácia ou o percentual de casos corretamente classificados (PCC) (como a maioria das amostras pertence à classe majoritária e será classificada corretamente). Portanto, se não avaliado adequadamente, um CSM pode ser visto como funcional mesmo sem nenhum poder discriminatório. Esse problema é agravado pelo maior custo associado ao Erro Tipo II.

Consequentemente, embora as Tabelas 3.2-3.4 indiquem medidas que consideram o Erro Tipo II (Erro Tipo II, *F-Score* e *Recall*), não há dados suficientes para fornecer uma comparação significativa entre os modelos utilizados. Além disso, é incomum que um modelo tenha o melhor desempenho em todas as métricas, o que torna ainda mais difícil comparar os modelos. Um bom exemplo vem de Liang *et al.* (2018), no qual a minimização do Erro Tipo I geralmente ocorre às custas de Erro Tipo II, e modelos com alta acurácia podem não ser adequados para CS, dada a taxa de FN. Portanto, as Tabelas 3.2-3.4 servem melhor como um instrumento de referência para comparação de modelos de acordo com a métrica de avaliação preferida.

Resultados também variam muito entre modelos, o que dificulta afirmar qualquer superioridade entre modelos. Zhou *et al.* (2021), por exemplo, relatam resultados excepcionais em todas as métricas para SVMs simples com FS (reforçando a importância das outras etapas de modelagem de CSM, além da escolha de um classificador). Neste caso, no entanto, os autores não modelaram nenhum *ensemble* para comparação, e é importante lembrar que um *ensemble* costuma ser pelo menos tão bom quanto seu melhor componente individual. Chopra e Bilare (2018) fornecem provas empíricas disso. Assim, a principal conclusão sobre a modelagem de problemas de classificação é a seguinte: se o objetivo principal é obter um desempenho elevado sem restrições, os modelos *ensemble* são a abordagem mais robusta e consistente, embora

demandem mais tempo e recursos. Por outro lado, se houver requisitos estritos do modelo (como a necessidade de interpretabilidade em CSMs), até que as ferramentas de XAI se tornem amplamente disponíveis na indústria financeira, modelos individuais de ML menos complexos são a melhor opção. Isso ocorre porque a interpretabilidade nem sempre vem em detrimento de desempenho, embora exija mais experiência do modelador.

3.3.1.8 Medidas de Interpretabilidade

Normalmente, os modelos são também utilizados por público não técnico, e a interpretação dos atributos que mais influenciam no seu resultado, sob a perspectiva da indústria financeira, adquire extrema importância (SARIEV; GERMANO; GUIDO, 2019). No que diz respeito à interpretabilidade do modelo, os resultados são mais preocupantes. Apenas 17,91% dos 67 artigos sobre risco de crédito ao consumidor abordam a interpretabilidade do modelo. À medida que o interesse por XAI aumenta, juntamente com as demandas regulatórias, observa-se que a maioria dos avanços na literatura ainda não está pronta para ser incorporada pela indústria financeira, uma vez que modelos mais complexos podem não se adequar a um sistema global de gerenciamento de risco de crédito, regulamentado conforme o Acordo de Basiléia II.

Dos 12 artigos que abordam a interpretação do modelo, mais da metade (58,33%) o fazem determinando medidas de importância dos atributos (*e.g.* CHEN; XIANG, 2017; FLOREZ-LOPEZ; RAMON-JERONIMO, 2015; CHOPRA; BILARE, 2018; WANG; XU; ZHOU, 2015; SARIEV; GERMANO, 2020; FITZPATRICK; MUES, 2016; PEARS; OETAMA, 2010). Isto é feito de maneiras diferentes pelos autores. Chopra e Bilare (2018) obtêm a importância de cada atributo por meio do *software RapidMiner*. Sariev e Germano (2020) analisam a correlação do atributo com a variável dependente (como em alguns métodos de FS). Fitzpatrick e Mues (2016) utilizam diferentes testes de estatísticos para determinar a importância relativa das variáveis. Wang, Xu e Zhou (2015) experimentam com *Log-Likelihood Ratio Occurrence*, enquanto Pears e Oetama (2010) usam *boosted trees*. Estudos que empregam algoritmos *rule-based* também abordam a interpretabilidade pelo número de regras e tamanho médio dos conjuntos de decisões (FLOREZ-LOPEZ; RAMON-JERONIMO, 2015; DUMITRESCU *et al.*, 2022).

No entanto, a abordagem de importância de atributos, como discutido, é amplamente criticada por muitos autores, que afirmam que não é uma forma viável de determinar os principais fatores de risco. Embora possa fornecer uma ideia geral de quais variáveis

contribuem mais para a previsão do modelo, técnicas como esta não constituem o melhor método de interpretabilidade, além de não atenderem aos padrões regulatórios.

Os métodos predominantes para estimar importância de atributos não são igualmente aplicáveis a todos os modelos e podem não ser consistentes entre eles, fornecendo resultados diferentes com a menor das alterações no modelo ou nos dados de treinamento. Esses métodos também podem ter viés em relação a atributos de grande escala ou muitos valores únicos, geralmente não considerando a interação entre fatores e não sendo capazes de determinar seus relacionamentos não lineares com a variável dependente. Portanto, métodos de importância de atributos fornecem uma visão global do comportamento do modelo, mas não uma interpretabilidade local, deixando de fornecer muitos *insights* sobre como um modelo se comporta com dados específicos.

Felizmente, outros métodos podem fornecer uma compreensão mais abrangente dos modelos. No entanto, eles parecem estar ainda em estágios iniciais de pesquisa quando se trata de CSMs. Nesse contexto, Moscato, Picariello e Sperlí (2021) experimentam com LIME, SHAP, *Anchors*, *Balanced English Explanations of Forecasts* (BEEF) e *Local Rule-based Explanations* (LORE), afirmando que o LORE apresenta melhores resultados, pois combina previsões locais para gerar explicações. Ariza-Garzón (2020) fornece o único outro estudo que usa SHAP para interpretação de modelos, enquanto Bussman *et al.* (2021) usa o TreeSHAP, mas para aplicações de risco de crédito corporativo.

Além disso, Pan (2021) tenta melhorar a interpretabilidade por meio de *Collaborative Filtering* (CF) em seus modelos de NN propostos. Martens *et al.* (2010) desenvolvem um modelo inerentemente interpretável usando o AntMiner+ (um algoritmo de classificação inspirado na abordagem metaheurística da Otimização de Colônia de Formigas ou *Ant Colony Optimization*), que consideram adequado para aplicações comerciais. E finalmente, Wang, Xu e Zhou (2015) testam uma ferramenta muito interessante inspirada em Janitza, Strobl e Boulesteix (2013), usando uma metodologia que analisa as permutações em AUC para RFs, buscando interpretar características, analisando o impacto que sua variação tem no modelo de acordo com a métrica de desempenho AUC.

3.4 Considerações Finais da Revisão Sistemática de Literatura

Este Capítulo teve como principais objetivos (a) desenvolver uma RSL sobre ML para fins de CS, de modo a (b) identificar e explorar os principais aspectos relacionados ao tema, (c) analisar e sumarizar os resultados a fim de responder as QPs.

A escolha pelo uso da RSL provou-se uma técnica satisfatória na exploração e análise da literatura científica existente. As diretrizes propostas por Denyer e Tranfield (2009) e Denyer *et al.* (2008) contribuíram significativamente para uma melhor condução desta pesquisa, prevenindo possíveis distorções e parcialidades dos autores sobre os resultados encontrados, atendendo assim ao primeiro objetivo proposto. Vale ressaltar que esta pesquisa não almejou esgotar todos os artigos sobre o tema, porém analisar sistematicamente uma amostra representativa que permitiu responder as QPs de modo a auxiliar futuros pesquisadores e profissionais no entendimento e apresentação da fronteira do conhecimento existente na área.

Até o momento, foi construído um embasamento teórico sólido acerca de CS e seu papel na administração do risco de crédito, bem como da modelagem de CSMs por meio de algoritmos de classificação, com enfoque nas técnicas de ML. Por meio da RSL, foram identificados os principais algoritmos de ML empregados no contexto de CS, assim como observadas diferentes técnicas que podem ou não compor outras etapas do processo de modelagem, destacando-se o pré-processamento de dados, *resampling*, FS e OHP.

Conforme os artigos selecionados, observou-se que a LR ainda vigora na indústria financeira para fins de CS. Contudo, existe um crescente interesse acadêmico e profissional por algoritmos de ML com este propósito. Numerosos modelos foram propostos em cima destes algoritmos e, analisando os mais populares, é possível afirmar que não existe, entre eles, um algoritmo de performance consistentemente superior. Apesar disso, algoritmos de ML costumam triunfar sobre as tradicionais metodologias estatísticas, em especial os algoritmos de *ensemble learning*.

As etapas de pré-processamento de dados, *resampling*, FS e OHP também demonstram enorme potencial para a otimização de CSMs, sendo encontradas evidências de que a sua devida implementação tem um efeito positivo nos modelos construídos em cima da maior parte dos algoritmos de ML. Contudo, grande parte das publicações acerca do tema pecam no processo de modelagem, não fazendo o devido uso das etapas mencionadas, lidando incorretamente com os dados e algoritmos e até mesmo interpretando equivocadamente as métricas de performance.

Também não evidenciou-se um padrão nos processos de modelagem desenvolvidos na literatura e nem propostas para estabelecer devidamente esta padronização, sendo inconsistente a utilização das demais etapas de modelagem discutidas. Esta falta de diretrizes para uma abordagem sistêmica do problema reforça a importância de *frameworks* para guiar pesquisadores e profissionais na otimização da modelagem de CSMs e, aliada ao grande potencial dos algoritmos de ML e sua atratividade para a indústria, justifica os objetivos deste trabalho e a sua importância em buscar desenvolver tais diretrizes.

De acordo com a análise conduzida, modelos de ML individuais são os classificadores mais comumente escolhidos. Regressões e modelos *ensemble*, no entanto, ainda aparecem entre os algoritmos mais utilizados, uma lista que consiste em: SVM, NN, DT, RF, algoritmos de *boosting* e, principalmente para referência, LR. Considerando que modelos complexos são consideravelmente mais caros, além de mais difíceis de compreender e reproduzir, a literatura pode se beneficiar mais de estruturas e diretrizes bem definidas que visam aproveitar ao máximo os CSMs. Isso inclui o uso de ML em várias etapas do desenvolvimento do modelo, como *resampling*, FS e HPO, sabendo que eles podem melhorar drasticamente o desempenho, além de servirem como uma forma de familiarizar os modeladores com o processo de desenvolvimento de CSMs. Dito isso, é preocupante que essas etapas de modelagem, como no caso da interpretabilidade, pareçam ser pouco representadas na literatura. Pesquisas futuras sobre essas etapas importantes da modelagem de risco de crédito podem ser úteis, investigando as melhores maneiras de realizá-las de acordo com a escolha.

Em relação à avaliação, muitos autores utilizam conjuntos de dados muito pequenos e/ou de baixa dimensionalidade e não disponíveis publicamente, o que dificulta avaliar/comparar a qualidade dos modelos e não reflete os desafios reais da previsão de risco de crédito ao consumidor. Como consequência, a literatura também pode se beneficiar de procedimentos de avaliação padronizados para garantir que o desempenho do modelo seja avaliado corretamente, e não por meio de conjuntos de dados e métricas de avaliação que possam não contribuir com informações úteis. *Designs* experimentais avaliados com procedimentos de *cross validation* também podem beneficiar potencialmente acadêmicos e profissionais da indústria.

Resumidamente, ML se tornou uma realidade crescente na área de CS e do gerenciamento de risco em geral. No entanto, ainda há necessidade de pesquisas adicionais sobre o desenvolvimento e a interpretabilidade de modelos antes que os métodos mais avançados possam ser adequadamente adotados para uso comercial por instituições financeiras. Por fim, recomenda-se que os pesquisadores usem e adaptem a metodologia de RSL para avaliar novos tópicos em suas respectivas áreas de especialização. Também incentiva-se uma análise mais aprofundada sobre o papel da IA no gerenciamento de riscos financeiros e como ela pode ser melhor explorada na indústria.

4. MODELAGEM

Com o objetivo de embasar empiricamente o desenvolvimento dos *frameworks* propostos, além do conhecimento técnico e do que vigora na literatura (assuntos explorados, respectivamente, na fundamentação teórica e nos resultados da RSL), foram construídos diferentes CSMs, partindo de todos os tipos de algoritmo de classificação discutidos. O procedimento experimental consistiu na modelagem de todas as combinações entre o classificador e as demais etapas abordadas, conforme expõe a Figura 4.1. Toda a modelagem foi feita em Python, por meio da biblioteca *scikit-learn*.

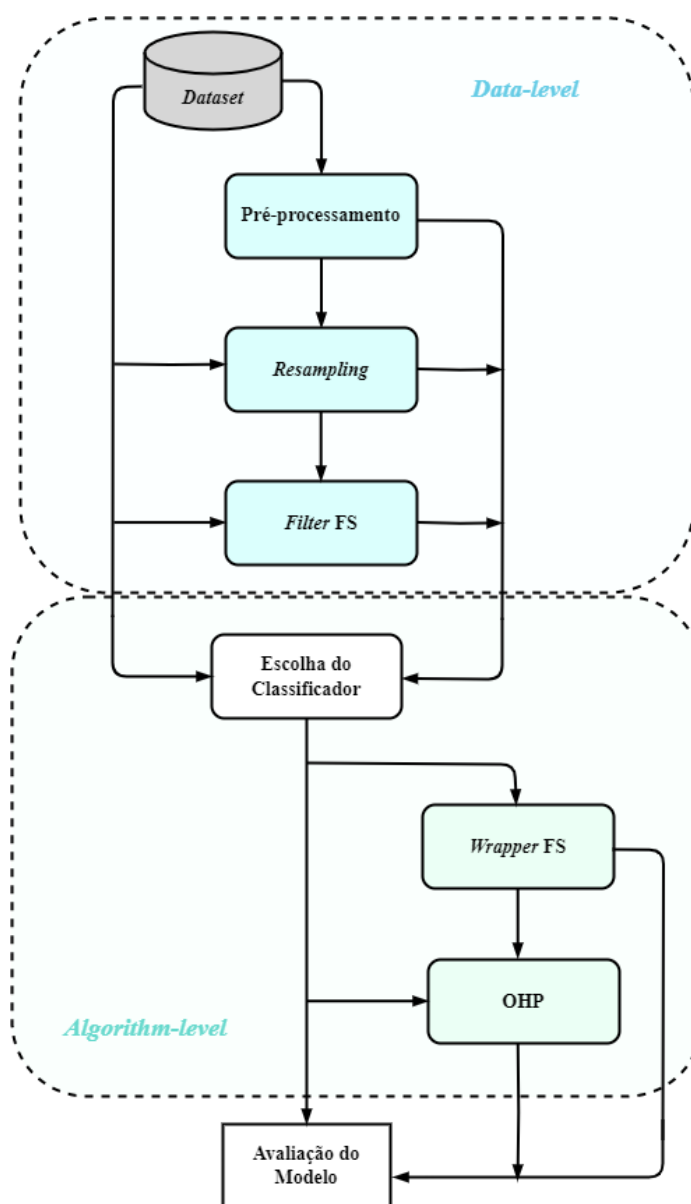


Figura 4.1 – Planejamento para a modelagem de CSMs
Fonte: Autoria própria (2023)

Todos os modelos foram avaliados nos três *datasets* mais populares da literatura (*australian*, *german* e *japanese credit*), que se encontram publicamente disponíveis no repositório de ML da UCI. Os algoritmos de classificação, a partir dos quais os CSMs foram modelados, encontram-se no Quadro 4.1.

Quadro 4.1 – Algoritmos de classificação utilizados

Tipo	Algoritmo
Estatístico	LR
ML Individual	DT SVM
ML Ensemble Homogêneo	RF AdaBoost (BOOST)
ML Ensemble Heterogêneo	<i>Stacking</i> (STACK)

Fonte: Autoria própria (2023)

Desta forma, além de cobrir a LR (tida como o *benchmark* da indústria financeira para a classificação do risco de crédito) e dois dos modelos individuais mais populares de ML, considerando que RF é uma forma de *bagging*, são também analisados todos os métodos de agregação discutidos para modelos *ensemble*.

Como observa-se na Figura 4.1, todos estes diferentes classificadores foram modelados e avaliados com e sem diversas etapas distintas de modelagem (lembrando que nesta pesquisa foram tratados o pré-processamento de dados, balanceamento de classes, FS e OHP. Para o balanceamento de classes, foi aplicada uma combinação de algoritmos de *oversampling* e *undersampling*, mais particularmente, em ordem respectiva: SMOTE e *Tomek's links*, igualando o número de amostras pertencentes a cada classe.

Considerando a significativa distinção entre as formas de FS, foram implementados ambos os métodos: *filter* e *wrapper*. Quanto aos respectivos algoritmos, em particular, optou-se pelo *filter* CFS, seguindo a sugestão de Radomska-Zalas e Becker (2020), e o *wrapper* mais popular de acordo com a RSL: *Sequential Forward Selection* (SFS). Ambos os distintos métodos de FS foram empregados para a redução do número de atributos de cada *dataset* para os cinco de maior potencial preditivo segundo o algoritmo.

A etapa de OHP foi cumprida com uma simples *grid search*, também o método mais comum para tal finalidade de acordo com a RSL, estabelecendo faixas para os parâmetros

escolhidos para cada classificador. Os diferentes parâmetros e suas respectivas faixas, assim como a etapa de pré-processamento de dados, serão abordados separadamente a seguir.

4.1 Descrição e Análise Exploratória dos Dados

Na Tabela 4.1 encontram-se os principais dados acerca dos *datasets* utilizados, mais particularmente, o número de amostras, atributos e percentual de inadimplentes.

Tabela 4.1– Descrição dos *datasets*

<i>Dataset</i>	Núm. Amostras	Núm. Atributos	% Inadimplentes
<i>Australian</i>	690	14	44,49%
<i>German</i>	1000	64	30%
<i>Japanese</i>	670	46	55,22%

Fonte: UCI Machine Learning Repository

A grande diferença entre o número de atributos se dá por causa da presença de variáveis categóricas nos *datasets German* e *Japanese*. Por isso, para prepará-las para o modelo, foi realizada a codificação destas variáveis, o que aumenta significativamente o número de atributos nestes casos (a codificação em si é explorada na seção seguinte, junto ao pré-processamento de dados). Nesta breve análise exploratória, é possível obter uma visão geral das bases, especialmente quanto ao balanceamento entre as classes. As Figuras 4.2-4.4 ilustram a distribuição entre classes nos *datasets Australian*, *German* e *Japanese*. Nos gráficos, as amostras são separadas por classe e dispersas por dois atributos numéricos quaisquer, permitindo que se enxergue o nível de desbalanceamento.

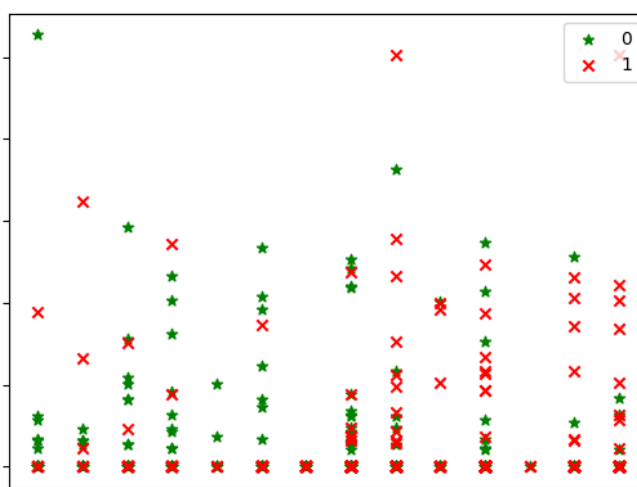


Figura 4.2 – Distribuição por classe entre atributos do *dataset Australian* do UCI ML Repository.

Fonte: Autoria própria (2023)

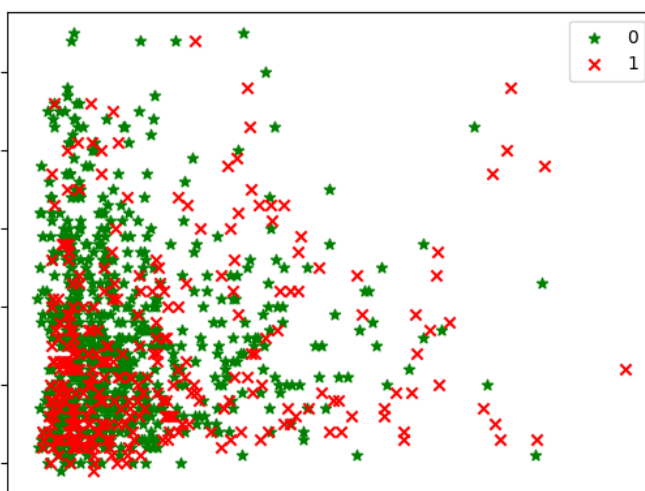


Figura 4.3 – Distribuição por classe entre atributos do *dataset German* do UCI ML Repository.
Fonte: Autoria própria (2023)

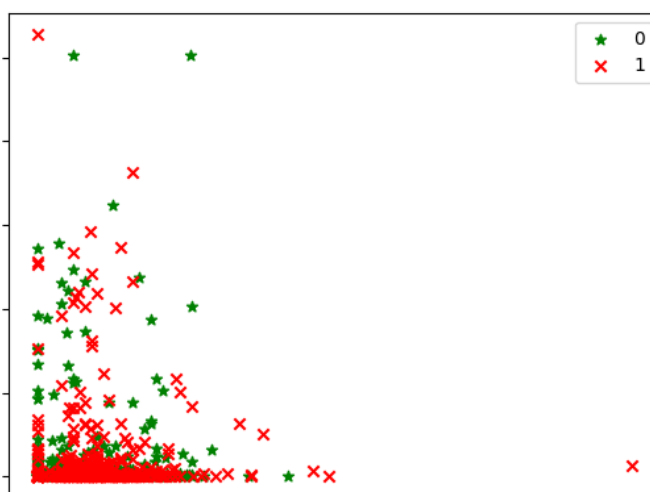


Figura 4.4 – Distribuição por classe entre atributos do *dataset Japanese* do UCI ML Repository.
Fonte: Autoria própria (2023)

Pode-se notar o diferente grau de desbalanceamento entre os *datasets*, o que deve impactar a importância da etapa de *resampling* em cada um dos deles. Da mesma forma, na etapa de *feature selection*, os resultados podem variar significativamente dependendo do número de atributos escolhidos. Neste sentido, este estudo se beneficia por conduzir a avaliação dos modelos em múltiplos *datasets*. No entanto, o objetivo é ter uma visão geral do efeito que as etapas de modelagem têm sobre o desempenho da classificação por meio dos algoritmos escolhidos.

4.2 Pré-processamento dos *datasets*

A etapa de pré-processamento de dados foi realizada por partes. Em primeiro lugar, para que seja possível a classificação, bem como o restante da modelagem, é necessário que as variáveis sejam numéricas. Desta forma, todos os modelos contam ao menos com uma etapa de pré-processamento: o *one-hot encoding*. Por meio desta técnica de codificação, as variáveis categóricas são representadas de forma numérica e binária, o que é alcançado criando colunas adicionais para cada categoria única da variável, recebendo o valor de 1 quando presente a categoria, e 0 caso contrário.

Entretanto, os modelos que foram considerados como pré-processados contaram ainda com a remoção de outliers, nos quais valores que estivessem além de três desvios-padrão da média de uma variável foram substituídos pelos limites superior ou inferior, reduzindo a influência de valores extremos e capazes de distorcer as análises subsequentes; e também a padronização (ou *standardization*), que transforma as variáveis para um número correspondente em uma escala com média zero e desvio padrão unitário, a fim de criar uma escala comparável, evitando que diferenças nas magnitudes entre os valores interfira no processo de aprendizado.

4.3 Aplicação e Resultados

Conforme a Figura 4.1, foram modelados, a partir de cada um dos classificadores do Quadro 4.1, CSMs pareados ou não com as demais etapas discutidas, indicadas nos nomes dos modelos da seguinte forma:

- PP: Inclui etapa de pré-processamento;
- FFS: Inclui etapa de *Filter FS*;
- WFS: Inclui etapa de *Wrapper FS*;
- RES: Inclui etapa de *resampling*;
- OHP: Inclui etapa de OHP.

Sabendo que foram testados 6 diferentes algoritmos de classificação, considerando os 3 diferentes *datasets* e implementando todas as possíveis combinações entre as etapas PP, FFS, WFS, RES e OHP, chegou-se a um total de 432 modelos. Para a avaliação de desempenho, cada modelo foi, por fim, avaliado nas métricas Erro Tipo I, Erro Tipo II, Acurácia e AUC com *10-fold cross-validation*, em que base de dados é dividida em dez partes de tamanhos similares

para que o modelo seja treinado e avaliado em cada uma delas. Os resultados encontram-se divididos por classificador nas Tabelas 4.2-4.17.

Visando facilitar a comparação entre os modelos como CSMs, foi também atribuído a cada um deles, por *dataset*, um *ranking* ponderado, referido como *ranking* final, considerando a sua performance em comparação aos demais modelos, porém priorizando as métricas em ordem respectiva: Erro Tipo II, AUC, acurácia e Erro Tipo I. Desta forma, para determinado *dataset*, o melhor modelo não é necessariamente aquele com melhor desempenho médio entre as métricas de performance, sendo favorecidos, no *ranking* final, os modelos que se destacam nas métricas de Erro Tipo II (conhecido o seu impacto na classificação do risco de crédito) e AUC (respectivamente, com pesos de 0,5 e 0,3), seguido do Erro Tipo I e Acurácia (ambos com peso de 0,1).

4.3.1 Modelos LR

Como já fora mencionado anteriormente, considerando os CSMs mais populares utilizados na indústria financeira, o algoritmo de classificação para a primeira série de modelos não poderia ser outro que não LR. Desta forma, os resultados, apresentados na Tabela 4.2, não só contribuem para o desenvolvimento dos *frameworks*, como fornecem um importante *benchmark* para conclusões acerca dos modelos subsequentes.

Tabela 4.2 – Resultados para os modelos LR

<i>Dataset</i>	Modelo	Acurácia	Erro Tipo I	Erro Tipo II	AUC	Rank Médio	Rank Final
Australian	LR	0,804	0,124	0,289	0,793	18,50	21,8
	LR_WFS	0,855	0,203	0,070	0,864	14,75	9,9
	LR_OHP	0,874	0,149	0,094	0,878	9,50	10,8
	LR_WFS_OHP	0,855	0,203	0,070	0,864	14,75	9,9
	LR_PP	0,871	0,131	0,119	0,875	10,75	14,9
	LR_PP_WFS	0,849	0,203	0,084	0,857	18,25	14,9
	LR_PP_OHP	0,872	0,151	0,094	0,878	10,25	11,3
	LR_PP_WFS_OHP	0,857	0,203	0,066	0,865	13,75	8,7
	LR_FFS	0,858	0,160	0,117	0,861	17,50	18,8
	LR_FFS_OHP	0,865	0,168	0,090	0,871	12,75	11,1
	LR_RES	0,807	0,135	0,256	0,805	18,25	21,1
	LR_RES_WFS	0,860	0,198	0,080	0,861	15,00	12,4
	LR_RES_OHP	0,866	0,151	0,118	0,865	15,00	17,4
	LR_RES_WFS_OHP	0,860	0,198	0,080	0,861	15,00	12,4
	LR_PP_FFS	0,859	0,165	0,110	0,863	17,00	18
	LR_PP_FFS_OHP	0,862	0,167	0,100	0,866	15,25	15,5
	LR_PP_RES	0,889	0,136	0,090	0,887	4,50	5,8

Continuação da Tabela 4.2 – Resultados para os modelos LR

	Modelo	Acurácia	Erro Tipo I	Erro Tipo II	AUC	Rank Médio	Rank Final
German	LR_PP_RES_WFS	0,889	0,147	0,079	0,887	4,50	4
	LR_PP_RES_OHP	0,893	0,115	0,102	0,891	5,25	9,5
	LR_PP_RES_WFS_OHP	0,889	0,147	0,079	0,887	4,50	4
	LR_RES_FFS	0,868	0,167	0,098	0,867	13,50	14,2
	LR_RES_FFS_OHP	0,871	0,167	0,093	0,870	12,25	12,3
	LR_PP_RES_FFS	0,885	0,140	0,092	0,884	6,75	8,1
	LR_PP_RES_FFS_OHP	0,885	0,140	0,092	0,884	6,75	8,1
	LR	0,742	0,135	0,541	0,662	13,50	17,4
	LR_WFS	0,749	0,080	0,643	0,639	12,25	17,5
	LR_OHP	0,721	0,292	0,255	0,727	11,00	9
	LR_WFS_OHP	0,674	0,352	0,268	0,690	18,25	14,9
	LR_PP	0,748	0,139	0,510	0,675	12,75	16,5
	LR_PP_WFS	0,749	0,073	0,661	0,633	13,00	19,2
	LR_PP_OHP	0,720	0,288	0,269	0,721	12,50	11,4
	LR_PP_WFS_OHP	0,658	0,379	0,250	0,685	17,75	12,7
	LR_FFS	0,741	0,097	0,644	0,630	15,00	19,6
	LR_FFS_OHP	0,678	0,346	0,275	0,690	18,50	16
	LR_RES	0,771	0,255	0,201	0,772	4,50	3
	LR_RES_WFS	0,707	0,211	0,373	0,708	13,75	14,7
	LR_RES_OHP	0,773	0,267	0,187	0,773	4,25	2,3
	LR_RES_WFS_OHP	0,712	0,202	0,374	0,712	13,25	14,7
	LR_PP_FFS	0,743	0,094	0,644	0,631	14,00	19
	LR_PP_FFS_OHP	0,698	0,298	0,318	0,692	16,50	15,2
	LR_PP_RES	0,761	0,254	0,223	0,761	5,75	4,7
	LR_PP_RES_WFS	0,695	0,241	0,364	0,698	15,00	15,2
	LR_PP_RES_OHP	0,763	0,263	0,210	0,763	5,50	4
	LR_PP_RES_WFS_OHP	0,696	0,242	0,363	0,698	14,50	14,4
	LR_RES_FFS	0,722	0,300	0,253	0,724	11,50	9,2
	LR_RES_FFS_OHP	0,723	0,285	0,266	0,724	10,50	9,4
	LR_PP_RES_FFS	0,714	0,318	0,251	0,716	12,50	9,2
	LR_PP_RES_FFS_OHP	0,714	0,318	0,251	0,716	12,50	9,2
	LR	0,869	0,105	0,152	0,872	11,75	8,5
LR_WFS	0,860	0,060	0,208	0,866	14,50	14,2	
LR_OHP	0,869	0,088	0,166	0,873	11,00	8,4	
LR_WFS_OHP	0,863	0,052	0,208	0,870	11,75	12,3	
LR_PP	0,873	0,108	0,142	0,875	10,50	7	
LR_PP_WFS	0,860	0,060	0,208	0,866	14,50	14,2	
LR_PP_OHP	0,869	0,078	0,175	0,873	12,25	11,5	
LR_PP_WFS_OHP	0,863	0,052	0,208	0,870	11,75	12,3	
LR_FFS	0,695	0,000	1,000	0,500	17,75	21,3	
LR_FFS_OHP	0,410	0,669	0,376	0,478	23,75	23,5	
LR_RES	0,868	0,093	0,170	0,869	14,00	12	
LR_RES_WFS	0,863	0,059	0,215	0,863	15,50	17,8	
LR_RES_OHP	0,878	0,081	0,167	0,876	10,25	8,7	
Japanese							

Continuação da Tabela 4.2 – Resultados para os modelos LR

Modelo	Acurácia	Erro Tipo I	Erro Tipo II	AUC	Rank Médio	Rank Final
LR_RES_WFS_OHP	0,868	0,050	0,215	0,868	13,00	15,8
LR_PP_FFS	0,860	0,060	0,208	0,866	14,50	14,2
LR_PP_FFS_OHP	0,860	0,060	0,208	0,866	14,50	14,2
LR_PP_RES	0,903	0,061	0,132	0,903	5,00	3,2
LR_PP_RES_WFS	0,891	0,040	0,173	0,894	6,25	8,1
LR_PP_RES_OHP	0,904	0,067	0,124	0,905	5,00	2,6
LR_PP_RES_WFS_OHP	0,893	0,040	0,170	0,895	5,50	7,2
LR_RES_FFS	0,860	0,061	0,218	0,860	19,25	20,9
LR_RES_FFS_OHP	0,862	0,061	0,215	0,862	18,00	19
LR_PP_RES_FFS	0,893	0,042	0,167	0,895	4,00	4,6
LR_PP_RES_FFS_OHP	0,893	0,042	0,167	0,895	4,00	4,6

Fonte: Autoria própria (2023)

Para garantir a reprodutibilidade dos modelos e garantir a consistência da análise comparativa, o parâmetro *random_state* foi fixado igualmente para todos os modelos (o que também foi para a avaliação com *10-fold cross-validation*). Os demais parâmetros pertinentes, bem como a sua faixa de variação na *grid search* que compõe a etapa de OHP destes modelos, encontram-se na Tabela 4.3. Já os melhores parâmetros para os modelos LR com OHP estão no Anexo B.

Tabela 4.3 – Parâmetros da LR e amplitude das faixas de variação na *grid search*

Parâmetro	Amplitude
<i>penalty</i>	['l1', 'l2', 'elasticnet']
C	[0.01, 100]
<i>class_weight</i>	[dict, 'balanced']
<i>max_iter</i>	[100, 1000]
<i>solver</i>	['lbfgs', 'newton-cg', 'newton-cholesky']

Fonte: *scikit-learn*

Apesar de alguns modelos partindo de LR apresentarem ótimos resultados em alguns dos *datasets*, considerando os *rankings* elaborados, o modelo partindo deste algoritmo com o melhor *ranking* médio entre os *datasets* obteve o pior resultado entre os classificadores analisados. Na Tabela 4.4 encontram-se os melhores modelos de LR conforme o *ranking*.

Tabela 4.4 – Resultados para os melhores modelos LR

	<u>Melhor Modelo</u>	<u>Rank Final</u>
<i>Australian</i>	LR_PP_RES_WFS	4,00
<i>German</i>	LR_RES_OHP	2,30
<i>Japanese</i>	LR_PP_RES_OHP	2,60
Geral	LR_PP_RES	4,57

Fonte: Autoria própria (2023)

Na Figura 4.5, pode ser observada ainda a variação da performance dos modelos que incluem cada uma das etapas, bem como o efeito da interação entre elas. No gráfico, encontra-se ilustrado, em ordem decrescente, o *rank* final médio de cada modelo LR nos três *datasets*, demonstrando o aumento de performance do CSM conforme incrementa-se o modelo.

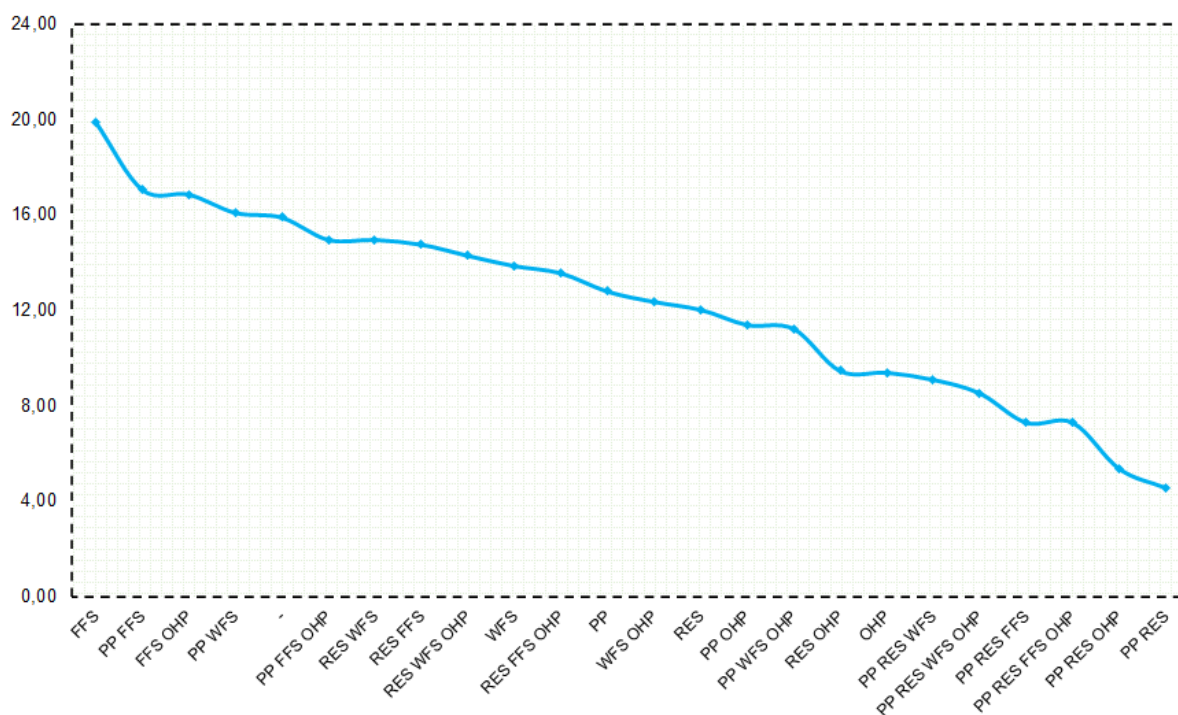


Figura 4.5 – Rank final médio de cada um dos modelos LR

Fonte: Autoria própria (2023)

Analisando a Figura 4.5, com base nas evidências dos modelos elaborados nesta pesquisa, nota-se que, no caso da modelagem de CSMs a partir de LR, as etapas de pré-processamento de dados e *resampling* não só são as que mais causam variações nos modelos, como são os principais fatores para o aumento em sua performance, estando presentes em todos os modelos da Tabela 4.4 (exceto para o *dataset german*, em que há irrisório decréscimo no melhor modelo quando pré-processado). Embora não apareça consistentemente entre os melhores modelos, etapas de FS também impactaram positivamente alguns dos modelos.

4.3.2 Modelos DT

Os resultados referentes a todos os modelos de DT encontram-se abaixo, na Tabela 4.5. A DT, mais particularmente, na biblioteca empregada para a modelagem, é o algoritmo CART.

Tabela 4.5 – Resultados para os modelos DT

<i>Dataset</i>	<i>Modelo</i>	<i>Acurácia</i>	<i>Erro Tipo I</i>	<i>Erro Tipo II</i>	<i>AUC</i>	<i>Rank Médio</i>	<i>Rank Final</i>	
<i>Australian</i>	DT	0,814	0,149	0,226	0,812	20,50	20,2	
	DT_WFS	0,859	0,154	0,122	0,862	11,00	8,8	
	DT_OHP	0,857	0,120	0,171	0,855	9,75	10,9	
	DT_WFS_OHP	0,864	0,163	0,101	0,868	9,50	6,6	
	DT_PP	0,813	0,149	0,229	0,811	21,00	21,4	
	DT_PP_WFS	0,813	0,149	0,229	0,811	21,00	21,4	
	DT_PP_OHP	0,857	0,120	0,171	0,855	9,75	10,9	
	DT_PP_WFS_OHP	0,857	0,120	0,171	0,855	9,75	10,9	
	DT_FFS	0,836	0,111	0,233	0,828	15,75	19,1	
	DT_FFS_OHP	0,878	0,082	0,169	0,874	5,00	7,2	
	DT_RES	0,826	0,159	0,192	0,825	19,25	18,7	
	DT_RES_WFS	0,846	0,195	0,111	0,847	15,00	11	
	DT_RES_OHP	0,860	0,104	0,176	0,860	8,75	11,3	
	DT_RES_WFS_OHP	0,849	0,202	0,097	0,850	13,75	9,5	
	DT_PP_FFS	0,819	0,141	0,234	0,813	20,25	21,9	
	DT_PP_FFS_OHP	0,861	0,142	0,124	0,867	9,50	8,4	
	DT_PP_RES	0,872	0,120	0,135	0,873	6,75	7,3	
	DT_PP_RES_WFS	0,893	0,131	0,085	0,892	5,00	3,4	
	DT_PP_RES_OHP	0,898	0,093	0,115	0,896	2,50	3,6	
	DT_PP_RES_WFS_OHP	0,894	0,131	0,082	0,894	4,25	2,5	
	DT_RES_FFS	0,819	0,132	0,227	0,820	18,50	19,4	
	DT_RES_FFS_OHP	0,857	0,127	0,157	0,858	10,25	10,1	
	DT_PP_RES_FFS	0,849	0,126	0,181	0,847	14,00	15,2	
	DT_PP_RES_FFS_OHP	0,849	0,126	0,181	0,847	14,00	15,2	
	<i>German</i>	DT	0,666	0,257	0,512	0,616	19,25	16,5
		DT_WFS	0,727	0,037	0,823	0,570	16,25	20,9
		DT_OHP	0,714	0,168	0,554	0,639	16,00	17
		DT_WFS_OHP	0,730	0,040	0,808	0,576	15,75	20,1
		DT_PP	0,665	0,256	0,519	0,612	19,50	17,2
		DT_PP_WFS	0,665	0,256	0,519	0,612	19,50	17,2
		DT_PP_OHP	0,715	0,169	0,547	0,642	15,00	15,4
		DT_PP_WFS_OHP	0,715	0,169	0,547	0,642	15,00	15,4
DT_FFS		0,695	0,174	0,615	0,606	19,00	20,2	
DT_FFS_OHP		0,740	0,121	0,585	0,647	13,00	15,4	
DT_RES		0,771	0,235	0,222	0,772	9,50	6,4	
DT_RES_WFS		0,801	0,077	0,319	0,802	4,75	5,9	
DT_RES_OHP		0,787	0,168	0,257	0,787	6,50	5,6	
DT_RES_WFS_OHP		0,803	0,074	0,318	0,804	3,75	4,9	

Continuação da Tabela 4.5 – Resultados para os modelos DT

	Modelo	Acurácia	Erro Tipo I	Erro Tipo II	AUC	Rank Médio	Rank Final
<i>Japanese</i>	DT_PP_FFS	0,694	0,175	0,615	0,605	19,75	20,7
	DT_PP_FFS_OHP	0,740	0,121	0,585	0,647	13,00	15,4
	DT_PP_RES	0,773	0,234	0,216	0,775	8,50	5,4
	DT_PP_RES_WFS	0,750	0,056	0,440	0,752	10,00	11,2
	DT_PP_RES_OHP	0,788	0,189	0,232	0,790	7,25	5,1
	DT_PP_RES_WFS_OHP	0,756	0,056	0,429	0,758	7,50	9
	DT_RES_FFS	0,753	0,155	0,334	0,755	9,75	9,7
	DT_RES_FFS_OHP	0,789	0,208	0,214	0,789	6,75	3,9
	DT_PP_RES_FFS	0,754	0,187	0,305	0,754	10,25	8,5
	DT_PP_RES_FFS_OHP	0,754	0,187	0,305	0,754	10,25	8,5
	DT	0,802	0,236	0,169	0,797	18,50	16,2
	DT_WFS	0,863	0,052	0,208	0,870	12,75	13,1
	DT_OHP	0,873	0,142	0,118	0,870	10,00	7,2
	DT_WFS_OHP	0,863	0,052	0,208	0,870	12,75	13,1
	DT_PP	0,802	0,236	0,169	0,797	18,50	16,2
	DT_PP_WFS	0,802	0,236	0,169	0,797	18,50	16,2
	DT_PP_OHP	0,873	0,142	0,118	0,870	10,00	7,2
	DT_PP_WFS_OHP	0,873	0,142	0,118	0,870	10,00	7,2
	DT_FFS	0,693	0,019	0,959	0,511	18,50	21,8
	DT_FFS_OHP	0,701	0,008	0,957	0,517	17,50	20,8
	DT_RES	0,841	0,162	0,157	0,841	16,50	13,2
	DT_RES_WFS	0,873	0,039	0,215	0,873	11,25	13,7
	DT_RES_OHP	0,875	0,115	0,134	0,875	9,25	7,5
	DT_RES_WFS_OHP	0,873	0,039	0,215	0,873	11,25	13,7
	DT_PP_FFS	0,860	0,060	0,208	0,866	15,50	15,6
	DT_PP_FFS_OHP	0,860	0,060	0,208	0,866	15,50	15,6
	DT_PP_RES	0,878	0,113	0,128	0,879	8,25	6,5
	DT_PP_RES_WFS	0,899	0,031	0,167	0,901	3,75	5,1
	DT_PP_RES_OHP	0,901	0,099	0,098	0,901	4,50	2,4
	DT_PP_RES_WFS_OHP	0,899	0,031	0,167	0,901	3,75	5,1
	DT_RES_FFS	0,867	0,052	0,215	0,867	14,75	16,7
	DT_RES_FFS_OHP	0,868	0,049	0,215	0,868	14,00	16,2
	DT_PP_RES_FFS	0,896	0,037	0,167	0,898	5,25	6,1
DT_PP_RES_FFS_OHP	0,896	0,037	0,167	0,898	5,25	6,1	

Fonte: Autoria própria (2023)

Neste caso, o parâmetro *random_state* foi também fixado. Os demais parâmetros pertinentes, bem como a sua faixa de variação na *grid search* que compõe a etapa de OHP dos modelos de DT, encontram-se na Tabela 4.6. Já os melhores parâmetros para os modelos DT com OHP estão no Anexo C.

Tabela 4.6 – Parâmetros da DT e amplitude das faixas de variação na *grid search*

Parâmetro	Amplitude
<i>criterion</i>	['gini', 'entropy']
<i>max_weight</i>	[None; 5, 25]
<i>min_samples_split</i>	[2, 20]
<i>min_samples_leaf</i>	[1, 8]
<i>max_features</i>	[None, 'sqrt', 'log2']

Fonte: *scikit-learn*

Na Tabela 4.7 e na Figura 4.6, respectivamente, encontram-se os melhores modelos de DT obtidos, bem como a variação de sua performance de acordo com as etapas incluídas em sua modelagem.

Tabela 4.7 – Resultados para os melhores modelos DT

	Melhor Modelo	Rank Final
<i>Australian</i>	DT_PP_RES_WFS_OHP	2,50
<i>German</i>	DT_RES_FFS_OHP	3,90
<i>Japanese</i>	DT_PP_RES_OHP	2,40
Geral	DT_PP_RES_OHP	3,70

Fonte: Autoria própria (2023)

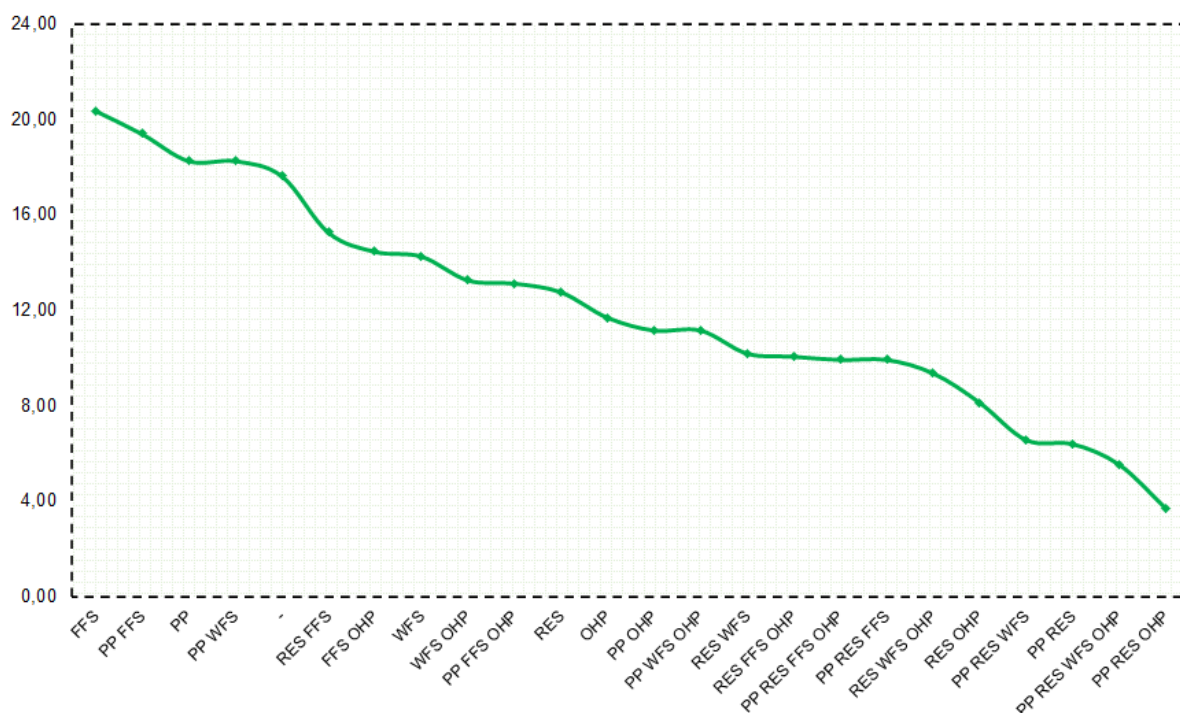


Figura 4.6 – Rank final médio de cada um dos modelos DT

Fonte: Autoria própria (2023)

É possível notar que, para o primeiro algoritmo de ML, já é evidenciado aumento geral de performance em relação ao mais popular modelo estatístico. Similarmente a LR, as etapas de pré-processamento de dados e *resampling* também demonstraram grande importância, contribuindo significativamente para o aumento de performance entre os modelos. Adicionalmente, desta vez, as etapas de FS e OHP demonstraram maior impacto, mais comumente incrementando a performance dos modelos no caso deste classificador.

4.3.3 Modelos SVM

Os resultados referentes a todos os modelos de SVM encontram-se na Tabela 4.8. Mais particularmente, foi utilizado o SVM com o *kernel Radial Basis Function* (RBF) ou SVM-RBF, como pode ser observado na Tabela 4.9, no parâmetro *solver*.

Tabela 4.8 – Resultados para os modelos SVM

<i>Dataset</i>	<i>Modelo</i>	<i>Acurácia</i>	<i>Erro Tipo I</i>	<i>Erro Tipo II</i>	<i>AUC</i>	<i>Rank Médio</i>	<i>Rank Final</i>
<i>Australian</i>	SVM	0,662	0,076	0,661	0,631	18,25	21,7
	SVM_WFS	0,855	0,203	0,070	0,864	14,75	10,5
	SVM_OHP	0,697	0,127	0,516	0,678	18,25	20,7
	SVM_WFS_OHP	0,857	0,203	0,069	0,864	14,00	9,6
	SVM_PP	0,862	0,145	0,123	0,866	12,50	13,8
	SVM_PP_WFS	0,862	0,145	0,123	0,866	12,50	13,8
	SVM_PP_OHP	0,872	0,139	0,107	0,877	8,75	10,5
	SVM_PP_WFS_OHP	0,872	0,139	0,107	0,877	8,75	10,5
	SVM_FFS	0,845	0,140	0,173	0,843	17,25	18,9
	SVM_FFS_OHP	0,864	0,163	0,100	0,869	12,25	11,9
	SVM_RES	0,682	0,102	0,533	0,682	17,50	20,6
	SVM_RES_WFS	0,858	0,202	0,080	0,859	15,25	12,1
	SVM_RES_OHP	0,721	0,125	0,431	0,722	16,50	19,2
	SVM_RES_WFS_OHP	0,863	0,193	0,080	0,863	13,25	10,9
	SVM_PP_FFS	0,859	0,142	0,139	0,860	15,00	17
	SVM_PP_FFS_OHP	0,855	0,197	0,075	0,864	14,00	10,2
	SVM_PP_RES	0,896	0,127	0,083	0,895	3,75	5,1
	SVM_PP_RES_WFS	0,891	0,155	0,061	0,892	6,50	4,2
	SVM_PP_RES_OHP	0,896	0,127	0,083	0,895	3,75	5,1
	SVM_PP_RES_WFS_OHP	0,894	0,155	0,055	0,895	5,00	2,6
SVM_RES_FFS	0,854	0,173	0,121	0,853	18,00	17,4	
SVM_RES_FFS_OHP	0,865	0,181	0,089	0,865	13,00	12,4	
SVM_PP_RES_FFS	0,885	0,151	0,084	0,882	8,25	8,3	
SVM_PP_RES_FFS_OHP	0,885	0,151	0,084	0,882	8,25	8,3	
<i>German</i>	SVM	0,707	0,017	0,938	0,523	17,50	21,4
	SVM_WFS	0,728	0,025	0,847	0,564	14,50	17,6
	SVM_OHP	0,710	0,016	0,932	0,526	16,25	20,1

Continuação da Tabela 4.8 – Resultados para os modelos SVM

	Modelo	Acurácia	Erro Tipo I	Erro Tipo II	AUC	Rank Médio	Rank Final
	SVM_WFS_OHP	0,728	0,025	0,847	0,564	14,50	17,6
	SVM_PP	0,767	0,084	0,581	0,667	10,50	12,4
	SVM_PP_WFS	0,767	0,084	0,581	0,667	10,50	12,4
	SVM_PP_OHP	0,774	0,105	0,511	0,692	9,50	10,8
	SVM_PP_WFS_OHP	0,774	0,105	0,511	0,692	9,50	10,8
	SVM_FFS	0,699	0,038	0,914	0,524	18,00	20,6
	SVM_FFS_OHP	0,704	0,168	0,596	0,618	18,25	17,7
	SVM_RES	0,550	0,166	0,732	0,551	20,00	19,8
	SVM_RES_WFS	0,768	0,150	0,313	0,769	9,50	9
	SVM_RES_OHP	0,793	0,094	0,320	0,793	6,75	7,9
	SVM_RES_WFS_OHP	0,803	0,141	0,252	0,804	6,25	5,5
	SVM_PP_FFS	0,727	0,126	0,621	0,627	16,00	17
	SVM_PP_FFS_OHP	0,733	0,133	0,582	0,642	14,50	15,2
	SVM_PP_RES	0,831	0,193	0,145	0,831	6,00	3,6
	SVM_PP_RES_WFS	0,756	0,256	0,232	0,756	11,00	7,8
	SVM_PP_RES_OHP	0,879	0,162	0,081	0,879	4,50	2,4
	SVM_PP_RES_WFS_OHP	0,779	0,256	0,186	0,779	8,50	5,6
	SVM_RES_FFS	0,634	0,425	0,306	0,635	18,00	14
	SVM_RES_FFS_OHP	0,745	0,293	0,217	0,745	11,75	7,9
	SVM_PP_RES_FFS	0,737	0,246	0,279	0,738	12,00	9,4
	SVM_PP_RES_FFS_OHP	0,737	0,246	0,279	0,738	12,00	9,4
	SVM	0,668	0,656	0,068	0,638	17,75	11,9
	SVM_WFS	0,860	0,060	0,208	0,866	14,25	14,5
	SVM_OHP	0,699	0,533	0,112	0,678	17,00	12,6
	SVM_WFS_OHP	0,863	0,052	0,208	0,870	12,75	13,5
	SVM_PP	0,869	0,096	0,162	0,871	12,00	11,2
	SVM_PP_WFS	0,869	0,096	0,162	0,871	12,00	11,2
	SVM_PP_OHP	0,873	0,112	0,140	0,874	9,50	7,2
	SVM_PP_WFS_OHP	0,873	0,112	0,140	0,874	9,50	7,2
	SVM_FFS	0,693	0,002	1,000	0,499	17,75	21,5
	SVM_FFS_OHP	0,701	0,006	0,963	0,516	17,00	20,6
	SVM_RES	0,678	0,541	0,102	0,679	17,00	11,6
	SVM_RES_WFS	0,868	0,050	0,215	0,868	13,25	15,5
	SVM_RES_OHP	0,725	0,397	0,152	0,725	17,00	14,2
	SVM_RES_WFS_OHP	0,871	0,043	0,215	0,871	10,50	13,6
	SVM_PP_FFS	0,860	0,060	0,208	0,866	14,25	14,5
	SVM_PP_FFS_OHP	0,860	0,060	0,208	0,866	14,25	14,5
	SVM_PP_RES	0,899	0,049	0,150	0,900	5,50	6
	SVM_PP_RES_WFS	0,896	0,045	0,158	0,899	6,00	7,2
	SVM_PP_RES_OHP	0,913	0,065	0,108	0,914	5,25	3,5
	SVM_PP_RES_WFS_OHP	0,903	0,046	0,144	0,905	4,50	5
	SVM_RES_FFS	0,857	0,059	0,228	0,857	17,50	19,4
	SVM_RES_FFS_OHP	0,867	0,053	0,215	0,866	15,00	17
	SVM_PP_RES_FFS	0,893	0,042	0,167	0,895	6,50	8,8
	SVM_PP_RES_FFS_OHP	0,893	0,042	0,167	0,895	6,50	8,8

Fonte: Autoria própria (2023)

Os demais parâmetros pertinentes e sua faixa de variação para a *grid search*, na etapa de OHP, também se encontram na Tabela 4.9. Já os melhores parâmetros para os modelos RF com OHP estão no Anexo D.

Tabela 4.9 – Parâmetros da SVM e amplitude das faixas de variação na *grid search*.

Parâmetro	Amplitude
C	[0.0001, 1000]
gamma	['scale', 'auto']
<i>solver</i>	['rbf']

Fonte: *scikit-learn*

Na Tabela 4.10 e na Figura 4.7, respectivamente, encontram-se os melhores modelos de SVM obtidos, bem como a variação de sua performance de acordo com as etapas incluídas em sua modelagem.

Tabela 4.3 – Resultados para os melhores modelos SVM

	Melhor Modelo	Rank Final
<i>Australian</i>	SVM_PP_RES_WFS_OHP	2,60
<i>German</i>	SVM_PP_RES_OHP	2,40
<i>Japanese</i>	SVM_PP_RES_OHP	3,50
Geral	SVM_PP_RES_OHP	3,67

Fonte: Autoria própria (2023)

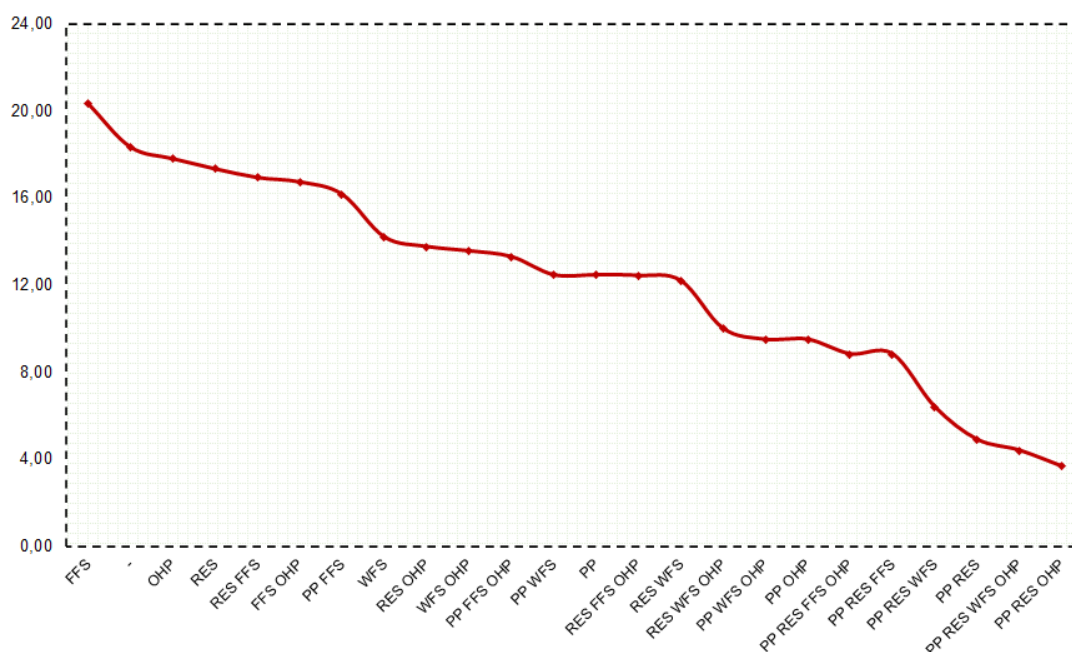


Figura 4.7 – Rank final médio de cada um dos modelos SVM

Fonte: Autoria própria (2023)

4.3.4 Modelos RF

Os resultados referentes a todos aos modelos RF encontram-se na Tabela 4.11.

Tabela 4.4 – Resultados para os modelos RF

<i>Dataset</i>	<i>Modelo</i>	<i>Acurácia</i>	<i>Erro Tipo I</i>	<i>Erro Tipo II</i>	<i>AUC</i>	<i>Rank Médio</i>	<i>Rank Final</i>
<i>Australian</i>	RF	0,872	0,111	0,146	0,872	13,50	14,2
	RF_WFS	0,855	0,189	0,089	0,861	17,00	12,8
	RF_OHP	0,887	0,106	0,118	0,888	8,75	8,7
	RF_WFS_OHP	0,859	0,191	0,074	0,867	15,25	9,9
	RF_PP	0,875	0,104	0,151	0,873	12,00	14,4
	RF_PP_WFS	0,875	0,104	0,151	0,873	12,00	14,4
	RF_PP_OHP	0,890	0,104	0,113	0,892	6,25	6,5
	RF_PP_WFS_OHP	0,890	0,104	0,113	0,892	6,25	6,5
	RF_FFS	0,843	0,127	0,192	0,841	21,50	22,8
	RF_FFS_OHP	0,880	0,088	0,159	0,877	10,00	13,8
	RF_RES	0,871	0,119	0,142	0,870	14,00	14,2
	RF_RES_WFS	0,849	0,156	0,147	0,849	19,75	18,5
	RF_RES_OHP	0,893	0,092	0,123	0,892	5,50	7
	RF_RES_WFS_OHP	0,869	0,094	0,170	0,868	14,50	17,4
	RF_PP_FFS	0,845	0,134	0,183	0,841	21,50	22,2
	RF_PP_FFS_OHP	0,862	0,127	0,148	0,862	17,75	17,7
	RF_PP_RES	0,908	0,093	0,092	0,907	3,50	4,2
	RF_PP_RES_WFS	0,893	0,131	0,085	0,892	7,50	5,2
	RF_PP_RES_OHP	0,915	0,084	0,085	0,915	1,75	2,5
	RF_PP_RES_WFS_OHP	0,897	0,125	0,082	0,897	5,75	3,7
RF_RES_FFS	0,829	0,160	0,180	0,830	23,00	22,8	
RF_RES_FFS_OHP	0,866	0,137	0,131	0,866	16,50	14,6	
RF_PP_RES_FFS	0,875	0,123	0,132	0,873	11,75	11,5	
RF_PP_RES_FFS_OHP	0,875	0,123	0,132	0,873	11,75	11,5	
<i>German</i>	RF	0,764	0,085	0,586	0,665	13,75	17,5
	RF_WFS	0,725	0,040	0,823	0,569	17,75	21,5
	RF_OHP	0,765	0,121	0,501	0,689	13,25	14,3
	RF_WFS_OHP	0,730	0,040	0,808	0,576	17,25	20,7
	RF_PP	0,762	0,089	0,582	0,665	14,00	17
	RF_PP_WFS	0,762	0,089	0,582	0,665	14,00	17
	RF_PP_OHP	0,765	0,122	0,499	0,690	12,75	12,9
	RF_PP_WFS_OHP	0,765	0,122	0,499	0,690	12,75	12,9
	RF_FFS	0,697	0,184	0,579	0,618	21,75	20,7
	RF_FFS_OHP	0,745	0,118	0,578	0,652	16,50	16,8
	RF_RES	0,848	0,095	0,204	0,850	5,50	4,6
	RF_RES_WFS	0,802	0,078	0,314	0,804	7,00	8,2
	RF_RES_OHP	0,857	0,093	0,190	0,859	4,00	3,2
	RF_RES_WFS_OHP	0,808	0,074	0,307	0,809	5,50	6,8
	RF_PP_FFS	0,698	0,183	0,579	0,619	20,50	19,6
	RF_PP_FFS_OHP	0,745	0,118	0,578	0,652	16,50	16,8

Continuação da Tabela 4.11 – Resultados para os modelos RF

	Modelo	Acurácia	Erro Tipo I	Erro Tipo II	AUC	Rank Médio	Rank Final
<i>Japanese</i>	RF_PP_RES	0,856	0,112	0,175	0,856	4,75	3,3
	RF_PP_RES_WFS	0,756	0,138	0,348	0,757	15,00	13,2
	RF_PP_RES_OHP	0,867	0,092	0,170	0,869	2,75	1,7
	RF_PP_RES_WFS_OHP	0,767	0,126	0,338	0,768	11,00	10,4
	RF_RES_FFS	0,760	0,183	0,293	0,762	13,25	10,3
	RF_RES_FFS_OHP	0,804	0,180	0,210	0,805	9,00	6,8
	RF_PP_RES_FFS	0,757	0,206	0,279	0,758	13,75	9,9
	RF_PP_RES_FFS_OHP	0,757	0,206	0,279	0,758	13,75	9,9
	RF	0,876	0,133	0,119	0,874	13,25	10,5
	RF_WFS	0,860	0,060	0,208	0,866	16,75	16,7
	RF_OHP	0,890	0,109	0,113	0,889	11,00	8,8
	RF_WFS_OHP	0,863	0,052	0,208	0,870	15,50	15,6
	RF_PP	0,876	0,133	0,119	0,874	13,25	10,5
	RF_PP_WFS	0,876	0,133	0,119	0,874	13,25	10,5
	RF_PP_OHP	0,894	0,122	0,094	0,892	10,00	7,2
	RF_PP_WFS_OHP	0,894	0,122	0,094	0,892	10,00	7,2
	RF_FFS	0,692	0,023	0,954	0,512	18,25	21,3
	RF_FFS_OHP	0,701	0,008	0,957	0,517	17,75	21,3
	RF_RES	0,884	0,110	0,123	0,883	12,75	11,3
	RF_RES_WFS	0,871	0,043	0,215	0,871	14,75	16,7
	RF_RES_OHP	0,906	0,096	0,091	0,906	6,25	3,9
	RF_RES_WFS_OHP	0,873	0,039	0,215	0,873	14,00	16,2
	RF_PP_FFS	0,860	0,060	0,208	0,866	16,75	16,7
	RF_PP_FFS_OHP	0,860	0,060	0,208	0,866	16,75	16,7
	RF_PP_RES	0,913	0,078	0,094	0,914	5,75	3,9
	RF_PP_RES_WFS	0,896	0,037	0,167	0,898	6,25	7,9
	RF_PP_RES_OHP	0,926	0,074	0,074	0,926	4,50	2,4
	RF_PP_RES_WFS_OHP	0,899	0,031	0,167	0,901	5,50	7,4
	RF_RES_FFS	0,867	0,052	0,215	0,867	16,50	18
	RF_RES_FFS_OHP	0,868	0,049	0,215	0,868	15,75	17,5
RF_PP_RES_FFS	0,896	0,037	0,167	0,898	6,25	7,9	
RF_PP_RES_FFS_OHP	0,896	0,037	0,167	0,898	6,25	7,9	

Fonte: Autoria própria (2023)

Como em LR e nas DTs, para as RFs, o parâmetro *random_state* foi fixado. Os demais parâmetros pertinentes e sua faixa de variação para a *grid search* encontram-se na Tabela 4.12. Já os melhores parâmetros para os modelos RF com OHP estão no Anexo E.

Tabela 4.5 – Parâmetros da RF e amplitude das faixas de variação na *grid search*.

Parâmetro	Amplitude
<i>criterion</i>	['gini', 'entropy']
<i>n_estimators</i>	[10, 200]
<i>max_depth</i>	[None; 5, 50]
<i>min_samples_split</i>	[2, 20]
<i>min_samples_leaf</i>	[1, 10]
<i>max_features</i>	[None, 'sqrt', 'log2']
<i>bootstrap</i>	[True, False]

Fonte: *scikit-learn*

Na Tabela 4.13 e na Figura 4.8, respectivamente, encontram-se os melhores modelos de RF obtidos, bem como a variação de sua performance de acordo com as etapas incluídas em suas respectivas modelagens.

Tabela 4.6 – Resultados para os melhores modelos RF

	Melhor Modelo	Rank Final
<i>Australian</i>	RF_PP_RES_OHP	2,50
<i>German</i>	RF_PP_RES_OHP	1,70
<i>Japanese</i>	RF_PP_RES_OHP	2,40
Geral	RF_PP_RES_OHP	2,20

Fonte: Autoria própria (2023)

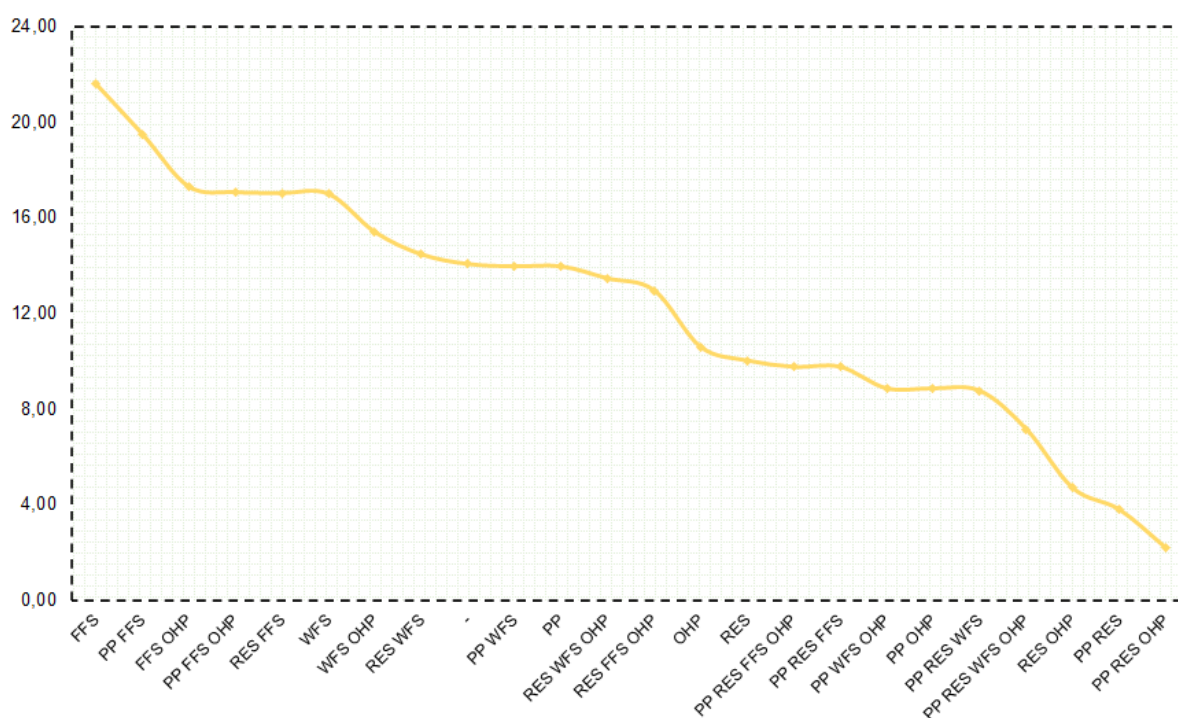


Figura 4.8 – Rank final médio de cada um dos modelos RF

Fonte: Autoria própria (2023)

Como esperado, conhecida a definição do nosso primeiro algoritmo de classificação *ensemble*, o comportamento dos modelos RF não é muito diferente dos modelos de DT. Cabe apontar que, para todos os *datasets*, o melhor modelo de RF foi o mesmo que, assim como nas DTs, conta com as etapas de pré-processamento de dados, *resampling* e OHP.

4.3.5 Modelos AdaBoost

Os resultados referentes a todos aos modelos AdaBoost encontram-se na Tabela 4.14.

Tabela 4.7 – Resultados para os modelos AdaBoost

<i>Dataset</i>	<i>Modelo</i>	<i>Acurácia</i>	<i>Erro Tipo I</i>	<i>Erro Tipo II</i>	<i>AUC</i>	<i>Rank Médio</i>	<i>Rank Final</i>
<i>Australian</i>	BOOST	0,814	0,147	0,228	0,812	20,75	21,5
	BOOST_WFS	0,859	0,154	0,122	0,862	14,25	12,3
	BOOST_OHP	0,872	0,139	0,102	0,879	8,50	6,4
	BOOST_WFS_OHP	0,867	0,146	0,114	0,870	11,75	10,5
	BOOST_PP	0,813	0,150	0,228	0,811	21,50	22
	BOOST_PP_WFS	0,813	0,150	0,228	0,811	21,50	22
	BOOST_PP_OHP	0,874	0,136	0,108	0,878	7,75	6,7
	BOOST_PP_WFS_OHP	0,874	0,136	0,108	0,878	7,75	6,7
	BOOST_FFS	0,839	0,130	0,196	0,837	16,00	18
	BOOST_FFS_OHP	0,875	0,085	0,172	0,872	8,75	12,7
	BOOST_RES	0,819	0,172	0,189	0,820	20,50	20
	BOOST_RES_WFS	0,843	0,202	0,111	0,843	16,50	13,2
	BOOST_RES_OHP	0,885	0,103	0,129	0,884	6,00	8
	BOOST_RES_WFS_OHP	0,852	0,195	0,099	0,853	14,75	10,7
	BOOST_PP_FFS	0,832	0,140	0,202	0,829	18,50	19,6
	BOOST_PP_FFS_OHP	0,867	0,100	0,163	0,868	10,25	12,5
	BOOST_PP_RES	0,876	0,123	0,123	0,877	7,75	9,1
	BOOST_PP_RES_WFS	0,896	0,131	0,080	0,895	4,50	3,2
	BOOST_PP_RES_OHP	0,912	0,085	0,091	0,912	1,75	2,1
	BOOST_PP_RES_WFS_OHP	0,898	0,134	0,071	0,898	4,00	2,4
<i>German</i>	BOOST_RES_FFS	0,829	0,172	0,170	0,829	19,75	18,7
	BOOST_RES_FFS_OHP	0,860	0,131	0,146	0,862	12,50	13,4
	BOOST_PP_RES_FFS	0,857	0,122	0,167	0,856	12,75	14,5
	BOOST_PP_RES_FFS_OHP	0,874	0,122	0,134	0,872	9,25	10,7
	BOOST	0,669	0,258	0,496	0,623	20,25	17,3
	BOOST_WFS	0,727	0,037	0,823	0,570	17,00	21,2
	BOOST_OHP	0,767	0,107	0,526	0,683	11,50	13,6
	BOOST_WFS_OHP	0,730	0,040	0,808	0,576	16,50	20,4
	BOOST_PP	0,675	0,249	0,500	0,626	19,00	16,8
	BOOST_PP_WFS	0,675	0,249	0,500	0,626	19,00	16,8
BOOST_PP_OHP	0,752	0,128	0,526	0,673	13,25	14,9	

Continuação da Tabela 4.14 – Resultados para os modelos AdaBoost

	Modelo	Acurácia	Erro Tipo I	Erro Tipo II	AUC	Rank Médio	Rank Final
	BOOST_PP_WFS_OHP	0,752	0,128	0,526	0,673	13,25	14,9
	BOOST_FFS	0,702	0,179	0,575	0,623	19,00	20,2
	BOOST_FFS_OHP	0,738	0,129	0,572	0,650	15,25	16,9
	BOOST_RES	0,764	0,234	0,234	0,766	12,00	8,8
	BOOST_RES_WFS	0,807	0,074	0,309	0,808	5,75	7,5
	BOOST_RES_OHP	0,859	0,083	0,197	0,860	2,75	2,3
	BOOST_RES_WFS_OHP	0,812	0,058	0,316	0,813	5,25	7,5
	BOOST_PP_FFS	0,695	0,189	0,578	0,616	20,00	21,2
	BOOST_PP_FFS_OHP	0,738	0,129	0,572	0,650	15,25	16,9
	BOOST_PP_RES	0,780	0,220	0,215	0,783	8,75	5,9
	BOOST_PP_RES_WFS	0,762	0,193	0,278	0,765	12,50	11
	BOOST_PP_RES_OHP	0,866	0,086	0,181	0,867	2,25	1,5
	BOOST_PP_RES_WFS_OHP	0,774	0,196	0,253	0,776	9,50	8
	BOOST_RES_FFS	0,771	0,172	0,282	0,773	9,50	9,4
	BOOST_RES_FFS_OHP	0,803	0,174	0,219	0,804	6,75	5,3
	BOOST_PP_RES_FFS	0,765	0,208	0,259	0,766	12,25	10,3
	BOOST_PP_RES_FFS_OHP	0,770	0,208	0,251	0,770	10,50	8,4
	BOOST	0,794	0,240	0,180	0,790	18,50	16,2
	BOOST_WFS	0,863	0,052	0,208	0,870	13,50	14
	BOOST_OHP	0,887	0,116	0,113	0,886	9,75	7,3
	BOOST_WFS_OHP	0,860	0,060	0,208	0,866	14,75	15,1
	BOOST_PP	0,794	0,240	0,180	0,790	18,50	16,2
	BOOST_PP_WFS	0,794	0,240	0,180	0,790	18,50	16,2
	BOOST_PP_OHP	0,893	0,122	0,097	0,890	8,75	5,7
	BOOST_PP_WFS_OHP	0,893	0,122	0,097	0,890	8,75	5,7
	BOOST_FFS	0,690	0,025	0,954	0,510	18,50	21,8
	BOOST_FFS_OHP	0,696	0,023	0,940	0,519	17,50	20,8
<i>Japanese</i>	BOOST_RES	0,848	0,157	0,148	0,848	16,50	13,2
	BOOST_RES_WFS	0,876	0,036	0,212	0,876	11,00	14,2
	BOOST_RES_OHP	0,895	0,093	0,118	0,894	8,25	6,5
	BOOST_RES_WFS_OHP	0,876	0,036	0,212	0,876	11,00	14,2
	BOOST_PP_FFS	0,860	0,060	0,208	0,866	14,75	15,1
	BOOST_PP_FFS_OHP	0,860	0,060	0,208	0,866	14,75	15,1
	BOOST_PP_RES	0,878	0,116	0,125	0,880	11,00	8,8
	BOOST_PP_RES_WFS	0,899	0,044	0,156	0,900	5,50	6,4
	BOOST_PP_RES_OHP	0,919	0,083	0,078	0,920	4,50	2,4
	BOOST_PP_RES_WFS_OHP	0,901	0,044	0,151	0,903	4,75	5,5
	BOOST_RES_FFS	0,867	0,052	0,215	0,867	15,00	17,4
	BOOST_RES_FFS_OHP	0,868	0,049	0,215	0,868	14,25	16,9
	BOOST_PP_RES_FFS	0,896	0,037	0,167	0,898	5,75	7,1
	BOOST_PP_RES_FFS_OHP	0,896	0,037	0,167	0,898	5,75	7,1

Fonte: Autoria própria (2023)

Para o *ensemble* do tipo *boosting*, foi escolhido o algoritmo AdaBoost, sendo que os *base learners* foram as DTs modeladas anteriormente. No caso dos modelos AdaBoost com a etapa de OHP, foram agregadas as DTs com os hiper parâmetros ótimos encontrados na *grid search* para estes modelos (Anexo C). Foi também fixado o parâmetro *random state* e, por fim, variados alguns outros hiper parâmetros, mais especificamente: *n_estimators* (amplitude de 10 a 500) e *learning_rate* (amplitude de 0.0001, 10).

Apresentando os melhores resultados em geral, o mesmo modelo AdaBoost, mais especificamente BOOST_PP_RES_OHP, ou seja, uma agregação por *boosting* de DTs com pré-processamento dos dados, *resampling* e hiper parâmetros otimizados, teve as melhores métricas de performance em todas os *datasets* entre os modelos AdaBoost (Tabela 4.15).

Tabela 4.8 – Resultados para os melhores modelos AdaBoost

	Melhor Modelo	Rank Final
<i>Australian</i>	BOOST_PP_RES_OHP	2,10
<i>German</i>	BOOST_PP_RES_OHP	1,50
<i>Japanese</i>	BOOST_PP_RES_OHP	2,40
Geral	BOOST_PP_RES_OHP	2,00

Fonte: Autoria própria (2023)

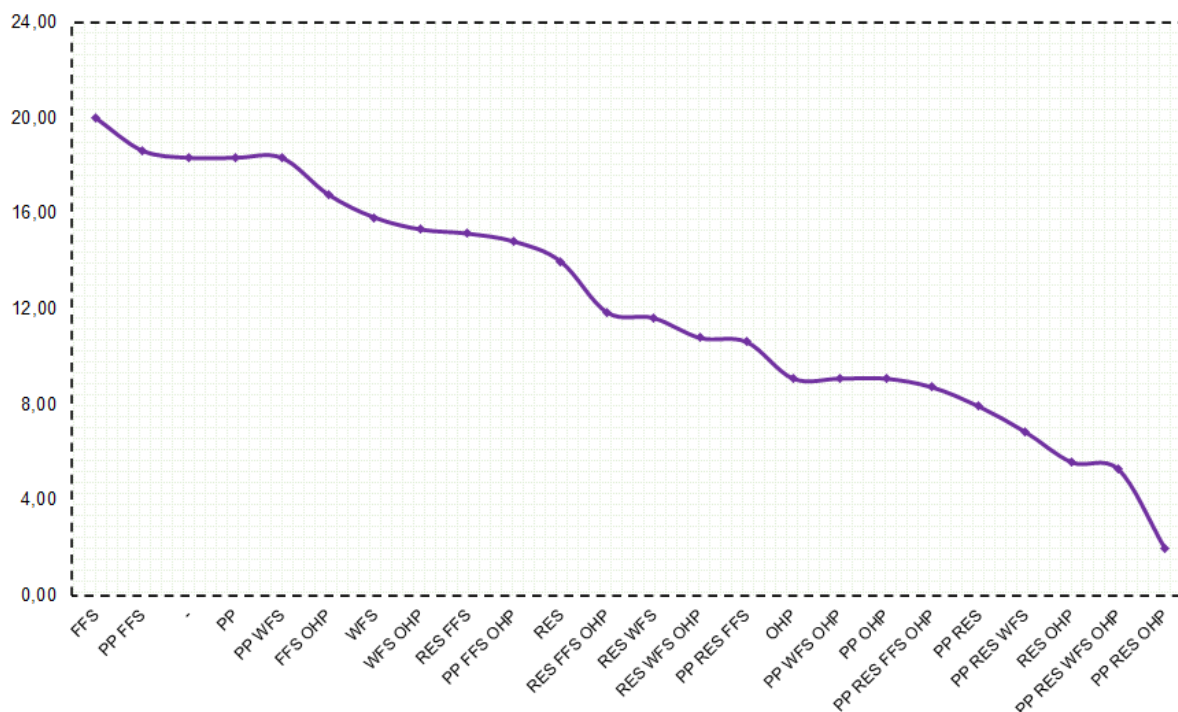


Figura 4.9 – Rank final médio de cada um dos modelos AdaBoost

Fonte: Autoria própria (2023)

4.3.6 Modelos de *Stacking*

Finalmente, os resultados referentes a todos os modelos de agregação por *stacking* encontram-se na Tabela 4.16.

Tabela 4.9 – Resultados para os modelos de *stacking*

<i>Dataset</i>	<i>Modelo</i>	<i>Acurácia</i>	<i>Erro Tipo I</i>	<i>Erro Tipo II</i>	<i>AUC</i>	<i>Rank Médio</i>	<i>Rank Final</i>
<i>Australian</i>	STACK	0,855	0,107	0,191	0,851	16,25	20,1
	STACK_WFS	0,849	0,168	0,131	0,850	19,25	16,3
	STACK_OHP	0,872	0,136	0,118	0,873	10,00	8,8
	STACK_WFS_OHP	0,851	0,165	0,124	0,855	17,25	14,1
	STACK_PP	0,854	0,127	0,170	0,852	16,75	19,3
	STACK_PP_WFS	0,854	0,129	0,166	0,852	17,00	18,4
	STACK_PP_OHP	0,877	0,127	0,113	0,880	7,75	7,3
	STACK_PP_WFS_OHP	0,864	0,134	0,132	0,867	13,00	12,2
	STACK_FFS	0,848	0,133	0,169	0,849	20,25	21,1
	STACK_FFS_OHP	0,868	0,106	0,160	0,867	11,50	14,6
	STACK_RES	0,871	0,103	0,155	0,871	9,75	12,9
	STACK_RES_WFS	0,865	0,129	0,143	0,864	13,00	13,6
	STACK_RES_OHP	0,882	0,103	0,132	0,883	6,75	8,9
	STACK_RES_WFS_OHP	0,861	0,129	0,153	0,859	14,75	15,7
	STACK_PP_FFS	0,841	0,138	0,183	0,839	22,25	22,9
	STACK_PP_FFS_OHP	0,851	0,155	0,145	0,850	19,50	18,2
	STACK_PP_RES	0,887	0,116	0,111	0,887	5,00	5,2
	STACK_PP_RES_WFS	0,883	0,136	0,101	0,882	7,75	5,5
	STACK_PP_RES_OHP	0,898	0,101	0,107	0,896	2,00	3
	STACK_PP_RES_WFS_OHP	0,890	0,124	0,098	0,889	3,25	2,5
	STACK_RES_FFS	0,852	0,146	0,147	0,854	18,00	17
	STACK_RES_FFS_OHP	0,866	0,178	0,089	0,866	12,50	8
	STACK_PP_RES_FFS	0,872	0,140	0,119	0,870	11,75	10,3
	STACK_PP_RES_FFS_OHP	0,889	0,125	0,101	0,887	4,50	4
<i>German</i>	STACK	0,747	0,129	0,541	0,665	16,50	16,6
	STACK_WFS	0,712	0,115	0,721	0,582	21,00	22,8
	STACK_OHP	0,775	0,102	0,512	0,693	11,25	12,9
	STACK_WFS_OHP	0,734	0,077	0,704	0,610	16,75	20,5
	STACK_PP	0,758	0,110	0,550	0,670	15,00	16,6
	STACK_PP_WFS	0,760	0,108	0,549	0,672	13,75	15,5
	STACK_PP_OHP	0,759	0,114	0,535	0,675	14,00	14,6
	STACK_PP_WFS_OHP	0,779	0,097	0,509	0,697	9,75	11,7
	STACK_FFS	0,726	0,138	0,597	0,633	19,75	19,3
	STACK_FFS_OHP	0,730	0,126	0,612	0,631	19,00	20
	STACK_RES	0,834	0,114	0,213	0,836	6,50	5,6
	STACK_RES_WFS	0,789	0,143	0,278	0,790	11,00	9,8
	STACK_RES_OHP	0,890	0,071	0,148	0,891	1,50	1,6
	STACK_RES_WFS_OHP	0,870	0,042	0,218	0,870	3,25	4,3
	STACK_PP_FFS	0,726	0,130	0,609	0,631	19,75	20,1
	STACK_PP_FFS_OHP	0,733	0,105	0,648	0,623	17,50	20,2

Continuação da Tabela 4.16 – Resultados para os modelos de *stacking*

	Modelo	Acurácia	Erro Tipo I	Erro Tipo II	AUC	Rank Médio	Rank Final
Japanese	STACK_PP_RES	0,848	0,121	0,182	0,849	6,50	5
	STACK_PP_RES_WFS	0,709	0,130	0,443	0,714	16,25	13,3
	STACK_PP_RES_OHP	0,886	0,100	0,128	0,886	2,75	1,9
	STACK_PP_RES_WFS_OHP	0,733	0,075	0,455	0,735	11,25	11,5
	STACK_RES_FFS	0,775	0,223	0,226	0,776	13,00	10,2
	STACK_RES_FFS_OHP	0,798	0,222	0,179	0,800	9,25	6,1
	STACK_PP_RES_FFS	0,777	0,222	0,222	0,778	11,75	9,1
	STACK_PP_RES_FFS_OHP	0,774	0,202	0,250	0,774	13,00	10,8
	STACK	0,863	0,156	0,124	0,860	15,75	12,7
	STACK_WFS	0,863	0,053	0,208	0,870	14,75	15,7
	STACK_OHP	0,887	0,109	0,117	0,887	10,25	8,3
	STACK_WFS_OHP	0,864	0,050	0,208	0,871	13,25	14,5
	STACK_PP	0,852	0,176	0,126	0,849	18,25	14,5
	STACK_PP_WFS	0,854	0,172	0,126	0,851	17,50	14
	STACK_PP_OHP	0,888	0,112	0,112	0,888	10,00	7,6
	STACK_PP_WFS_OHP	0,888	0,122	0,105	0,886	10,25	7,3
	STACK_FFS	0,665	0,069	0,941	0,495	20,75	22,3
	STACK_FFS_OHP	0,701	0,006	0,963	0,516	17,75	21,3
	STACK_RES	0,875	0,120	0,133	0,874	12,75	11,3
	STACK_RES_WFS	0,871	0,039	0,218	0,871	12,75	16,1
	STACK_RES_OHP	0,897	0,099	0,108	0,896	7,75	6,1
	STACK_RES_WFS_OHP	0,870	0,042	0,218	0,870	13,75	16,7
	STACK_PP_FFS	0,845	0,111	0,191	0,849	19,00	18,2
	STACK_PP_FFS_OHP	0,860	0,060	0,208	0,866	15,75	16,5
	STACK_PP_RES	0,906	0,084	0,103	0,907	4,75	3,1
	STACK_PP_RES_WFS	0,899	0,031	0,167	0,901	5,50	7,4
	STACK_PP_RES_OHP	0,913	0,087	0,086	0,914	4,25	2,3
	STACK_PP_RES_WFS_OHP	0,900	0,028	0,167	0,902	4,75	6,9
	STACK_RES_FFS	0,854	0,145	0,172	0,841	19,25	18,1
	STACK_RES_FFS_OHP	0,868	0,049	0,215	0,868	14,50	17
	STACK_PP_RES_FFS	0,894	0,040	0,167	0,897	7,50	8,6
	STACK_PP_RES_FFS_OHP	0,896	0,037	0,167	0,898	6,50	8

Fonte: Autoria própria (2023)

Para nosso único exemplo de um *ensemble* heterogêneo, a agregação por *stacking* foi feita tendo como *base learners* LR e ambos os algoritmos individuais de ML modelados anteriormente (DT e SVM), assim como o *ensemble* homogêneo RF. Como no caso dos modelos AdaBoost, para os modelos de *stacking* com OHP, foram utilizados como *base learners* os referidos modelos anteriores, já com os parâmetros ajudados conforme suas respectivas *grid searches*.

Para a combinação dos *base learners*, a classificação final (parâmetro *final_estimator*) foi feita por meio do algoritmo GBM. O único parâmetro variado foi o *stack_method*, que

implementa três formas distintas para a agregação (*'predict_proba'*, *'decision_function'* e *'predict'*). Na Tabela 4.17 encontram-se os melhores modelos de *stacking* e suas respectivas posições nos *rankings*.

Tabela 4.10 – Resultados para os melhores modelos de *stacking*

	Melhor Modelo	Rank Final
<i>Australian</i>	STACK_PP_RES_WFS_OHP	2,50
<i>German</i>	STACK_RES_OHP	1,60
<i>Japanese</i>	STACK_PP_RES_OHP	2,30
Geral	STACK_PP_RES_OHP	2,40

Fonte: A autoria própria (2023)

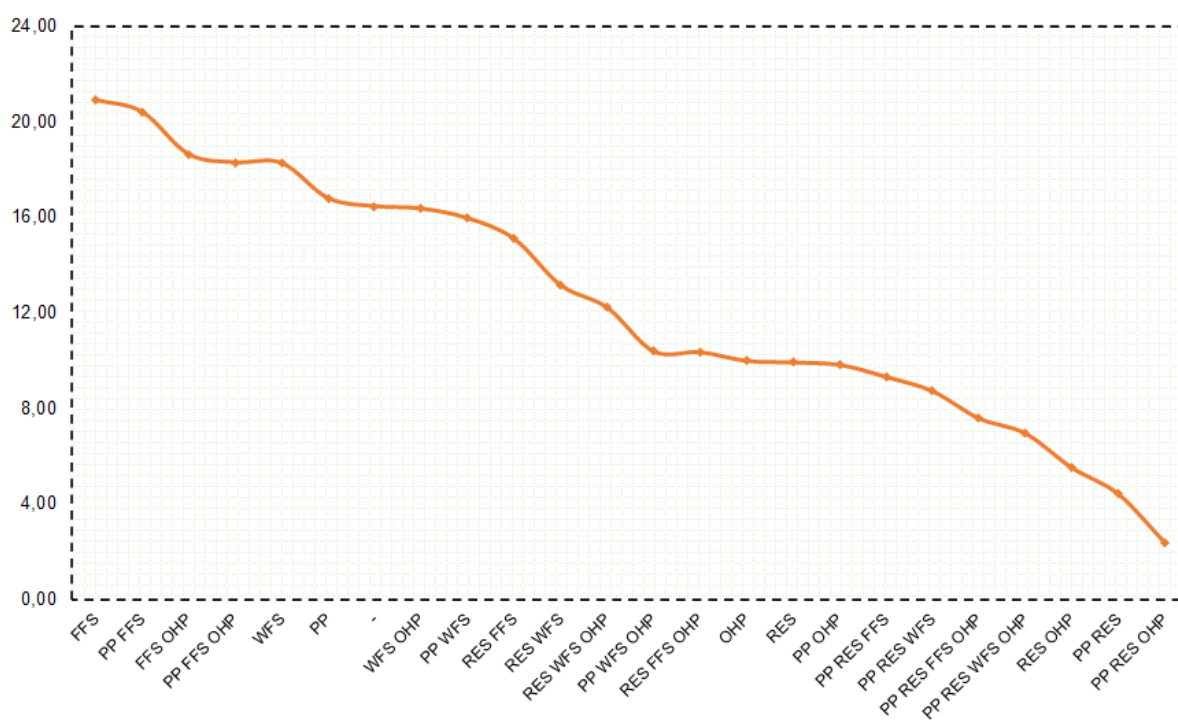


Figura 4.10 – Rank final médio de cada um dos modelos de *stacking*

Fonte: A autoria própria (2023)

Como foi o caso de todos os modelos de ML, as etapas de de pré-processamento de dados, *resampling* e OHP mantiveram-se essenciais para os modelos de melhor performance e, verdadeiramente, pode-se notar que uma modelagem diligente de CSMs, considerando cada etapa e suas alternativas, é muito mais importante em termos de performance do que a escolha do classificador.

5. ANÁLISE DOS RESULTADOS E DESENVOLVIMENTO DOS FRAMEWORKS

5.1 ANÁLISE GERAL DOS MODELOS

Além de analisar os modelos individualmente, evidenciando a performance de cada classificador e o efeito que cada etapa tem sobre ele, apresentam-se aqui alguns dados sobre o resultado em geral nas Tabelas 5.1-5.4.

Tabela 5.1 – Resultados médios dos modelos por *dataset*

Dataset	Média - Erro Tipo I	Média - Erro Tipo II	Média - Acurácia	Média - AUC
<i>Australian</i>	0,14	0,14	0,86	0,86
<i>German</i>	0,16	0,43	0,75	0,71
<i>Japanese</i>	0,09	0,23	0,85	0,84
Geral	0,13	0,27	0,82	0,80

Fonte: Autoria própria (2023)

Segundo a Tabela 5.1, analisando os resultados médios divididos por métrica de performance e *dataset*, já pode ser evidenciado o potencial de discriminação dos modelos. Similarmente aos resultados da literatura descritos nas Tabelas 3.2-3.4, em ordem respectiva, os modelos de melhor performance foram aqueles treinados e avaliados nos *datasets* *australian*, *japanese* e *german*. Contudo, apesar dos valores médios de AUC, segundo a Tabela 3.1, significarem poder de discriminação satisfatório, encontram-se ainda um pouco distantes dos melhores modelos avaliados pelos artigos da RSL nos mesmos *datasets*. Verifica-se, então, nas Tabelas 5.2-5.4, quais foram os melhores modelos por métrica e *dataset*.

Tabela 5.2 – Resultados dos melhores modelos por métrica no *dataset* UCI *australian*

Métrica	Modelo	Resultado
Erro Tipo I	SVM	0,01
	DT_FFS_OHP	
	RF_PP_RES_OHP	
Erro Tipo II	BOOST_FFS_OHP	0,06
	BOOS_PP_RES_OHP	
	SVM_PP_RES_WFS	
Acurácia	SVM_PP_RES_WFS_OHP	0,91
	RF_PP_RES_OHP	
AUC	BOOST_PP_RES_OHP	0,92
	RF_PP_RES	

Fonte: Autoria própria (2023)

A Tabela 5.2 ilustra como diferentes modelos podem se destacar em diferentes métricas de avaliação. De fato, apesar dos modelos individuais DT e SVM apresentarem bons resultados

para as métricas de Erro Tipo I e II (Modelos SVM tiveram, inclusive, o menor Erro Tipo II), é possível perceber que vigoram, em maior parte, algoritmos *ensemble*. O mesmo comportamento pode ser observado a seguir nas Tabelas 5.3-5.4, referentes aos demais *datasets*.

Tabela 5.3 – Resultados dos melhores modelos por métrica no *dataset* UCI *german*

Métrica	Modelo	Resultado
Erro Tipo I	STACK_PP_RES_OHP	0,10
Erro Tipo II	STACK_RES_OHP	0,07
Acurácia	STACK_RES_OHP	0,89
	STACK_PP_RES_OHP	
AUC	STACK_RES_OHP	0,89
	STACK_PP_RES_OHP	

Fonte: Autoria própria (2023)

Tabela 5.4 – Resultados dos melhores modelos por métrica no *dataset* UCI *japanese*

Métrica	Modelo	Resultado
Erro Tipo I	RF_PP_RES_OHP	0,07
Erro Tipo II	RF_PP_RES_OHP	0,07
Acurácia	RF_PP_RES_OHP	0,93
AUC	RF_PP_RES_OHP	0,93

Fonte: Autoria própria (2023)

Na verdade, para o *datasets german* e *japanese*, a prevalência de modelos *ensemble* é ainda mais evidente, sendo que os mesmos modelos (respectivamente, um *ensemble* com agregação por *stacking*, e RF) detém os melhores resultados para todas as métricas de performance consideradas.

Apesar de conhecer os melhores modelos de acordo com as métricas escolhidas, considerando que o objetivo é avaliar CSMs (onde existe um custo diferente associado a cada tipo de erro), ainda é difícil fazer quaisquer afirmações sobre quais os modelos mais adequados. Considerando os *rankings* ponderados elaborados por *dataset*, buscando explorar melhor como se comportam os modelos, a pontuação dos modelos por tipo de classificador e *dataset* pode ser vista nas Figuras 5.1-5.3.

Nos gráficos contidos nas figuras, os modelos, divididos por tipo (estatístico, ML individual e *ensembles*), se encontram distribuídos conforme o seu *rank* final. Apesar de ainda destacarem-se os modelos *ensemble*, que mais aparecem na região inferior dos gráficos (portanto, melhores nos *rankings*), é possível notar agora que há pouca distinção entre os melhores modelos para cada tipo de classificador. Na verdade, a performance dos diferentes tipos de modelos varia muito como um todo, e *ensembles* encontram-se tanto entre as melhores quanto as piores alternativas.

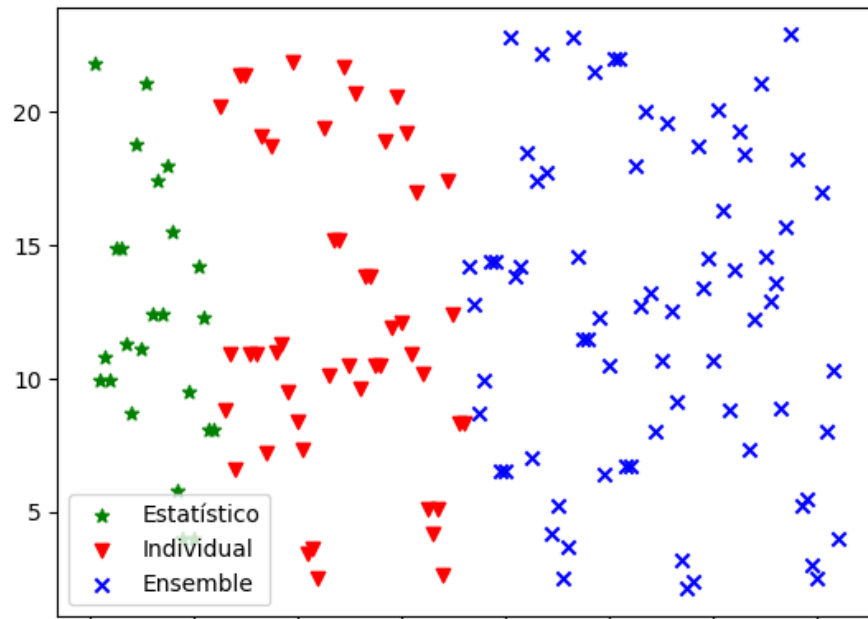


Figura 5.1 – Rank final dos modelos no *dataset* UCI *australian*
 Fonte: Autoria própria (2023)

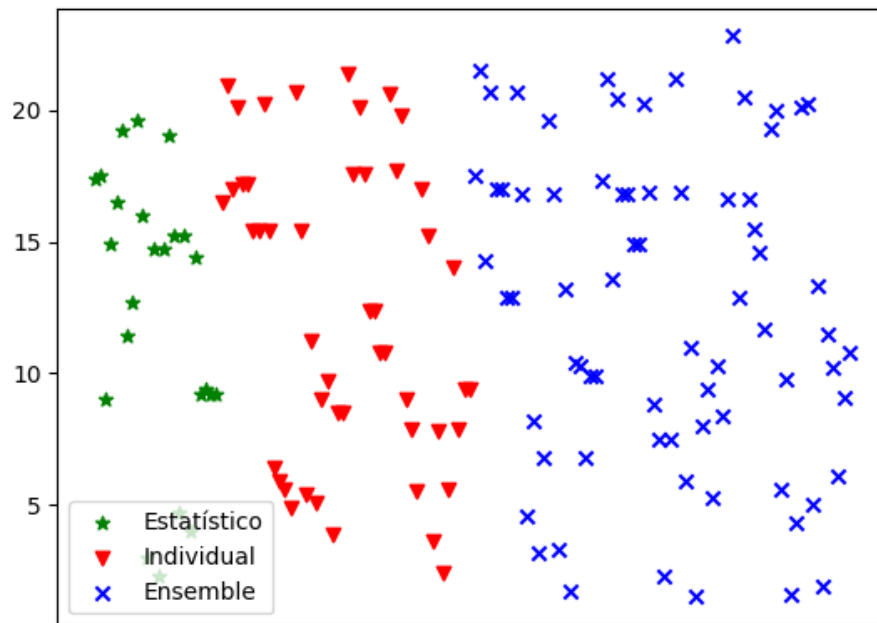


Figura 5.2 – Rank final dos modelos no *dataset* UCI *german*
 Fonte: Autoria própria (2023)

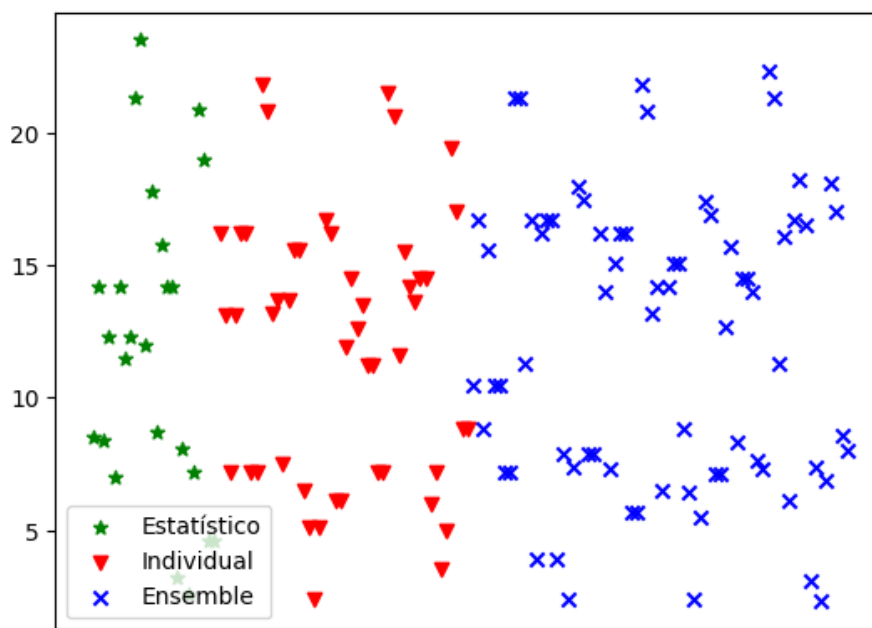


Figura 5.3 – Rank final dos modelos no *dataset UCI japanese*
 Fonte: Autoria própria (2023)

Esta ampla faixa de variação do desempenho dos modelos evidencia o efeito das etapas de modelagem, que gradualmente incrementam e aperfeiçoam os modelos, aumentando sua performance, o que pode ser verificado também nas Tabelas 4.2-4.17, e graficamente nas Figuras 4.5-4.10. Contudo, ainda existe interesse em distinguir estas etapas, o que ainda não é possível. Observando as Tabelas 5.2 e 5.3, por exemplo, nota-se que os modelos SVM_PP_RES_WFS e SVM_PP_RES_WFS_OHP, assim como STACK_RES_OHP e STACK_PP_RES_OHP demonstram o mesmo resultado, respectivamente, para Erro Tipo II e acurácia. Ou seja, a etapa de OHP não melhorou a performance do modelo SVM no *dataset australian*, e o pré-processamento não beneficiou a agregação por *stacking* no *dataset german*.

Para contornar este problema, além de contribuir para a construção dos *frameworks*, foi conduzida uma análise de importância das etapas, muito similar aos métodos de FS já discutidos e utilizados nos modelos. Neste caso, no entanto, foi utilizado o algoritmo *ensemble* RF (uma agregação de 100 DTs) para uma regressão. Desta forma, tentando prever a posição de um modelo no *ranking* com base nas etapas de modelagem que o compõe, o algoritmo é capaz de avaliar quais delas mais impactam a variável dependente (sua posição no *ranking* final).

A análise de importância de etapas foi conduzida por algoritmo de classificação, ilustrando quais etapas mais impactaram cada tipo de classificador, e encontra-se na Figura 5.4.

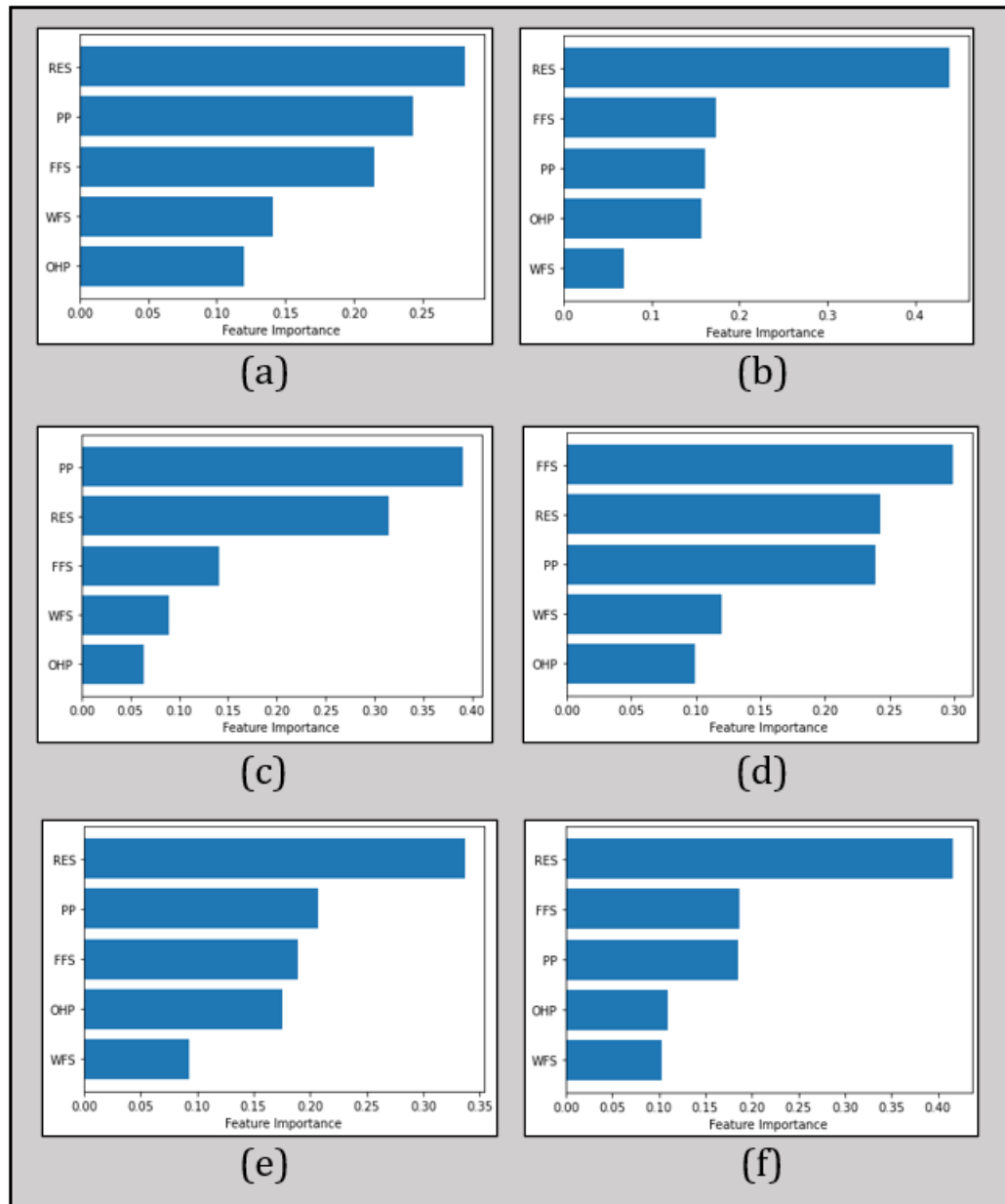


Figura 5.4 – Resultado da análise de importância de etapas por meio de RF para (a) LR (b) DT (c) SVM (d) RF (e) AdaBoost (f) *Stacking*

Fonte: Autoria própria (2023)

Sabendo agora do impacto que cada uma das discutidas etapas de modelagem para CSMs têm sobre cada um dos algoritmos de classificação estudados, temos um ponto de partida para o desenvolvimento dos *frameworks*.

5.2 FRAMEWORKS PROPOSTOS

Buscando desenvolver CSMs em acordo com os requisitos estipulados (especialmente os do Acordo de Basileia II), dentro do escopo dos modelos discutidos no Capítulo 4 e os seus resultados, e priorizando o balanço entre desempenho e interpretabilidade, uma alternativa simples e direta se encontra no primeiro *framework* desenvolvido, na Figura 5.5.

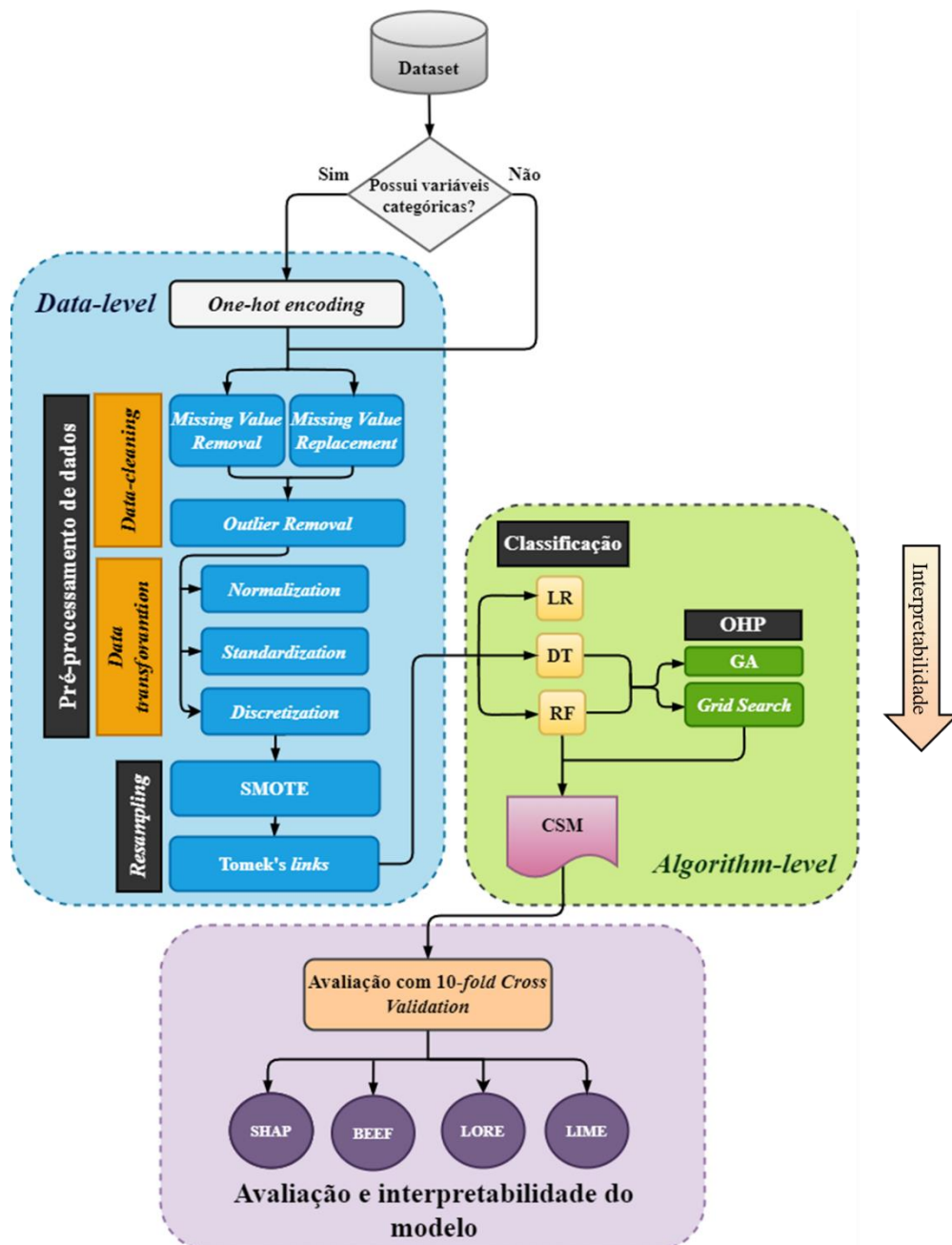


Figura 5.5 – Framework de modelagem de CSMs para aplicação prática
Fonte: Autoria própria (2023)

O primeiro passo, no *data-level*, é preparar os dados para as análises subsequentes. Sem transformar as variáveis categóricas e remover ou substituir amostras inadequadas (respectivamente, nas etapas de *one-hot encoding* e *missing value replacement* ou *removal*), sequer é possível partir para a classificação. Concluídas estas etapas preliminares, considerando os classificadores selecionados, é sugerível que a modelagem inclua o que foi referido neste estudo como pré-processamento de dados. Removendo o que forem considerados *outliers*, e aplicando alguma forma de transformação dos valores dos atributos (seja por *normalization*, *standardization* ou *discretization*), os resultados, assim como a própria literatura, sugerem um consistente impacto positivo no desempenho dos modelos.

Ainda no *data-level*, já pode ser realizado o balanceamento de classes, combinando os pontos positivos tanto das técnicas de *oversampling*, como SMOTE, quanto das de *undersampling* (e.g. Tomek's links), e lidando com o desbalanceamento logo em seguida do pré-processamento, antes da etapa de classificação. A não ser que não exista desbalanceamento (o que é improvável no caso de bases de dados reais), trata-se de etapa essencial, agregando positivamente os modelos até mesmo nos *datasets* com menor grau de desequilíbrio entre classes, como atesta o seu destaque na Figura 5.4. Prezando pela simplicidade do modelo, se for preferível ater-se a uma única técnica de balanceamento de dados (seja pelo seu volume ou até pela interpretabilidade), recomenda-se a ordem com que aparecem no *framework*, em consonância com Marqués, García e Sánchez (2013a), e suas conclusões sobre as vantagens dos métodos de *oversampling*.

Sendo a interpretabilidade um ponto chave para o desenvolvimento de CSMs, é necessário destacar que os classificadores também se encontram em ordem preferencial. Isto é, do mais para o menos interpretável. Sobre esta avaliação de interpretabilidade, para referência, a Figura 5.6 divide os classificadores quanto sua performance (levando-se em conta o modelo com melhor *rank* final médio entre os três *datasets* empregados), complexidade e interpretabilidade (conforme entendimento proveniente da RSL).

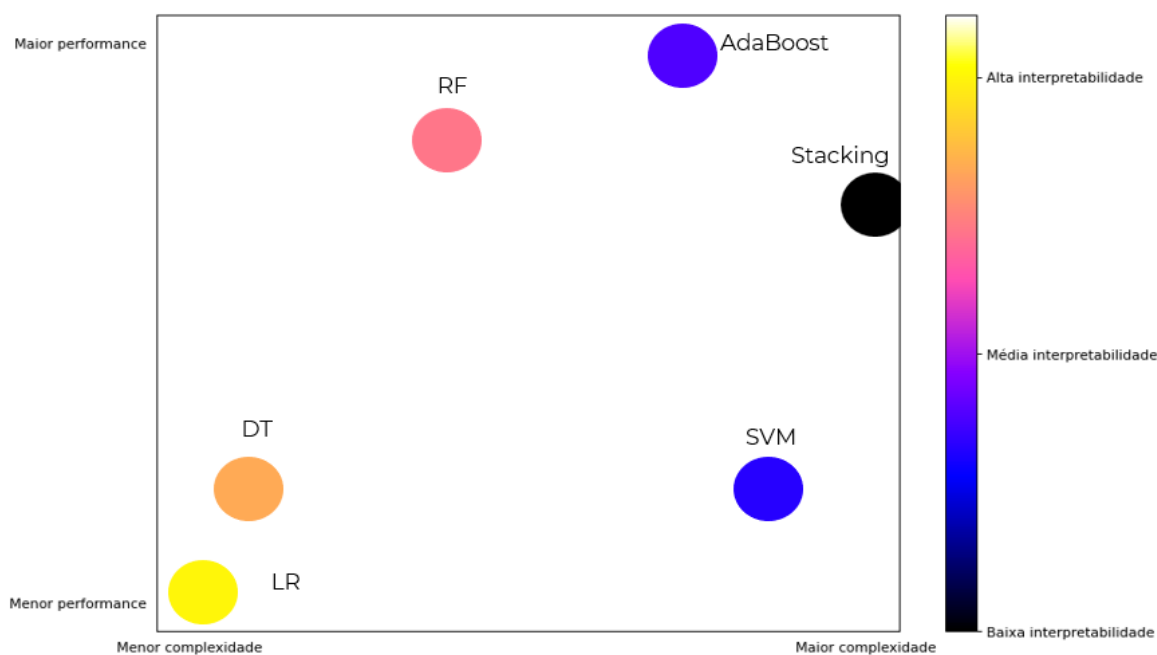


Figura 5.6 – Divisão dos classificadores conforme complexidade, interpretabilidade e performance nos CSMs
Fonte: Autoria própria (2023)

LR, frequentemente tido como a primeira escolha de algoritmo de classificação para CSMs, além de simples e interpretável, é há décadas a referência na indústria. Desta forma, seguindo o *framework* proposto na Figura 5.5, LR serve ao mínimo como *benchmark* para os subsequentes modelos de ML.

Analisando a Figura 4.5, em consonância com a Figura 5.4 (a), com base nas evidências dos modelos elaborados nesta pesquisa, nota-se que, no caso da modelagem de CSMs a partir de LR, as etapas de pré-processamento de dados e *resampling* não só são as que mais causam variações nos modelos, como são os principais fatores para o aumento em sua performance.

Quanto aos modelos individuais de ML, como sugere a Figura 5.6, entre os mais interpretáveis estão as DTs. Pela própria forma com que são estruturadas, é relativamente simples visualizar e interpretar as regras que compõe uma DT, o que faz destes algoritmos ótimas opções para sistemas de gestão de risco de crédito. Como para LR, o pré-processamento e balanceamento de dados demonstraram grande importância na modelagem de DTs, que desta vez, segundo a Figura 5.4 (b), sofrem maior impacto com as etapas de FS e OHP, apesar do efeito das etapas de FS ser inconsistente, não necessariamente contribuindo positivamente para o desempenho dos modelos, como pode ser observado nos melhores modelos DT, na Tabela 4.7.

Finalmente, o algoritmo RF também se encontra incluído no primeiro *framework*. Levando em conta os modelos do Capítulo 4, a performance dos modelos construídos partindo

deste classificador por si só já embasam a sua consideração. Apesar de ter sua interpretabilidade prejudicada em razão da agregação dos *base learners* (tratando-se de *ensemble*), por ser um algoritmo muito versátil e popular, seja na academia ou indústria (contando com incontáveis referências para aplicações distintas), acredita-se que o seu potencial desempenho em CSMs deve ao menos ser avaliado frente a maior dificuldade de explicação do modelo.

No caso de RF, particularmente, conforme a Figura 5.4 (d), a etapa de FS foi, desta vez, a maior responsável pela variação de performance entre modelos. Esta variação, contudo, como nas DTs, nem sempre é positiva (basta observar os melhores modelos RF na Tabela 4.13). Portanto, objetivando praticidade, robustez e garantia de desempenho, esta etapa foi descartada do *framework* na Figura 5.5, o que não significa, contudo, que não possa ser considerada, podendo ser abordada pelo modelador, especialmente quando houver excesso de atributos e restrições de tempo e recursos no processo de modelagem. Afinal, o número ótimo de atributos vai sempre depender do problema, fazendo necessária maior experimentação com FS, e adequando mais esta etapa ao escopo do segundo *framework* (Figura 5.7).

Além disso, conforme fontes alternativas de dados se tornam realidade em aplicações de gerenciamento de riscos, Lessmann *et al.* (2015) argumentam que *Big Data* significa uma quebra de paradigmas na área, e que metodologias mais avançadas de análise de dados devem gradualmente se tornarem mais aceitas. Neste sentido, sendo RF não só um *ensemble* homogêneo, mas um algoritmo muito comum de agregação de DTs (que, por sua vez, são interpretáveis), pode-se afirmar que, ao menos para fins de CS, se trata do mais forte candidato entre os *ensembles*.

Adicionalmente, para os modelos de DT e RF, como aparenta ser o caso dos algoritmos de ML em geral, a etapa de OHP carece de maior atenção, ao contrário de LR. Vale apontar que, também no interesse de uma modelagem parcimoniosa, é importante considerar outros algoritmos de OHP. Uma boa alternativa para a *grid search* é o GA que, conforme Amaral (2021), é uma forma mais eficiente de condução desta etapa, não dependendo de uma busca exaustiva dos hiper parâmetros.

No último passo que precede a tomada de decisão, o *framework* da Figura 5.5 possibilita a comparação entre os modelos, para que sejam avaliados em conformidade com os objetivos do modelador, que deve avaliar diligentemente o *trade-off* dos modelos de ML, na hipótese de sua implementação ser justificada por significativo aumento de performance. Por fim, resta interpretar o modelo final, preferencialmente pelos métodos conhecidos como *model agnostic* (e.g. SHAP *values*, BEEF, LORE e LIME), ou seja, que independem do modelo, facilitando o entendimento das saídas do CSM.

Diferentemente do *framework* da Figura 5.5, que almeja guiar a modelagem de CSMs de forma direta e eficiente, priorizando não só a interpretabilidade como a simplicidade, e descartando algumas das etapas discutidas em favor daquelas com o maior impacto positivo nos modelos, o segundo *framework* (Figura 5.7) inclui também a classificação por meio do SVM, bem como as duas formas restantes de agregação para modelos *ensemble* (*boosting* e *stacking*). Em primeiro lugar, para todos os casos, são incluídas as etapas de pré-processamento e balanceamento de dados, que até aqui mostraram-se essenciais. Além disso, também foi incluída agora a etapa de FS.

Apesar de, considerando os resultados dos modelos, as metodologias de FS terem efeitos inconsistentes sobre a performance dos modelos (o que, como discutido, pode variar significativamente com o número de atributos selecionados), supondo inexistir empecilho para tal, é interessante que se experimente não só com os diferentes tipos de algoritmos de FS (especialmente *filters* e *wrappers*), mas também com o número de atributos selecionados por eles. Neste caso, uma combinação ótima entre os atributos pode, se não incrementar a performance do modelo, colaborar para torná-lo mais eficiente por meio de uma redução de dimensionalidade que não prejudique sua performance. Vale reforçar aqui as conclusões de Zhang *et al.* (2022), que mostram empiricamente que o desempenho do modelo não necessariamente aumenta com o emprego de mais atributos.

Para os *boosting ensembles*, podem ser agregadas as DTs e, no *ensemble* homogêneo por meio de *stacking*, quaisquer dos demais modelos. Por fim, deve ser implementada a OHP, assegurando que sejam experimentados diferentes parâmetros para os modelos e proporcionando, às custas de mais tempo e intensidade computacional, uma modelagem mais completa, que não descarta possíveis melhorias em decorrência desta etapa, por mais que nem sempre o incremento em performance seja significativo. Além da avaliação e interpretação dos modelos de ML, estes podem ser comparados com LR, tido como *benchmark*, e cujo resultado serve de base para validar ou não os potenciais CSMs, avaliando-se apropriadamente o *trade-off* entre o ganho de desempenho e a maior dificuldade de interpretação.

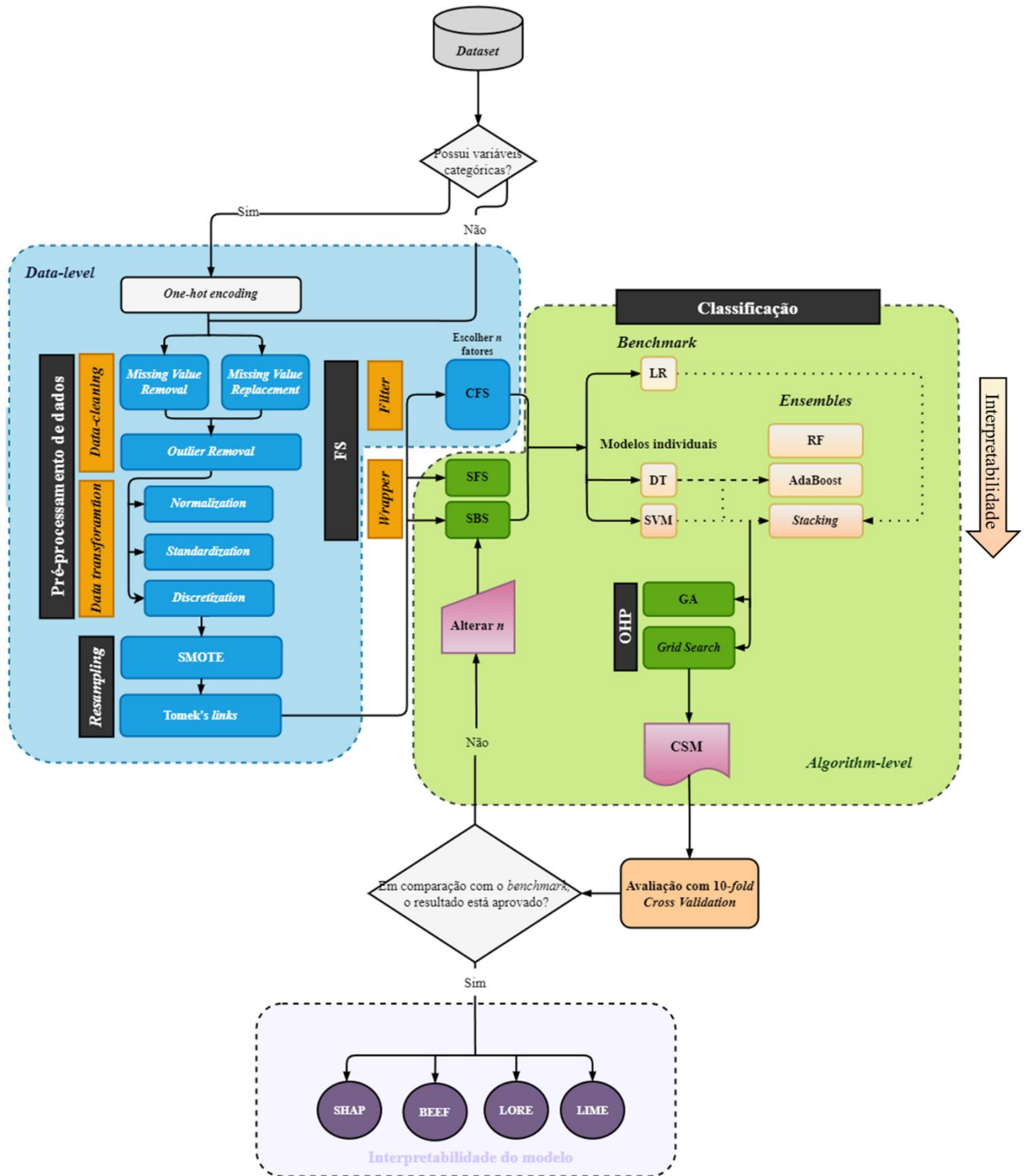


Figura 5.7 – Framework geral e irrestrito para modelagem de CSMs
 Fonte: Autoria própria (2023)

6. CONCLUSÕES E PROPOSTAS PARA ESTUDOS FUTUROS

Nas últimas décadas, principalmente em resposta à crise econômica de 2008, a intensificação dos padrões regulatórios da indústria financeira, aliada ao avanço das tecnologias de IA, tem pressionado as instituições financeiras a adotarem novas metodologias para seu gerenciamento de riscos. Mais particularmente, tratando de CS e seu papel na análise do risco de crédito, na qual se evidencia a modelagem de um problema de classificação, algoritmos de ML tem capturado cada vez mais o interesse dos profissionais e acadêmicos na área. Além de demonstrar grande potencial para superar os resultados dos métodos tradicionais de construção de CSMs, modelos de ML são capazes de processar e analisar um volume maior de estruturas de dados mais complexas.

Contudo, há de se destacar que este potencial de superioridade de performance tem um custo, existindo um *trade-off* com o nível de complexidade da modelagem e a própria interpretabilidade do modelo, que é de grande importância quando se trata de CSMs. Este ponto parece ser ainda o que mais restringe a difusão ampla destes modelos numa indústria em que, como observado neste estudo, vigoram ainda modelos estatísticos, em especial a LR. Na academia, por sua vez, o interesse por algoritmos de ML neste contexto é mais pronunciado, sendo encontrados diversos tipos de algoritmo de ML em níveis variados de complexidade e desempenho.

Certamente, em conformidade com a expectativa gerada pela RSL, os modelos desenvolvidos nesta pesquisa ilustram como ML, de fato, é uma alternativa adequada para lidar com problemas de classificação. Algoritmos de classificação de ML, que se concentram na área do aprendizado supervisionado, têm potencial para alta performance na classificação de grandes quantidades e diferentes tipos de dados, especialmente quando modelados e implementados apropriadamente. Oportunamente, o vasto campo abrangido por *data science* e ML também envolve soluções analíticas para problemas além da classificação. Na verdade, a maior parte do processo de modelagem pode ser beneficiada por técnicas de ML, seja na etapa de FS (*e.g.* RF utilizado para avaliar a importância de cada etapa de modelagem na Figura 5.4), OHP (*e.g.* GA) ou *resampling*.

No que diz respeito a tarefa de classificação em si, assim como a literatura, a interpretação dos resultados dos modelos (Capítulos 4 e 5) sugere que inexistente uma única técnica invariavelmente superior, apesar de algoritmos de ML usualmente triunfem sobre os modelos estatísticos tradicionais. Aumentos em performance geralmente acompanham o aumento da complexidade dos modelos (o que pode ser observado na melhor performance dos métodos

ensemble). Ainda assim, resultados são inconsistentes entre classificadores, e o ganho em performance deve ser avaliado conforme o mencionado *trade-off* com complexidade e interpretabilidade. Portanto, na verdade, em termos de desempenho, o que mais importa não é a escolha ou complexidade do classificador escolhido, e sim o restante da modelagem do CSM como um todo.

Considerando apenas os algoritmos de classificação escolhidos, não acompanhados das demais etapas de modelagem avaliadas nesta pesquisa, os resultados dos modelos conduzidos, além de fundamentarem a inexistência da superioridade de um único classificador, ilustram bem este ponto. Enquanto o modelo RF, por exemplo, obteve o melhor desempenho entre os classificadores em um dos *datasets*, LR teve o melhor em outro, enquanto o SVM (algoritmo de relativa complexidade) mostrou pouquíssimo poder de discriminação em todos. Isto leva a conclusão de que, no contexto de CS, a proposição de modelos de alta complexidade na busca por alta performance, apesar de tentadora, é improdutivo. Na prática, algoritmos de classificação simples podem trazer ótimos resultados, desde que o CSM seja modelado adequadamente a partir deles.

Sabendo que CSMs devem ser utilizados também por profissionais que desconhecem as especificidades técnicas do modelo, parece mais vantajoso se ater a modelos de menor complexidade, porém robustos e bem trabalhados, do que empregar mais esforço e recursos em CSMs excessivamente complexos e que, sem garantia de significativo aumento de performance, tem sua implementação, uso e interpretabilidade prejudicados. Isto não significa dizer que não existe lugar para classificadores de ML em CS, mas que etapas como pré-processamento de dados, balanceamento de classes, FS e OHP causam maior impacto do que a escolha do algoritmo de classificação. Por fim, neste sentido, o papel de ML se torna cada vez maior em *data science* como um todo, e o seu uso para a classificação do risco de crédito deve ser bem avaliado, analisando-se o custo-benefício, principalmente em relação a interpretabilidade do modelo.

Consequentemente, conhecida a importância e o potencial para melhorias da modelagem adequada de CSMs partindo de algoritmos de classificação, parece problemático o fato de que o aumento do interesse acadêmico em ML não é devidamente acompanhado de etapas como o pré-processamento de dados, *resampling*, FS, OHP e interpretabilidade, lembrando que boa parte das publicações não descreve apropriadamente o seu emprego nos modelos desenvolvidos. A inconsistência e falta de padrão, seja na modelagem ou na avaliação e interpretação de modelos, não só compromete os resultados e a implementação dos CSMs, como prejudica a difusão do conhecimento acerca do assunto. Isto reforça a importância de

diretrizes e *frameworks* para auxiliar pesquisadores e profissionais na otimização do processo de modelagem da classificação do risco de crédito ao consumidor, colaborando para o progresso da indústria financeira e para a reprodutibilidade e aperfeiçoamento dos CSMs desenvolvidos e discutidos na academia.

As etapas de pré-processamento de dados, *resampling*, FS e OHP todas demonstraram capacidade de aperfeiçoamento dos modelos, frequentemente impactando de maneira significativa conforme incrementados aos modelos. Não restam dúvidas de que encontra-se devidamente evidenciado o efeito positivo demonstrado pelas etapas consideradas, sendo necessário que não só os profissionais, mas as publicações acerca do tema as considerem de acordo. A escolha do classificador, por outro lado, especialmente tratando de ML, encontra-se restrita até que ferramentas de XAI sejam melhor estudadas e difundidas. Em resposta, o *framework* da Figura 5.1 propõe uma alternativa prática e direta para a modelagem do risco de crédito ao consumidor por meio dos mais populares e efetivos (conforme a RSL) modelos estatísticos e de ML interpretável, prezando pela eficiência e simplicidade de modelagem e implementação.

Nesta linha, tendências emergentes na área de classificação de risco de crédito ao consumidor, e que podem fundamentar pesquisas e trabalhos futuros, incluem modelos *cost-sensitive*, que contabilizam os diferentes custos da classificação incorreta, desenvolvendo algoritmos de classificação voltados para CS e que busca lidar apropriadamente com o Erro Tipo II; além de metodologias de *data augmentation*, em que *datasets* são ligeiramente modificados e o modelo é treinado em suas diferentes versões; e *threshold optimization*, para identificar os limites ótimos de decisão, encontrando o melhor valor para distinguir as classes negativa e positiva.

Por outro lado, desconsiderando restrições rigorosas em termos de tempo, recurso e interpretabilidade (e conseqüentemente mais apropriado para aplicações acadêmicas e não comerciais), o segundo *framework* proposto (Figura 5.2) engloba as principais etapas pertinentes para a modelagem de CSMs por meio dos algoritmos de classificação abordados nesta pesquisa.

Os *frameworks* propostos podem ser empregados por pesquisadores, assim como profissionais e por ECAIs na modelagem para a classificação do risco de crédito ao consumidor. Apesar de não esgotar as possibilidades de etapas de modelagem, bem como as inúmeras metodologias para cada uma delas, tanto os modelos resultantes dos *frameworks* como o *setup* experimental adotado podem servir de referência para futuras pesquisas e desenvolvimento de

modelos, especialmente conforme evoluem os requisitos regulatórios, assim como surgem novos métodos e tecnologias aplicáveis neste contexto.

No caso de abordagens acadêmicas de CS ou, em geral, da modelagem de CSMs desprovida de restrições de eficiência e interpretabilidade, sugestões para pesquisas e trabalhos futuros incluem novas tendências como modelos híbridos e *deep learning* (DASTILE; XOLANE; CELIK, 2020). Novos estudos comparativos podem beneficiar a discussão com *setups* experimentais similares, porém empregando novas etapas de modelagem ou diferentes metodologias para aquelas discutidas nesta pesquisa, assim como outros algoritmos de classificação, avaliando também, por etapa, quais metodologias são melhor pareadas com quais classificadores. Quanto a estudos comparativos, sugere-se também a aplicação práticas dos *frameworks*, seja para incrementá-los, ou para avaliar a efetividade das propostas para problemas reais de CS.

Por fim, a literatura de previsão por meio de ML, em geral, não limitando-se ao desenvolvimento de CSMs, deve progredir no campo de XAI, demonstrando enorme potencial ao viabilizar a transparência e interpretabilidade de modelos mais avançados, robustos e assertivos, o que certamente resultará em impacto positivo em incontáveis áreas, especialmente na gestão de riscos financeiros.

REFERÊNCIAS BIBLIOGRÁFICAS

- ABDOU, H. A.; POINTON, J. **Credit scoring, statistical techniques and evaluation criteria: a review of the literature**. *Intelligent systems in accounting, finance and management*, v. 18, n. 2-3, p. 59-88, 2011.
- ABDOU, H. A.; TSAFACK, M. D. D.; NTIM, C. G.; BAKER, R. D. **Predicting creditworthiness in retail banking with limited scoring data**. *Knowledge-Based Systems*, v. 103, p. 89-103, 2016.
- ABEDINI, M.; AHMADZADEH, F.; NOOROSSANA, R. **Customer credit scoring using a hybrid data mining approach**. *Kybernetes*, 2016.
- AGHAEIRAD, A.; CHEN, N.; RIBEIRO, B. **Improve credit scoring using transfer of learned knowledge from self-organizing map**. *Neural Computing and Applications*, v. 28, n. 6, p. 1329-1342, 2017.
- ALABAS, C.; ALTIPARMAK, F.; DENGIZ, B. **A comparison of the performance of artificial intelligence techniques for optimizing the number of kanbans**. *Journal of the Operational Research Society*, v. 53, n. 8, p. 907–914, 2002.
- ALA'RAJ, M.; ABBOD, M. F. **Classifiers consensus system approach for credit scoring**. *Knowledge-Based Systems*, v. 104, p. 89-105, 2016.
- ALEKSANDROVA, Y.; **Comparing performance of machine learning algorithms for default risk prediction in peer-to-peer lending**. *TEM Journal*, v. 10, n. 1, p. 133-143, 2021.
- ALPAYDIN, E. **Introduction to machine learning**. MIT press, 2020.
- AMPOUNTOLAS, A.; NYARKO NDE, T.; DATE, P.; CONSTANTINESCU, C. **A machine learning approach for micro-credit scoring**. *Risks*, v. 9, n. 3, p. 50, 2021.
- ANTONAKIS, A. C.; SFAKIANAKIS, M. E. **Assessing naive Bayes as a method for screening credit applicants**. *Journal of applied Statistics*, v. 36, n. 5, p. 537-545, 2009.
- APOSTOLIK, R.; DONOHUE, C.; WENT, P. **Foundations of banking risk: an overview of banking, banking risks, and risk-based banking regulation**. John Wiley, 2009.
- ARMAKI, A. G.; FALLAH, M. F.; ALBORZI, M.; MOHAMMADZADEH, A. **A hybrid meta-learner technique for credit scoring of banks' customers**. *Engineering, Technology & Applied Science Research*, v. 7, n. 5, p. 2073-2082, 2017.
- ASLAM, U.; TARIQ AZIZ, H. I.; SOHAIL, A.; BATCHA, N. K. **An empirical study on loan default prediction models**. *Journal of Computational and Theoretical Nanoscience*, v. 16, n. 8, p. 3483-3488, 2019.
- BACH, M.; WERNER, A.; PALT, M. **The proposal of undersampling method for learning from imbalanced datasets**. *Procedia Computer Science*, v. 159, p. 125-134, 2019.
- BAESENS, B.; ROESCH, D.; SCHEULE, H. **Credit risk analytics: Measurement techniques, applications, and examples in SAS**. John Wiley & Sons, 2016.

BAESENS, B.; VAN GESTEL, T.; VAIENE, S.; STEPANOVA, M.; SUYKENS, J.; VANTHIENEN, J. **Benchmarking state-of-the-art classification algorithms for credit scoring**. Journal of the operational research society, v. 54, n. 6, p. 627-635, 2003.

BAI, M.; ZHENG, Y.; SHEN, Y. **Gradient boosting survival tree with applications in credit scoring**. Journal of the Operational Research Society, v. 73, n. 1, p. 39-55, 2022.

BANSAL, M.; GOYAL, A.; CHOUDHARY, A. **A comparative analysis of K-Nearest Neighbour, Genetic, Support Vector Machine, Decision Tree, and Long Short-Term Memory algorithms in machine learning**. Decision Analytics Journal, p. 100071, 2022.

BAO, W.; LIANJU, N.; YUE, K. **Integration of unsupervised and supervised machine learning algorithms for credit risk assessment**. Expert Systems with Applications, v. 128, p. 301-315, 2019.

BARROS, R. C.; BASGALUPP, M. P.; DE CARVALHO, A.C.; FREITAS, A.A. **A survey of evolutionary algorithms for decision-tree induction**. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), v. 42, n. 3, p. 291-312, 2011.

BASHIRI, M.; GERANMAYEH, A. F. **Tuning the parameters of an artificial neural network using central composite design and genetic algorithm**. Scientia Iranica, v. 18, n. 6, p. 1600–1608, 2011.

BEQUÉ, A.; LESSMANN, S. **Extreme learning machines for credit scoring: An empirical evaluation**. Expert Systems with Applications, v. 86, p. 42-53, 2017.

BERGMEIR, C.; BENÍTEZ, J. M. **On the use of cross-validation for time series predictor evaluation**. Information Sciences, v. 191, p. 192–213, 2012.

BERKA, P. **Using the LISp-Miner System for Credit Risk Assessment**. Neural Network World, v. 26, n. 5, p. 497-518, 2016.

BERTRAND, W. M.; FRANSOO, J. C. **Operations management research methodologies using quantitative modeling**. International Journal of Operations and Production Management, v. 22, n. 2, p. 241–264, 2002.

BOOTH, A. **Clear and present questions: formulating questions for evidence-based practice**. Library Hi Tech, v. 24, n. 3, p. 355–368, 2006.

BOUGHACI, D.; ALKHAWALDEH, A. K. **A cooperative classification system for credit scoring**. In: Smart Technologies and Innovation for a Sustainable Future: Proceedings of the 1st American University in the Emirates International Research Conference—Dubai, UAE 2017. Springer International Publishing, p. 11-20, 2019.

BOU-HAMAD, I. **Bayesian credit ratings: A random forest alternative approach**. Communications in Statistics-Theory and Methods, v. 46, n. 15, p. 7289-7300, 2017.

BOZ, Z.; GUNNEC, D.; BIRBIL, S. I.; ÖZTÜRK, M. K. **Reassessment and monitoring of loan applications with machine learning**. Applied Artificial Intelligence, v. 32, n. 9-10, p. 939-955, 2018.

- BREIMAN, L. **Bagging predictors**. Machine learning, v. 24, n. 2, p. 123-140, 1996.
- BREIMAN, L. **Random forests**. Machine Learning, v. 45, p. 5–32, 2001.
- CAO, Y.; XU, Y.; DU, J. **Multi-variable estimation-based safe screening rule for small sphere and large margin support vector machine**. Knowledge-Based Systems, v. 191, p. 105223, 2020.
- CHEN, H.; XIANG, Y. **The study of credit scoring model based on group lasso**. Procedia computer science, v. 122, p. 677-684, 2017.
- CHEN, T. **Credit Default Risk Prediction of Lenders with Resampling Methods**. In: 2021 3rd International Conference on Machine Learning, Big Data and Business Intelligence (MLBDBI). IEEE. p. 123-127, 2021.
- CHEN, W.; MA, C.; MA, L. **Mining the Customer Credit Using Hybrid Support Vector Machine Technique**. Expert Systems With Applications, 36(4), 7611–7616. 2009.
- CHEN, X.; LI, S.; XU, X.; MENG, F.; CAO, W. **A novel GSCI-based ensemble approach for credit scoring**. IEEE Access, v. 8, p. 222449-222465, 2020.
- CHOI, H.; KOO, J-Y.; PARK, C. **Fused least absolute shrinkage and selection operator for credit scoring**. Journal of Statistical Computation and Simulation, v. 85, n. 11, p. 2135-2147, 2015.
- CHOPRA, A.; BHILARE, P. **Application of ensemble models in credit scoring models**. Business Perspectives and Research, v. 6, n. 2, p. 129-141, 2018.
- COENEN, L.; VERBEKE, W.; GUNS, T. **Machine learning methods for short-term probability of default: A comparison of classification, regression and ranking methods**. Journal of the Operational Research Society, v. 73, n. 1, p. 191-206, 2022.
- CORAZZA, M.; DE MARCH, D.; DI TOLLO, G. **Design of adaptive Elman networks for credit risk assessment**. Quantitative Finance, v. 21, n. 2, p. 323-340, 2021.
- CRISCI, C.; GHATTAS, B.; PERERA, G. **A review of supervised machine learning algorithms and their applications to ecological data**. Ecological Modelling, v. 240, p. 113–122, 2012.
- CROOK, J. **Credit scoring: An overview**. European Journal of Operational Research, 95, 24-37. 1996.
- DAI, B., GU, C., ZHAO, E.; QIN, X. **Statistical model optimized random forest regression model for concrete dam deformation monitoring**. Structural Control and Health Monitoring, 25(6), e2170. 2018.
- DAMANIK, I. S., WINDARTO, A. P., WANTO, A., PONINGSIH; ANDANI, S. R.; SAPUTRA, W. **Decision tree optimization in C4.5 algorithm using genetic algorithm**. Journal of Physics: Conference Series, 1255(1), 012012. 2019.

DASTILE, X., CELIK, T.; POTSANE, M. **Statistical and machine learning models in credit scoring: A systematic literature survey**. Applied Soft Computing, 91, 106263. 2020.

DE BOCK, K. W.; VAN DEN POEL, D. **Reconciling performance and interpretability in customer churn prediction using ensemble learning based on generalized additive models**. Expert Systems with Applications, 39(8), 6816-6826. 2012.

DE CASTRO VIEIRA, J. R., BARBOZA, F., SOBREIRO, V. A.; KIMURA, H. **Machine learning models for credit analysis improvements: Predicting low-income families' default**. Applied Soft Computing, 83, 105640. 2019.

DE JONG, K. A. **Analysis of the behavior of a class of genetic adaptive systems**. Technical Report No. 185, Department of Computer and Communication Sciences, University of Michigan. 1975.

DE LA FUENTE, R.; SMITH, R. **Metamodeling a system dynamics model: A contemporary comparison of methods**. In Winter Simulation Conference (WSC) (pp. 1926-1937). IEEE. 2017.

DE SOUSA JUNIOR, W. T., MONTEVECHI, J. A. B., DE CARVALHO MIRANDA, R.; CAMPOS, A. T. **Discrete simulation-based optimization methods for industrial engineering problems: A systematic literature review**. Computers & Industrial Engineering, 128, 526-540. 2019.

DE SOUSA JUNIOR, W. T., MONTEVECHI, J. A. B., DE CARVALHO MIRANDA, R.; CAMPOS, A. T. **Discrete simulation-based optimization methods for industrial engineering problems: A systematic literature review**. Computers & Industrial Engineering, 128, 526-540, 2019.

DE SOUZA, G. F. M., MELANI, A. H. D. A., MICHALSKI, M. A. D. C.; DA SILVA, R. F. (Eds.). **Reliability analysis and asset management of engineering systems**. Elsevier. 2021.

DENYER, D., TRANFIELD, D.; VAN AKEN, J. E. **Developing design propositions through research synthesis**. Organization Studies, 29(3), 393-413. 2008.

DENYER, D.; TRANFIELD, D. **Producing a systematic review**. In D. A. Buchanan & A. Bryman (Eds.), The Sage handbook of organizational research methods (pp. 671-689). 2009.

DIETTERICH, T. G. **Machine-learning research: Four current directions**. AI Magazine, 18(4), 96-136. 1997.

DJEUNDJE, V. B., CROOK, J., CALABRESE, R.; HAMID, M. **Enhancing credit scoring with alternative data**. Expert Systems with Applications, 163, 113766, 2021.

DO AMARAL, J. V. S., MONTEVECHI, J. A. B., DE CARVALHO MIRANDA, R.; DE SOUSA JUNIOR, W. T. **Metamodel-based simulation optimization: A systematic literature review**. Simulation Modelling Practice and Theory, 114, 102403, 2022.

DUDA, R. O., HART, P. E.; STORK, D. G. Chapter 10: **Unsupervised learning and clustering**. In Pattern Classification, 517-599, 2001.

DUFOUR, J. M.; NEVES, J. **Finite-sample inference and nonstandard asymptotics with Monte Carlo tests and R**. In Handbook of statistics (Vol. 41, pp. 3-31), Elsevier, 2019.

DUMITRESCU, E. I., HUÉ, S; HURLIN, C. **Machine Learning or Econometrics for Credit Scoring: Let's Get the Best of Both Worlds**, 2021.

DUMITRESCU, E., HUÉ, S., HURLIN, C.; TOKPAVI, S. **Machine learning for credit scoring: Improving logistic regression with non-linear decision-tree effects**. European Journal of Operational Research, 297(3), 1178-1192, 2022.

DURAND, D. **Credit-rating formulae**. In Risk Elements in Consumer Instalment Financing (pp. 83-91), NBER, 1941.

EDGAR, T.; MANZ, D. **Research methods for cyber security**, Syngress, 2017.

FENERICH, A., STEINER, M. T. A., NETO, P. J. S., TOCHETTO, E., TSUTSUMI, D., ASSEF, F. M.; DOS SANTOS, B. S. **Use of machine learning techniques in bank credit risk analysis**. Revista Internacional de Metodos Numericos Para Calculo y Diseno En Ingenieria, 36(3), 2020.

FERREIRA, P. H., LOUZADA, F.; DINIZ, C. **Credit scoring modeling with state-dependent sample selection: A comparison study with the usual logistic modeling**. Pesquisa Operacional, 35, 39-56, 2015.

FEURER, M.; HUTTER, F. **Hyperparameter optimization**. Automated machine learning: Methods, systems, challenges, 3-33, 2019.

FINLAY, S. **Multiple classifier architectures and their application to credit risk assessment**. European Journal of Operational Research, 210(2), 368-378, 2011.

FISHER, R. A. **The use of multiple measurements in taxonomic problems**. Annals of eugenics, 7(2), 179-188, 1936.

FITZPATRICK, T.; MUES, C. **An empirical comparison of classification algorithms for mortgage default prediction: evidence from a distressed mortgage market**. European Journal of Operational Research, 249(2), 427-439, 2016.

FIX, E., HODGES, J.L. **Discriminatory analysis, nonparametric discrimination: Consistency properties**. Technical Report 4, USAF School of Aviation Medicine, Randolph Field, Texas, 1951.

FLOREZ-LOPEZ, R.; RAMON-JERONIMO, J. M. **Enhancing accuracy and interpretability of ensemble strategies in credit risk assessment**. A correlated-adjusted decision forest proposal. Expert Systems with Applications, 42(13), 5737-5753, 2015.

FREUND, Y.; SCHAPIRE, R. E. **Experiments with a new boosting algorithm**. In ICML Vol. 96, pp. 148-156, Julho 1996.

GALINDO, J.; TAMAYO, P. **Credit risk assessment using statistical and machine learning: basic methodology and risk modeling applications**. Computational economics, 15, 107-143, 2000.

GARCÍA, V., MARQUÉS, A. I.; SÁNCHEZ, J. S. **Improving risk predictions by preprocessing imbalanced credit data**. In Neural Information Processing: 19th International Conference, ICONIP 2012, Doha, Qatar, November 12-15, 2012, Proceedings, Part II 19, 68-75, 2012.

GAVANKAR, S. S.; SAWARKAR, S. D. **Eager decision tree**. In: 2017 2nd International Conference for Convergence in Technology (I2CT). IEEE, 2017. p. 837-840.

GHORBANI, R.; GHOUSI, R. **Comparing different resampling methods in predicting students' performance using machine learning techniques**. IEEE Access, 8, 67899-67911, 2020.

GIL, A. C. **Como elaborar projetos de pesquisa**. Vol. 4, p. 175. São Paulo: Atlas, 2002.

GIRI, P. K., DE, S. S., DEHURI, S.; CHO, S. B. **Biogeography based optimization for mining rules to assess credit risk**. Intelligent Systems in Accounting, Finance and Management, 28(1), 35-51, 2021.

GOH, R. Y.; LEE, L. S. **Credit scoring: a review on support vector machines and metaheuristic approaches**. Advances in Operations Research, 2019.

GUDIVADA, V. N., IRFAN, M. T., FATHI, E.; RAO, D. L. **Cognitive analytics: Going beyond big data analytics and machine learning**. In Handbook of statistics (Vol. 35, pp. 169-205), Elsevier, 2016.

GUO, Y., CHUNG, F. L., LI, G.; ZHANG, L. **Multi-label bioinformatics data classification with ensemble embedded feature selection**. IEEE Access, 7, 103863-103875, 2019.

GUPTA, G. **A self explanatory review of decision tree classifiers**. In International conference on recent advances and innovations in engineering. ICRAIE-2014. pp. 1-7, IEEE, Maio 2014.

HAMDOUN, N.; RGUIBI, K. **Impact of AI and machine learning on financial industry: application on Moroccan credit risk scoring**. Journal of Advanced Research in Dynamical and Control Systems, 11, 1041-1048, 2019.

HAN, L., HAN, L.; ZHAO, H. **Orthogonal support vector machine for credit scoring**. Engineering Applications of Artificial Intelligence, 26(2), 848-862, 2013.

HAND, D. J. **Classifier technology and the illusion of progress**, 2006.

HAND, D. J. **New uses of statistics in retail banking**. American Journal of Mathematical and Management Sciences, 20(3-4), 305-344, 2000.

HAND, D. J.; HENLEY, W. E. **Statistical classification methods in consumer credit scoring: a review**. Journal of the Royal Statistical Society: Series A (Statistics in Society), 160(3), 523-541, 1997.

HANDHIKA, T., FAHRUROZI, A., ZEN, R. I. M., LESTARI, D. P.; SARI, I. **Modified Average of the Base-Level Models in the Hill-Climbing Bagged Ensemble Selection Algorithm for Credit Scoring**. Procedia Computer Science, 157, 229-237, 2019.

HASSOUN, M. H. **Fundamentals of artificial neural networks**. MIT press, 1995.

HASTIE, T., TIBSHIRANI, R., FRIEDMAN, J. H.; FRIEDMAN, J. H. **The elements of statistical learning: data mining, inference, and prediction** (Vol. 2, pp. 1-758). New York: Springer, 2009.

HOLLAND, J. H. **Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence**. MIT press, 1992.

HONORIO, J.; ORTIZ, L. E. **Learning the structure and parameters of large-population graphical games from behavioral data**. J. Mach. Learn. Res., 16(1), 1157-1210, 2015.

HOSMER, D. W., JOVANOVIC, B.; LEMESHOW, S. **Best subsets logistic regression**. Biometrics, 1265-1270, 1989.

HOU, Z., DAI, Z., LAO, W., WANG, Y.; LU, W. **Application of mixed-integer nonlinear optimization programming based on ensemble surrogate model for dense nonaqueous phase liquid source identification in groundwater**. Environmental Engineering Science, 36(6), 699-709, 2019.

HSIEH, N. C.; HUNG, L. P. **A data-driven ensemble classifier for credit scoring analysis**. Expert systems with Applications, 37(1), 534-545, 2010.

HU, L., GAO, W., ZHAO, K., ZHANG, P.; WANG, F. **Feature selection considering two types of feature relevancy and feature interdependency**. Expert Systems with Applications, 93, 423-434, 2018.

HUANG, J., WANG, H.; WANG, W. **Variable selection in classification model via quadratic programming**. Communications in Statistics-Simulation and Computation, 47(7), 1922-1939, 2018.

HUANG, S. C.; HUANG, M. H. **Using SVMs with embedded recursive feature selections for credit rating forecasting**. Journal of Statistics and Management Systems, 13(1), 165-177, 2010.

HUANG, Z., CHEN, H., HSU, C. J., CHEN, W. H.; WU, S. **Credit rating analysis with support vector machines and neural networks: a market comparative study**. Decision support systems, 37(4), 543-558, 2004.

HÜLLEN, G., ZHAI, J., KIM, S. H., SINHA, A., REALFF, M. J.; BOUKOUVALA, F. **Managing uncertainty in data-driven simulation-based optimization**. Computers & Chemical Engineering, 136, 106519, 2020.

HURLEY, M.; ADEBAYO, J. **Credit scoring in the era of big data**. Yale JL & Tech., 18, 148, 2016.

HURLIN, C.; PÉRIGNON, C. **Machine Learning and Data New Sources for Credit Scoring**. Revue d'economie financiere, 135(3), 21-50, 2019.

JANIKOW, C. Z. **Fuzzy decision trees: issues and methods**. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 28(1), 1-14, 1998.

JANITZA, S., STROBL, C.; BOULESTEIX, A. L. **An AUC-based permutation variable importance measure for random forests**. BMC bioinformatics, 14(1), 1-11, 2013.

JENG, J. T. **Hybrid approach of selecting hyperparameters of support vector machine for regression**. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 36(3), 699-709, 2006.

KENNEDY, K., NAMEE, B. M.; DELANY, S. J. **Using semi-supervised classifiers for credit scoring**. Journal of the Operational Research Society, 64, 513-529, 2013.

KHASHMAN, A. **A neural network model for credit risk evaluation**. International Journal of Neural Systems, 19(04), 285-294, 2009.

KHEMAKHEM, S., BEN SAID, F.; BOUJELBENE, Y. **Credit risk assessment for unbalanced datasets based on data mining, artificial neural network and support vector machines**. Journal of Modelling in Management, 13(4), 932-951, 2018.

KITCHENHAM, B. **Procedures for performing systematic reviews**. Keele, UK, Keele University, 33(2004), 1-26, 2004.

KLODA, L. A., BORUFF, J. T.; CAVALCANTE, A. S. **A comparison of patient, intervention, comparison, outcome (PICO) to a new, alternative clinical question framework for search skills, search results, and self-efficacy: a randomized controlled trial**. Journal of the Medical Library Association: JMLA, 108(2), 185, 2020.

KOKATE, S.; CHETTY, M. S. R. **Credit risk assessment of loan defaulters in commercial banks using voting classifier ensemble learner machine learning model**. International Journal of Safety and Security Engineering, 11(5), 565-572, 2021.

KRUPPA, J., SCHWARZ, A., ARMINGER, G.; ZIEGLER, A. **Consumer credit risk: Individual probability estimates using machine learning**. Expert systems with applications, 40(13), 5125-5131, 2013.

KUMAR, R.; VERMA, R. **Classification algorithms for data mining: A survey**. International Journal of Innovations in Engineering and Technology (IJJET), 1(2), 7-14, 2012.
LEO, M., SHARMA, S.; MADDULETY, K. **Machine learning in banking risk management: A literature review**. Risks, 7(1), 29, 2019.

LESSMANN, S., BAESSENS, B., SEOW, H. V.; THOMAS, L. C. **Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research.** European Journal of Operational Research, 247(1), 124-136, 2015.

LESSMANN, Stefan; STAHLBOCK, Robert; CRONE, Sven F. **Genetic algorithms for support vector machine model selection.** IEEE International Conference on Neural Networks - Conference Proceedings, p. 3063–3069, 2006.

LEVY, Y.; ELLIS, T. J. **A systems approach to conduct an effective literature review in support of information systems research.** Informing Science, 9, 2006.

LI, H., QIU, H., SUN, S., CHANG, J.; TU, W. **Credit scoring by one-class classification driven dynamical ensemble learning.** Journal of the Operational Research Society, 73(1), 181-190, 2022.

LI, J., LIU, H., YANG, Z.; HAN, L. **A credit risk model with small sample data based on G-XGBoost.** Applied Artificial Intelligence, 35(15), 1550-1566, 2021.

LI, Y. F., NG, S. H., XIE, M.; GOH, T. N. **A systematic comparison of metamodeling techniques for simulation optimization in decision support systems.** Applied soft computing, 10(4), 1257-1273, 2010.

LI, Y. S., CHI, H., SHAO, X. Y., QI, M. L.; XU, B. G. **A novel random forest approach for imbalance problem in crime linkage.** Knowledge-Based Systems, 195, 105738, 2020.

LI, Y.; CHEN, W. **Entropy method of constructing a combined model for improving loan default prediction: A case study in China.** Journal of the Operational Research Society, 72(5), 1099-1109, 2021.

LI, Z., ZHANG, J., YAO, X.; KOU, G. **How to identify early defaults in online lending: a cost-sensitive multi-layer learning framework.** Knowledge-Based Systems, 221, 106963, 2021.

LIANG, D., TSAI, C. F., DAI, A. J.; EBERLE, W. **A novel classifier ensemble approach for financial distress prediction.** Knowledge and Information Systems, 54, 437-462, 2018.

LIN, W. Y., HU, Y. H.; TSAI, C. F. **Machine learning in financial crisis prediction: a survey.** IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 42(4), 421-436, 2011.

LIU, X. Y., LIANG, Y., WANG, S., YANG, Z. Y.; YE, H. S. **A hybrid genetic algorithm with wrapper-embedded approaches for feature selection.** IEEE Access, 6, 22863-22874, 2018.

LIU, X., FU, H.; LIN, W. **A modified support vector machine model for credit scoring.** International Journal of Computational Intelligence Systems, 3(6), 797-804, 2010.

MALHOTRA, R.; MALHOTRA, D. K. **Identifying potential loan defaulters in the credit union environment: a comparative analysis of statistical and neural network models.** Journal of Information Technology Case and Application Research, 2(2), 20-48, 2000.

MANCISIDOR, R. A., KAMPPFMEYER, M., AAS, K.; JENSSEN, R. **Deep generative models for reject inference in credit scoring**. Knowledge-Based Systems, 196, 105758, 2020.

MARQUES, A. I., GARCÍA, V.; SÁNCHEZ, J. S. **A literature review on the application of evolutionary computing to credit scoring**. Journal of the Operational Research Society, 64, 1384-1399, 2013.

MARQUÉS, A. I.; GARCÍA, V.; SÁNCHEZ, J. S. **On the suitability of resampling techniques for the class imbalance problem in credit scoring**. Journal of the Operational Research Society, 64(7), 1060-1070, 2013a.

MARTENS, D., VAN GESTEL, T., DE BACKER, M., HAESSEN, R., VANTHIENEN, J.; BAESENS, B. **Credit rating prediction using ant colony optimization**. Journal of the Operational Research Society, 61, 561-573, 2010.

METHLEY, A. M., CAMPBELL, S., CHEW-GRAHAM, C., McNALLY, R.; CHERAGHI-SOHI, S. **PICO, PICOS and SPIDER: a comparison study of specificity and sensitivity in three search tools for qualitative systematic reviews**. BMC health services research, 14(1), 1-10, 2014.

MEYER-BÄSE, A., MEYER-BAESE, A.; SCHMID, V. J. **Pattern Recognition and Signal Analysis in Medical Imaging**. Academic Press, 2004.

MIGUEL, PAC *et al.* **Metodologia de Pesquisa em Engenharia de Produção e Gestão de Operações**. 3^o ed. Rio de Janeiro: Elsevier, 2018.

MISRA, S., LI, H.; HE, J. **Noninvasive fracture characterization based on the classification of sonic wave travel times**. Machine learning for subsurface characterization, 243, 2020.

ITROFF, I. I., BETZ, F., PONDY, L. R.; SAGASTI, F. **On managing science in the systems age: two schemas for the study of science as a whole systems phenomenon**. Interfaces, 4(3), 46-58, 1974.

MOHANTY, N., JOHN, A. L. S., MANMATHA, R.; RATH, T. M. **Shape-based image classification and retrieval**. In: Handbook of Statistics, Vol. 31, pp. 249-267. Elsevier, 2013.

MONGEON, P.; PAUL-HUS, A. **The journal coverage of Web of Science and Scopus: a comparative analysis**. Scientometrics, 106, 213-228, 2016.

MOSCATELLI, M., PARLAPIANO, F., NARIZZANO, S.; VIGGIANO, G. **Corporate default forecasting with machine learning**. Expert Systems with Applications, 161, 113567, 2020.

MOSCATO, V., PICARIELLO, A.; SPERLÍ, G. **A benchmark of machine learning approaches for credit score prediction**. Expert Systems with Applications, 165, 113986, 2021.

MUNKHDALAI, L., MUNKHDALAI, T., NAMSRAI, O. E., LEE, J. Y.; RYU, K. H. **An empirical comparison of machine-learning methods on bank client credit assessments.** Sustainability, 11(3), 699, 2019.

NIKLIS, D., DOUMPOS, M.; ZOPOUNIDIS, C. **Combining market and accounting-based models for credit scoring using a classification scheme based on support vector machines.** Applied Mathematics and Computation, 234, 69-81, 2014.

OLIVEIRA, A. L., BRAGA, P. L., LIMA, R. M.; CORNÉLIO, M. L. **GA-based method for feature selection and parameters optimization for machine learning regression applied to software effort estimation.** Information and Software Technology, 52(11), 1155-1166, 2010.

OLIVEIRA, J. B., LIMA, R. S.; MONTEVECHI, J. A. B. **Perspectives and relationships in Supply Chain Simulation: A systematic literature review.** Simulation Modelling Practice and Theory, 62, 166-191, 2016.

ONEL, M., KIESLICH, C. A., GUZMAN, Y. A., FLOUDAS, C. A.; PISTIKOPOULOS, E. N. **Big data approach to batch process monitoring: Simultaneous fault detection and diagnosis using nonlinear support vector machine-based feature selection.** Computers & Chemical Engineering, 115, 46-63, 2018.

ORLOVA, E. V. **Decision-making techniques for credit resource management using machine learning and optimization.** Information, 11(3), 144, 2020.

ØSTERGÅRD, T., JENSEN, R. L.; MAAGAARD, S. E. **A comparison of six metamodeling techniques applied to building performance simulations.** Applied Energy, 211, 89-103, 2018.

PAASS, G.; KINDERMANN, J. **Bayesian classification trees with overlapping leaves applied to credit-scoring.** In: Pacific-Asia Conference on Knowledge Discovery and Data Mining, pp. 234-245. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998.

PAN, Y., CHEN, S., WU, D.; DOLGUI, A. **CF-NN: a novel decision support model for borrower identification on the peer-to-peer lending platform.** International Journal of Production Research, 59(22), 6963-6974, 2021.

PAPOUSKOVA, M.; HÁJEK, P. **Two-stage consumer credit risk modelling using heterogeneous ensemble learning.** Decision Support Systems, 118, 33-45, 2019.

PLAWIAK, P., ABDAR, M.; ACHARYA, U. R. **Application of new deep genetic cascade ensemble of SVM classifiers to predict the Australian credit scoring.** Applied Soft Computing, 84, 105740, 2019.

POLIKAR, R. **Ensemble based systems in decision making.** IEEE Circuits and Systems Magazine, 6(3), 21-45, 2006.

PUDJIHARTONO, N., FADASON, T., KEMPA-LIEHR, A. W.; O'SULLIVAN, J. M. **A review of feature selection methods for machine learning-based disease risk prediction.** Frontiers in Bioinformatics, 2, 927312, 2022.

RANI, A., KUMAR, N., KUMAR, J.; SINHA, N. K. **Machine learning for soil moisture assessment**. In: Deep Learning for Sustainable Agriculture, pp. 143-168. Academic Press, 2022.

ROY, D. K.; DATTA, B. **Trained meta-models and evolutionary algorithm based multi-objective management of coastal aquifers under parameter uncertainty**. Journal of Hydroinformatics, 20(6), 1247-1267, 2018.

SAIA, R., CARTA, S.; FENU, G. **A wavelet-based data analysis to credit scoring**. In: Proceedings of the 2nd International Conference on Digital Signal Processing, pp. 176-180, 2018.

SANEI, S.; CHAMBERS, J. A. **EEG signal processing**. John Wiley & Sons, 2013.

SANTANA, L. E., SILVA, L., CANUTO, A. M., PINTRO, F.; VALE, K. O. **A comparative analysis of genetic algorithm and ant colony optimization to select attributes for a heterogeneous ensemble of classifiers**. In: IEEE Congress on Evolutionary Computation, pp. 1-8, 2010.

SARIEV, E.; GERMANO, G. **Bayesian regularized artificial neural networks for the estimation of the probability of default**. Quantitative Finance, 20(2), 311-328, 2020.

SAUNDERS, A.; CORNETT, M. M. **Financial Institutions Management: A Risk Management Approach**. McGraw-Hill Irwin, 2008.

SEUFERT, Eric Benjamin. **The freemium business model**. Freemium Economics, p. 1-27, 2014.

SHAJALAL, M., HÁJEK, P.; ABEDIN, M. Z. **Product backorder prediction using deep neural network on imbalanced data**. International Journal of Production Research, 61(1), 302-319, 2023.

SHALEV-SHWARTZ, S.; BEN-DAVID, S. **Understanding machine learning: From theory to algorithms**. Cambridge University Press, 2014.

SHEN, F., WANG, R.; SHEN, Y. **A cost-sensitive logistic regression credit scoring model based on multi-objective optimization approach**. Technological and Economic Development of Economy, 26(2), 405-429, 2020.

SHI, J., ZHANG, S. Y.; QIU, L. M. **Credit scoring by feature-weighted support vector machines**. Journal of Zhejiang University SCIENCE C, 14(3), 197-204, 2013.

SIAMI, M., GHOLAMIAN, M. R.; BASIRI, J. **An application of locally linear model tree algorithm with a combination of feature selection in credit scoring**. International Journal of Systems Science, 45(10), 2213-2222, 2014.

STEIN, G., CHEN, B., WU, A. S.; HUA, K. A. **Decision tree classifier for network intrusion detection with GA-based feature selection**. In: Proceedings of the 43rd Annual Southeast Regional Conference-Volume 2, pp. 136-141, 2005.

STORTI, B. A., DORELLA, J. J., ROMAN, N. D., PERALTA, I.; ALBANESI, A. E. **Improving the efficiency of a Savonius wind turbine by designing a set of deflector plates with a metamodel-based optimization approach.** *Energy*, 186, 115814, 2019.

STROBL, C., BOULESTEIX, A. L., KNEIB, T., AUGUSTIN, T.; ZEILEIS, A. **Conditional variable importance for random forests.** *BMC Bioinformatics*, 9, 1-11, 2008.

SUBASI, A. **Practical machine learning for data analysis using Python.** Academic Press, 2020.

SUN, J., LEE, Y. C., LI, H.; HUANG, Q. H. **Combining B&B-based hybrid feature selection and the imbalance-oriented multiple-classifier ensemble for imbalanced credit risk assessment.** *Technological and Economic Development of Economy*, 21(3), 351-378, 2015.

SWAIN, Philip H.; HAUSKA, Hans. The decision tree classifier: Design and potential. *IEEE Transactions on Geoscience Electronics*, v. 15, n. 3, p. 142-147, 1977.

TAKAWIRA, O.; MWAMBA, J. W. M. **Sovereign Credit Ratings Analysis Using the Logistic Regression Model.** *Risks*, 10(4), 70, 2022.

TELES, G., RODRIGUES, J. J., RABÊLO, R. A.; KOZLOV, S. A. **Comparative study of support vector machines and random forests machine learning algorithms on credit operation.** *Software: Practice and Experience*, 51(12), 2492-2500, 2021.

THOMAS, L. C. **A survey of credit and behavioural scoring: forecasting financial risk of lending to consumers.** *International Journal of Forecasting*, 16(2), 149-172, 2000.

THOMAS, L. C. **Consumer finance: Challenges for operational research.** *Journal of the Operational Research Society*, 61(1), 41-52, 2010.

THOMAS, L., CROOK, J.; EDELMAN, D. **Credit scoring and its applications.** Society for Industrial and Applied Mathematics, 2017.

TOLLES, J.; MEURER, W. J. **Logistic regression: relating patient characteristics to outcomes.** *JAMA*, 316(5), 533-534, 2016.

TOMCZAK, J. M.; ZIĘBA, M. **Classification restricted Boltzmann machine for comprehensible credit scoring model.** *Expert Systems with Applications*, 42(4), 1789-1796, 2015.

TORVEKAR, N.; GAME, P. S. **Predictive analysis of credit score for credit card defaulters.** *Int. J. Recent Technol. Eng*, 7(1), 4, 2019.

TOUNSI, Y., HASSOUNI, L.; ANOUN, H. **An enhanced comparative assessment of ensemble learning for credit scoring.** *International Journal of Machine Learning and Computing*, 8(5), 409-415, 2018.

TRIPATHI, D., EDLA, D. R., KUPPILI, V.; BABLANI, A. **Evolutionary extreme learning machine with novel activation function for credit scoring.** *Engineering Applications of Artificial Intelligence*, 96, 103980, 2020.

TRIVEDI, S. K. **A study on credit scoring modeling with different feature selection and machine learning approaches.** *Technology in Society*, 63, 101413, 2020.

TRUCCO, E., MACGILLIVRAY, T.; XU, Y. **Computational retinal image analysis: tools, applications and perspectives.** Academic Press, 2019.

TSAI, C. F., HSU, Y. F.; YEN, D. C. **A comparative study of classifier ensembles for bankruptcy prediction.** *Applied Soft Computing*, 24, 977-984, 2014.

TURRIONI, J. B.; MELLO, C. H. P. **Metodologia de pesquisa em engenharia de produção: estratégias, métodos e técnicas para condução de pesquisas quantitativas e qualitativas.** Itajubá: Unifei, 2012.

VAN GESTEL, I. T., BAESSENS, B., GARCIA, I. J.; VAN DIJCKE, P. **A support vector machine approach to credit scoring.** In *Forum Financier-revue Bancaire et Financiera Bank en Financiewezen*. pp. 73-82, 2003.

VAN LIEBERGEN, B. **Machine learning: a revolution in risk management and compliance?.** *Journal of Financial Transformation*, 45, 60-67, 2017.

VAPNIK, V. **The nature of statistical learning theory.** Springer Science & Business Media, 1999.

WAAD, B., GHAZI, B. M.; MOHAMED, L. **A three-stage feature selection using quadratic programming for credit scoring.** *Applied Artificial Intelligence*, 27(8), 721-742, 2013.

WAH, Y. B., IBRAHIM, N., HAMID, H. A., ABDUL-RAHMAN, S.; FONG, S. **Feature Selection Methods: Case of Filter and Wrapper Approaches for Maximising Classification Accuracy.** *Pertanika Journal of Science & Technology*, 26(1), 2018.

WANG, G., HAO, J., MA, J.; JIANG, H. **A comparative assessment of ensemble learning for credit scoring.** *Expert Systems with Applications*, 38(1), 223-230, 2011.

WANG, H., XU, Q.; ZHOU, L. **Large unbalanced credit scoring using lasso-logistic regression ensemble.** *PloS One*, 10(2), e0117844, 2015.

WANG, Y., WANG, S.; LAI, K. K. **A new fuzzy support vector machine to evaluate credit risk.** *IEEE Transactions on Fuzzy Systems*, 13(6), 820-831, 2005.

WANG, Y., ZHANG, Y., LU, Y.; YU, X. **A Comparative Assessment of Credit Risk Model Based on Machine Learning—a case study of bank loan data.** *Procedia Computer Science*, 174, 141-149, 2020.

WENG, C. H.; HUANG, C. K. **A Hybrid Machine Learning Model for Credit Approval.** *Applied Artificial Intelligence*, 35(15), 1439-1465, 2021.

WEST, D. **Neural network credit scoring models.** *Computers & Operations Research*, 27(11-12), 1131-1152, 2000.

WILDRIDGE, V.; BELL, L. **How CLIP became ECLIPSE: a mnemonic to assist in searching for health policy/management information.** *Health Information & Libraries Journal*, 19(2), 113-115, 2002.

WOLLMANN, Dewey; TORTATO, Ubiratã. **Proposal for a model to hierarchize strategic decisions according to criteria of value innovation, sustainability and budgetary constraint.** *Journal of Cleaner Production*, v. 231, p. 278–289, 2019.

WU, C. F., HUANG, S. C., CHIOU, C. C.; WANG, Y. M. **A predictive intelligence system of credit scoring based on deep multiple kernel learning.** *Applied Soft Computing*, 111, 107668, 2021.

WU, D.; OLSON, D. L. **Enterprise risk management: coping with model risk in a large bank.** *Journal of the Operational Research Society*, 61(2), 179-190, 2010.

WU, J., CHEN, X. Y., ZHANG, H., XIONG, L. D., LEI, H.; DENG, S. H. **Hyperparameter optimization for machine learning models based on Bayesian optimization.** *Journal of Electronic Science and Technology*, 17(1), 26-40, 2019.

XIAO, Y.; WATSON, M. **Guidance on conducting a systematic literature review.** *Journal of Planning Education and Research*, 39(1), 93-112, 2019.

YANG, S.; BERDINE, G. **The receiver operating characteristic (ROC) curve.** *The Southwest Respiratory and Critical Care Chronicles*, 5(19), 34-36, 2017.

YETURU, K. **Machine learning algorithms, applications, and practices in data science.** In *Handbook of Statistics* (Vol. 43, pp. 81-206), Elsevier, 2020.

YONGJUN, P., MINGHAO, P., KIEJUNG, P.; KEUN, H. Y. **An ensemble correlation-based gene selection algorithm for cancer classification with gene expression data.** *Bioinformatics*, 28(24), 3306-3315, 2012.

YU, L., WANG, S.; LAI, K. K. **Credit risk assessment with a multistage neural network ensemble learning approach.** *Expert Systems with Applications*, 34(2), 1434-1444, 2008.

YU, X., YANG, Q., WANG, R., FANG, R.; DENG, M. **Data cleaning for personal credit scoring by utilizing social media data: An empirical study.** *IEEE Intelligent Systems*, 35(2), 7-15, 2020.

YUSTA, S. C. **Different metaheuristic strategies to solve the feature selection problem.** *Pattern Recognition Letters*, 30(5), 525-534, 2009.

ZHANG, D., ZHOU, X., LEUNG, S. C.; ZHENG, J. **Vertical bagging decision trees model for credit scoring.** *Expert Systems with Applications*, 37(12), 7838-7843, 2010.

ZHANG, T., ZHANG, W., WEI, X. U.; HAIJING, H. A. O. **Multiple instance learning for credit risk assessment with transaction data.** *Knowledge-Based Systems*, 161, 65-77, 2018.

ZHANG, Y., LI, S., WANG, T.; ZHANG, Z. **Divergence-based feature selection for separate classes.** *Neurocomputing*, 101, 32-42, 2013.

ZHANG, Z., CHI, G., COLOMBAGE, S.; ZHOU, Y. **Credit scoring model based on a novel group feature selection method: The case of Chinese small-sized manufacturing enterprises**. Journal of the Operational Research Society, 73(1), 122-138, 2022.

ZHENG, B. **Financial default payment predictions using a hybrid of simulated annealing heuristics and extreme gradient boosting machines**. International Journal of Internet Technology and Secured Transactions, 9(4), 404-425, 2019.

ZHOU, Y., SHAMSU UDDIN, M., HABIB, T., CHI, G.; YUAN, K. **Feature selection in credit risk modeling: an international evidence**. Economic research-Ekonomska istraživanja, 34(1), 3064-3091, 2021.

ZIEMBA, P., BECKER, J., BECKER, A., RADOMSKA-ZALAS, A., PAWLUK, M.; WIERZBA, D. **Credit decision support based on real set of cash loans using integrated machine learning algorithms**. Electronics, 10(17), 2099, 2021.

ZIEMBA, P., RADOMSKA-ZALAS, A.; BECKER, J. **Client evaluation decision models in the credit scoring tasks**. Procedia Computer Science, 176, 3301-3309, 2020.

Continuação da Tabela A.1 – Resultados para todos os modelos de classificação de risco de crédito ao consumidor

Autor	Dataset	Accuracy	AUC	Precision	F-score	TPR (Sensitivity)	Recall	TNR (Specificity)	KSS	PCC	FPR ↓	GC	Type I Error ↓	Type II Error ↓	Brier Score ↓	Training Time (s)	H-Measure	NPV	Negative Recall
Bao, Lianju e Yue (2019)	Chinese P2P Company/UCI German/USI Australian	0,919/0,81 3/0,866	0,958/0,8 59/0,941	0,885/0,68 /0,870	-	-	0,940/0,844 /0,983	-	-	-	-	-	0,012/0,11 9/0,088	0,06/0,156/ 0,017	-	-	-	-	-
Wang, Xu e Zhou (2015)	Kaggle Dataset UCI	-	0,87	-	0,41	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Gholamian e Basiri (2014)	Australian/UCI German	0,886/0,77	-	-	-	-	-	-	-	-	-	-	0,075/0,11 4	0,107/0,39	-	-	-	-	-
Martens <i>et al.</i> (2010)	Unspecified SME Data	0,742/0,87 7	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Li e Chen (2021)	Rural Chinese Bank Dataset UCI	0,911	0,798	-	-	-	-	-	0,484	-	-	-	-	-	0,079	-	-	-	-
Zhou <i>et al.</i> (2021)	Australian/UCI Japanese/UCI German	0,88/0,998/ 0,888	0,954/0,9 99/0,958	-	-	-	-	-	-	-	-	-	0,075/0,00 1/0,075	0,068/0,00 3/0,07	-	-	-	-	-
Choi, Koo e Park (2015)	UCI German	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Malhotra e Malhotra (2000)	Pooled data from 12 American Credit Unions	0,715	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Boz <i>et al.</i> 2018	Koçfinans Personal Car Loan Dataset (Turkish Consumer Financing Company) Innovative Microfinance Limited (IML): a micro-lending institution in Ghana	0,945	0,77	-	-	1	-	0,448	-	-	-	-	-	-	-	-	-	-	-
AMPOUNTOLA S <i>et al.</i> (2021)	UCI Australian	0,81	0,96	0,8	0,74	-	0,69	-	-	-	-	-	-	-	-	-	-	-	-
KHASHMAN (2009)	UCI Australian	-	-	-	-	-	-	-	-	0,8263	-	-	-	-	-	-	-	-	-
Teles <i>et al.</i> (2021)	Not specified	-	0,9834	1	0,97	0,946	-	-	-	-	-	-	-	-	-	-	-	-	-
Mancisidor <i>et al.</i> (2020)	LendingClub/Santander Consumer Bank Datasets	-	0,6363/0,7431	0,1825/0,2303	-	-	0,4957/1	-	-	-	-	0,275 5/0,4851	-	-	-	-	0,0632/0,1764	-	-
Li <i>et al.</i> (2021)	LendingClub Dataset	-	0,688	-	0,422	-	-	-	0,272	-	-	-	-	-	-	5	-	-	-
De Castro Vieira <i>et al.</i> (2019)	Brazil Affordable Housing Contracts ("Minha Casa Minha Vida")	0,9668	0,9043	-	-	0,7623	-	1	0,6925	-	-	-	-	-	0,0303	-	-	-	-

Continuação da Tabela A.1 – Resultados para todos os modelos de classificação de risco de crédito ao consumidor

Autor	Dataset	Accuracy	AUC	Precision	F-score	TPR (Sensitivity)	Recall	TNR (Specificity)	KSS	PCC	FPR ↓	GC	Type I Error ↓	Type II Error ↓	Brier Score ↓	Training Time (s)	H-Measure	NPV	Negative Recall
Papouškova e Hajek (2019)	LendingClub (Kaggle Dataset)/Czech Dataset	0,8858/0,8045	0,9572/0,9055	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Djeundje <i>et al.</i> (2021)	Lenddo Datasets (Mexican Bank and Nigerian Bank data) separados em Ensemble A e Ensemble B	-	0,6402/0,6241	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Moscato, Picariello e Sperli (2021)	LendingClub (Kaggle Dataset)	0,773	0,72	-	-	0,99	-	0,877	-	-	0,32	-	-	-	-	-	-	-	-
Handhika <i>et al.</i> (2019)	UCI German	-	0,759	-	-	-	-	-	-	0,73	-	-	-	-	0,17629	-	-	-	-
Boughaci e Alkhalwaldeh (2017)	UCI German/UCI Australian/UCI Polish/UCI Japanese/"Give Me Some Credit" (Kaggle Dataset)	-	0,959/0,977/0,828/0,982/0,913	0,918/0,941/0,923/0,936/0,954	-	-	0,918/0,941/0,931/0,936/0,958	-	-	-	-	-	-	-	-	1,079/0,901/3,915/0,899/4,03	-	-	-
Liang <i>et al.</i> (2018)	Tawian Bankruptcy Dataset/China Bankruptcy Dataset	0,8797/0,7760	-	-	-	-	-	-	-	-	-	-	0,0961/0,3600	0,0753/0,0086	-	-	-	-	-
Waad, Ghazi e Mohamed (2013)	UCI Australian/UCI German/HMEQ Dataset	-	0,958/0,769/0,888	0,931/0,681/0,848	0,958/0,568/0,621	0,976/0,589/0,494	0,972/0,590/0,546	-	-	-	0,062/0,113/0,010	-	-	-	-	-	-	-	-
Sariev e Germano (2020)	East-European (not publicly available) Dataset/Polish Dataset collected from Emerging Markets Information Service	-	-	-	-	-	-	-	-	0,7	-	0,61	-	-	-	1	-	-	-

Continuação da Tabela A.1 – Resultados para todos os modelos de classificação de risco de crédito ao consumidor

Autor	Dataset	Accuracy	AUC	Precision	F-score	TPR (Sensitivity)	Recall	TNR (Specificity)	KSS	PCC	FPR ↓	GC	Type I Error ↓	Type II Error ↓	Brier Score ↓	Training Time (s)	H-Measure	NPV	Negative Recall
Pan (2021)	CashBUS (Chinese p2p lending platform)	0,88	-	0,95	0,93	-	0,96	-	-	-	-	-	-	-	-	-	-	0,39	0,62
Sun <i>et al.</i> (2015)	UCI German	0,7843	-	-	-	0,7268	-	0,9209	-	-	-	-	-	-	-	-	-	-	-
Li <i>et al.</i> (2022)	Rong360 Financial Algorithm Challenge/Len dingClub Datasets	-	0,8103/0, 8938/0,8 529/0,90 29/0,872 1	-	-	-	-	-	0,4827/0,4638 /0,4486/0,454 7/0,4629	-	0,1222/0 1222/0, 1297/0,1 024/0,11 30	-	-	-	-	27,48/5,01/ 2,43/7,13/6 ,20	-	-	-
Zhang <i>et al.</i> (2022)	Romanian Commercial Bank	0,868/0,75 3/0,679/0,9 33	0,931/0,8 01/0,644/ 0,864	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Bai, Zhen e Shen (2022)	Lending Club (Kaggle)/Chin ese Consumer Market Private Dataset	-	0,7702	-	-	-	-	-	0,4245	-	-	-	-	-	-	-	-	-	-
Marqués, García e Sánchez (2013a)	UCI Australian/UC I German/UCI Japanese/UCS D/Iranian	-	0,891/0,7 53/0,888/ 0,826/0,7 32	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Kennedy, Namee e Delany (2013)	UCI Australian/UC I German/Irania n Dataset/UCI Japanese/PAK DD Competition Dataset/Polan d Dataset/Spani sh bank Dataset/Thom as <i>et al.</i> (2002) Dataset/UCS D Dataset	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0,884/0,703/0,809/ 0,672/0,595/0,700/ 0,647/0,595/0,647	-	-

Continuação da Tabela A.1 – Resultados para todos os modelos de classificação de risco de crédito ao consumidor

Autor	Dataset	Accuracy	AUC	Precision	F-score	TPR (Sensitivity)	Recall	TNR (Specificity)	KSS	PCC	FPR ↓	GC	Type I Error ↓	Type II Error ↓	Brier Score ↓	Training Time (s)	H-Measure	NPV	Negative Recall
Weng, Huang e Cheng-Kui (2021)	UCI German/UCI Australian/UCI I ML Repository Credit- Approval Dataset	0,89/0,94/0 ,94	-	0,93/0,92/ 0,95	0,92/0,94/0,93	0,96/0,96/0,94	-	-	-	-	-	-	-	-	-	-	-	-	-
	Liu, Fu e Lin (2010)	UCI German/UCI Australian UCI	0,755/0,86 8	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Armaki <i>et al.</i> (2017)	UCI Australian/UCI I German/UCI Japanese/Mell at Bank Iranian Consumer Loan Dataset	0,999/0,99 8/0,999/0,9 99	-	-	0,998/0,997/0,999/ 0,992	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Aleksandrova (2021)	Lending Club Dataset	0,766	0,701	-	-	0,897	-	0,903	-	-	-	-	-	-	-	-	-	-	-
Ziemba <i>et al.</i> (2021)	Not specified	-	-	-	-	-	-	-	-	-	-	0,828	-	-	-	-	-	-	-
Marqués, García e Sánchez (2013a)	RRDai Dataset	0,984	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Fonte: Artigos consultados na RSL

ANEXO B – MELHORES PARÂMETROS PARA OS MODELOS LR COM A ETAPA DE OHP

Tabela B.1 – Parâmetros ótimos encontrados na *grid search* para os modelos LR

<i>Australian</i>	<i>German</i>	<i>Japanese</i>
LR_OHP: {'C': 0.5, 'class_weight': 'balanced', 'max_iter': 100, 'penalty': 'l2', 'solver': 'newton-cg'}	LR_OHP: {'C': 0.1, 'class_weight': 'balanced', 'max_iter': 200, 'penalty': 'l2', 'solver': 'lbfgs'}	LR_OHP: {'C': 0.5, 'class_weight': 'balanced', 'max_iter': 1000, 'penalty': 'l2', 'solver': 'lbfgs'}
LR_WFS_OHP: {'C': 0.5, 'class_weight': 'balanced', 'max_iter': 100, 'penalty': 'l2', 'solver': 'lbfgs'}	LR_WFS_OHP: {'C': 1, 'class_weight': 'balanced', 'max_iter': 100, 'penalty': 'l2', 'solver': 'lbfgs'}	LR_WFS_OHP: {'C': 10, 'class_weight': 'balanced', 'max_iter': 100, 'penalty': 'l2', 'solver': 'lbfgs'}
LR_PP_OHP: {'C': 10, 'class_weight': 'balanced', 'max_iter': 100, 'penalty': 'l2', 'solver': 'lbfgs'}	LR_PP_OHP: {'C': 0.1, 'class_weight': 'balanced', 'max_iter': 100, 'penalty': 'l2', 'solver': 'lbfgs'}	LR_PP_OHP: {'C': 0.5, 'class_weight': 'balanced', 'max_iter': 100, 'penalty': 'l2', 'solver': 'lbfgs'}
LR_PP_WFS_OHP: {'C': 0.0001, 'class_weight': 'balanced', 'max_iter': 100, 'penalty': 'l2', 'solver': 'lbfgs'}	LR_PP_WFS_OHP: {'C': 20, 'class_weight': 'balanced', 'max_iter': 100, 'penalty': 'l2', 'solver': 'lbfgs'}	LR_PP_WFS_OHP: {'C': 10, 'class_weight': 'balanced', 'max_iter': 100, 'penalty': 'l2', 'solver': 'lbfgs'}
LR_FFS_OHP: {'C': 1, 'class_weight': 'balanced', 'max_iter': 100, 'penalty': 'l2', 'solver': 'lbfgs'} LR_RES_OHP: {'C': 1, 'class_weight': 'balanced', 'max_iter': 100, 'penalty': 'l2', 'solver': 'newton-cg'}	LR_FFS_OHP: {'C': 0.1, 'class_weight': 'balanced', 'max_iter': 100, 'penalty': 'l2', 'solver': 'lbfgs'} LR_RES_OHP: {'C': 0.5, 'class_weight': 'balanced', 'max_iter': 500, 'penalty': 'l2', 'solver': 'lbfgs'}	LR_FFS_OHP: {'C': 0.0001, 'class_weight': 'balanced', 'max_iter': 100, 'penalty': 'l2', 'solver': 'lbfgs'} LR_RES_OHP: {'C': 50, 'class_weight': 'balanced', 'max_iter': 100, 'penalty': 'l2', 'solver': 'lbfgs'}
LR_RES_WFS_OHP: {'C': 20, 'class_weight': 'balanced', 'max_iter': 100, 'penalty': 'l2', 'solver': 'lbfgs'}	LR_RES_WFS_OHP: {'C': 1, 'class_weight': 'balanced', 'max_iter': 100, 'penalty': 'l2', 'solver': 'lbfgs'}	LR_RES_WFS_OHP: {'C': 5, 'class_weight': 'balanced', 'max_iter': 100, 'penalty': 'l2', 'solver': 'lbfgs'}
LR_PP_FFS_OHP: {'C': 0.5, 'class_weight': 'balanced', 'max_iter': 100, 'penalty': 'l2', 'solver': 'lbfgs'}	LR_PP_FFS_OHP: {'C': 0.01, 'class_weight': 'balanced', 'max_iter': 100, 'penalty': 'l2', 'solver': 'lbfgs'}	LR_PP_FFS_OHP: {'C': 0.5, 'class_weight': 'balanced', 'max_iter': 100, 'penalty': 'l2', 'solver': 'lbfgs'}
LR_PP_RES_OHP: {'C': 0.01, 'class_weight': 'balanced', 'max_iter': 100, 'penalty': 'l2', 'solver': 'lbfgs'}	LR_PP_RES_OHP: {'C': 0.1, 'class_weight': 'balanced', 'max_iter': 100, 'penalty': 'l2', 'solver': 'lbfgs'}	LR_PP_RES_OHP: {'C': 10, 'class_weight': 'balanced', 'max_iter': 100, 'penalty': 'l2', 'solver': 'lbfgs'}
LR_PP_RES_WFS_OHP: {'C': 1, 'class_weight': 'balanced', 'max_iter': 100, 'penalty': 'l2', 'solver': 'lbfgs'}	LR_PP_RES_WFS_OHP: {'C': 0.5, 'class_weight': 'balanced', 'max_iter': 100, 'penalty': 'l2', 'solver': 'lbfgs'}	LR_PP_RES_WFS_OHP: {'C': 0.5, 'class_weight': 'balanced', 'max_iter': 100, 'penalty': 'l2', 'solver': 'lbfgs'}
LR_RES_FFS_OHP: {'C': 1, 'class_weight': 'balanced', 'max_iter': 100, 'penalty': 'l2', 'solver': 'lbfgs'}	LR_RES_FFS_OHP: {'C': 0.1, 'class_weight': 'balanced', 'max_iter': 100, 'penalty': 'l2', 'solver': 'lbfgs'}	LR_RES_FFS_OHP: {'C': 5, 'class_weight': 'balanced', 'max_iter': 100, 'penalty': 'l2', 'solver': 'lbfgs'}
LR_WFS_OHP: {'C': 0.01, 'class_weight': 'balanced', 'max_iter': 100, 'penalty': 'l2', 'solver': 'lbfgs'}	LR_WFS_OHP: {'C': 0.01, 'class_weight': 'balanced', 'max_iter': 100, 'penalty': 'l2', 'solver': 'lbfgs'}	LR_WFS_OHP: {'C': 1, 'class_weight': 'balanced', 'max_iter': 100, 'penalty': 'l2', 'solver': 'lbfgs'}

Fonte: Autoria própria (2023)

ANEXO C – PARÂMETROS DOS MODELOS DT COM OHP

Tabela C.1 – Parâmetros ótimos encontrados na *grid search* para os modelos DT

<i>Australian</i>	<i>German</i>	<i>Japanese</i>
DT_OHP: {'criterion': 'entropy', 'max_depth': 5, 'max_features': None, 'min_samples_leaf': 8, 'min_samples_split': 2}	DT_OHP: {'criterion': 'entropy', 'max_depth': 10, 'max_features': None, 'min_samples_leaf': 8, 'min_samples_split': 2}	DT_OHP: {'criterion': 'gini', 'max_depth': 10, 'max_features': 'log2', 'min_samples_leaf': 4, 'min_samples_split': 15}
DT_WFS_OHP: {'criterion': 'gini', 'max_depth': 5, 'max_features': 'sqrt', 'min_samples_leaf': 2, 'min_samples_split': 10}	DT_WFS_OHP: {'criterion': 'gini', 'max_depth': None, 'max_features': None, 'min_samples_leaf': 8, 'min_samples_split': 2}	DT_WFS_OHP: {'criterion': 'gini', 'max_depth': None, 'max_features': None, 'min_samples_leaf': 1, 'min_samples_split': 2}
DT_PP_OHP: {'criterion': 'entropy', 'max_depth': 5, 'max_features': None, 'min_samples_leaf': 8, 'min_samples_split': 2}	DT_PP_OHP: {'criterion': 'entropy', 'max_depth': 10, 'max_features': None, 'min_samples_leaf': 8, 'min_samples_split': 2}	DT_PP_OHP: {'criterion': 'gini', 'max_depth': 10, 'max_features': 'log2', 'min_samples_leaf': 4, 'min_samples_split': 15}
DT_PP_WFS_OHP: {'criterion': 'entropy', 'max_depth': 5, 'max_features': None, 'min_samples_leaf': 8, 'min_samples_split': 2}	DT_PP_WFS_OHP: {'criterion': 'entropy', 'max_depth': 10, 'max_features': None, 'min_samples_leaf': 8, 'min_samples_split': 2}	DT_PP_WFS_OHP: {'criterion': 'gini', 'max_depth': 10, 'max_features': 'log2', 'min_samples_leaf': 4, 'min_samples_split': 15}
DT_FFS_OHP: {'criterion': 'entropy', 'max_depth': 5, 'max_features': None, 'min_samples_leaf': 1, 'min_samples_split': 2}	DT_FFS_OHP: {'criterion': 'entropy', 'max_depth': None, 'max_features': 'sqrt', 'min_samples_leaf': 8, 'min_samples_split': 2}	DT_FFS_OHP: {'criterion': 'gini', 'max_depth': 5, 'max_features': 'sqrt', 'min_samples_leaf': 2, 'min_samples_split': 10}
DT_RES_OHP: {'criterion': 'entropy', 'max_depth': 5, 'max_features': None, 'min_samples_leaf': 4, 'min_samples_split': 2}	DT_RES_OHP: {'criterion': 'entropy', 'max_depth': 10, 'max_features': 'log2', 'min_samples_leaf': 2, 'min_samples_split': 5}	DT_RES_OHP: {'criterion': 'entropy', 'max_depth': 10, 'max_features': 'log2', 'min_samples_leaf': 6, 'min_samples_split': 2}
DT_RES_WFS_OHP: {'criterion': 'gini', 'max_depth': None, 'max_features': 'sqrt', 'min_samples_leaf': 8, 'min_samples_split': 2}	DT_RES_WFS_OHP: {'criterion': 'entropy', 'max_depth': None, 'max_features': None, 'min_samples_leaf': 2, 'min_samples_split': 10}	DT_RES_WFS_OHP: {'criterion': 'gini', 'max_depth': None, 'max_features': None, 'min_samples_leaf': 1, 'min_samples_split': 2}
DT_PP_FFS_OHP: {'criterion': 'gini', 'max_depth': 5, 'max_features': 'sqrt', 'min_samples_leaf': 6, 'min_samples_split': 15}	DT_PP_FFS_OHP: {'criterion': 'entropy', 'max_depth': None, 'max_features': 'sqrt', 'min_samples_leaf': 8, 'min_samples_split': 2}	DT_PP_FFS_OHP: {'criterion': 'gini', 'max_depth': None, 'max_features': None, 'min_samples_leaf': 1, 'min_samples_split': 2}
DT_PP_RES_OHP: {'criterion': 'gini', 'max_depth': 5, 'max_features': None, 'min_samples_leaf': 2, 'min_samples_split': 2}	DT_PP_RES_OHP: {'criterion': 'gini', 'max_depth': 15, 'max_features': None, 'min_samples_leaf': 1, 'min_samples_split': 10}	DT_PP_RES_OHP: {'criterion': 'gini', 'max_depth': None, 'max_features': None, 'min_samples_leaf': 8, 'min_samples_split': 2}
DT_PP_RES_WFS_OHP: {'criterion': 'gini', 'max_depth': None, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 5}	DT_PP_RES_WFS_OHP: {'criterion': 'entropy', 'max_depth': None, 'max_features': None, 'min_samples_leaf': 2, 'min_samples_split': 5}	DT_PP_RES_WFS_OHP: {'criterion': 'gini', 'max_depth': None, 'max_features': None, 'min_samples_leaf': 1, 'min_samples_split': 2}
DT_RES_FFS_OHP: {'criterion': 'gini', 'max_depth': None, 'max_features': None, 'min_samples_leaf': 8, 'min_samples_split': 2}	DT_RES_FFS_OHP: {'criterion': 'entropy', 'max_depth': 5, 'max_features': None, 'min_samples_leaf': 4, 'min_samples_split': 20}	DT_RES_FFS_OHP: {'criterion': 'gini', 'max_depth': None, 'max_features': None, 'min_samples_leaf': 1, 'min_samples_split': 5}

Continuação da Tabela C.1 – Parâmetros ótimos encontrados na *grid search* para os modelos DT

<i>Australian</i>	<i>German</i>	<i>Japanese</i>
DT_PP_RES_OHP: {'criterion': 'entropy', 'max_depth': 10, 'max_features': 'sqrt', 'min_samples_leaf': 8, 'min_samples_split': 2}	DT_PP_RES_OHP: {'criterion': 'gini', 'max_depth': 10, 'max_features': None, 'min_samples_leaf': 2, 'min_samples_split': 15}	DT_PP_RES_OHP: {'criterion': 'gini', 'max_depth': None, 'max_features': None, 'min_samples_leaf': 1, 'min_samples_split': 2}

Fonte: Autoria própria (2023)

ANEXO D – PARÂMETROS DOS MODELOS SVM COM OHP

Tabela D.1 – Parâmetros ótimos encontrados na *grid search* para os modelos SVM

<i>Australian</i>	<i>German</i>	<i>Japanese</i>
SVM_OHP: {'C': 250, 'gamma': 'scale', 'kernel': 'rbf'}	SVM_OHP: {'C': 5, 'gamma': 'scale', 'kernel': 'rbf'}	SVM_OHP: {'C': 50, 'gamma': 'scale', 'kernel': 'rbf'}
SVM_WFS_OHP: {'C': 15, 'gamma': 'scale', 'kernel': 'rbf'}	SVM_WFS_OHP: {'C': 0.5, 'gamma': 'scale', 'kernel': 'rbf'}	SVM_WFS_OHP: {'C': 5, 'gamma': 'auto', 'kernel': 'rbf'}
SVM_PP_OHP: {'C': 0.1, 'gamma': 'scale', 'kernel': 'rbf'}	SVM_PP_OHP: {'C': 5, 'gamma': 'auto', 'kernel': 'rbf'}	SVM_PP_OHP: {'C': 10, 'gamma': 'auto', 'kernel': 'rbf'}
SVM_PP_WFS_OHP: {'C': 0.1, 'gamma': 'scale', 'kernel': 'rbf'}	SVM_PP_WFS_OHP: {'C': 5, 'gamma': 'auto', 'kernel': 'rbf'}	SVM_PP_WFS_OHP: {'C': 10, 'gamma': 'auto', 'kernel': 'rbf'}
SVM_FFS_OHP: {'C': 30, 'gamma': 'scale', 'kernel': 'rbf'}	SVM_FFS_OHP: {'C': 15, 'gamma': 'auto', 'kernel': 'rbf'}	SVM_FFS_OHP: {'C': 500, 'gamma': 'scale', 'kernel': 'rbf'}
SVM_RES_OHP: {'C': 1000, 'gamma': 'scale', 'kernel': 'rbf'}	SVM_RES_OHP: {'C': 5, 'gamma': 'auto', 'kernel': 'rbf'}	SVM_RES_OHP: {'C': 250, 'gamma': 'scale', 'kernel': 'rbf'}
SVM_RES_WFS_OHP: {'C': 100, 'gamma': 'scale', 'kernel': 'rbf'}	SVM_RES_WFS_OHP: {'C': 500, 'gamma': 'scale', 'kernel': 'rbf'}	SVM_RES_WFS_OHP: {'C': 15, 'gamma': 'scale', 'kernel': 'rbf'}
SVM_PP_FFS_OHP: {'C': 0.1, 'gamma': 'scale', 'kernel': 'rbf'}	SVM_PP_FFS_OHP: {'C': 5, 'gamma': 'auto', 'kernel': 'rbf'}	SVM_PP_FFS_OHP: {'C': 0.5, 'gamma': 'scale', 'kernel': 'rbf'}
SVM_PP_RES_OHP: {'C': 1, 'gamma': 'scale', 'kernel': 'rbf'}	SVM_PP_RES_OHP: {'C': 20, 'gamma': 'scale', 'kernel': 'rbf'}	SVM_PP_RES_OHP: {'C': 5, 'gamma': 'scale', 'kernel': 'rbf'}
SVM_PP_RES_WFS_OHP: {'C': 5, 'gamma': 'scale', 'kernel': 'rbf'}	SVM_PP_RES_WFS_OHP: {'C': 1000, 'gamma': 'auto', 'kernel': 'rbf'}	SVM_PP_RES_WFS_OHP: {'C': 500, 'gamma': 'scale', 'kernel': 'rbf'}
SVM_RES_FFS_OHP: {'C': 45, 'gamma': 'scale', 'kernel': 'rbf'}	SVM_RES_FFS_OHP: {'C': 10, 'gamma': 'auto', 'kernel': 'rbf'}	SVM_RES_FFS_OHP: {'C': 10, 'gamma': 'scale', 'kernel': 'rbf'}
SVM_PP_RES_OHP: {'C': 15, 'gamma': 'scale', 'kernel': 'rbf'}	SVM_PP_RES_OHP: {'C': 500, 'gamma': 'auto', 'kernel': 'rbf'}	SVM_PP_RES_OHP: {'C': 20, 'gamma': 'scale', 'kernel': 'rbf'}

Fonte: Autoria própria (2023)

ANEXO E – PARÂMETROS DOS MODELOS RF COM OHP

Tabela E.1 – Parâmetros ótimos encontrados na *grid search* para os modelos RF

<i>Australian</i>	<i>German</i>	<i>Japanese</i>
RF_OHP: {'bootstrap': True, 'criterion': 'entropy', 'max_depth': 10, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 10, 'n_estimators': 10}	RF_OHP: {'bootstrap': True, 'criterion': 'gini', 'max_depth': None, 'max_features': None, 'min_samples_leaf': 2, 'min_samples_split': 20, 'n_estimators': 20}	RF_OHP: {'bootstrap': True, 'criterion': 'entropy', 'max_depth': 10, 'max_features': 'sqrt', 'min_samples_leaf': 2, 'min_samples_split': 2, 'n_estimators': 20}
RF_WFS_OHP: {'bootstrap': True, 'criterion': 'gini', 'max_depth': None, 'max_features': None, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 50}	RF_WFS_OHP: {'bootstrap': True, 'criterion': 'gini', 'max_depth': None, 'max_features': None, 'min_samples_leaf': 6, 'min_samples_split': 2, 'n_estimators': 10}	RF_WFS_OHP: {'bootstrap': True, 'criterion': 'gini', 'max_depth': None, 'max_features': None, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 50}
RF_PP_OHP: {'bootstrap': True, 'criterion': 'entropy', 'max_depth': 10, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 10, 'n_estimators': 10}	RF_PP_OHP: {'bootstrap': True, 'criterion': 'gini', 'max_depth': 10, 'max_features': None, 'min_samples_leaf': 1, 'min_samples_split': 15, 'n_estimators': 20}	RF_PP_OHP: {'bootstrap': False, 'criterion': 'gini', 'max_depth': None, 'max_features': 'log2', 'min_samples_leaf': 1, 'min_samples_split': 15, 'n_estimators': 50}
RF_PP_WFS_OHP: {'bootstrap': True, 'criterion': 'entropy', 'max_depth': 10, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 10, 'n_estimators': 10}	RF_PP_WFS_OHP: {'bootstrap': True, 'criterion': 'gini', 'max_depth': 10, 'max_features': None, 'min_samples_leaf': 1, 'min_samples_split': 15, 'n_estimators': 20}	RF_PP_WFS_OHP: {'bootstrap': False, 'criterion': 'gini', 'max_depth': None, 'max_features': 'log2', 'min_samples_leaf': 1, 'min_samples_split': 15, 'n_estimators': 50}
RF_FFS_OHP: {'bootstrap': False, 'criterion': 'gini', 'max_depth': 5, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 20, 'n_estimators': 10}	RF_FFS_OHP: {'bootstrap': False, 'criterion': 'entropy', 'max_depth': None, 'max_features': 'sqrt', 'min_samples_leaf': 10, 'min_samples_split': 2, 'n_estimators': 200}	RF_FFS_OHP: {'bootstrap': False, 'criterion': 'gini', 'max_depth': 5, 'max_features': 'sqrt', 'min_samples_leaf': 4, 'min_samples_split': 2, 'n_estimators': 20}
RF_RES_OHP: {'bootstrap': False, 'criterion': 'entropy', 'max_depth': 10, 'max_features': 'sqrt', 'min_samples_leaf': 6, 'min_samples_split': 2, 'n_estimators': 10}	RF_RES_OHP: {'bootstrap': True, 'criterion': 'gini', 'max_depth': 10, 'max_features': 'log2', 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 100}	RF_RES_OHP: {'bootstrap': True, 'criterion': 'entropy', 'max_depth': None, 'max_features': 'log2', 'min_samples_leaf': 1, 'min_samples_split': 10, 'n_estimators': 20}
RF_RES_WFS_OHP: {'bootstrap': False, 'criterion': 'entropy', 'max_depth': 5, 'max_features': None, 'min_samples_leaf': 1, 'min_samples_split': 15, 'n_estimators': 10}	RF_RES_WFS_OHP: {'bootstrap': True, 'criterion': 'entropy', 'max_depth': 10, 'max_features': 'sqrt', 'min_samples_leaf': 2, 'min_samples_split': 10, 'n_estimators': 10}	RF_RES_WFS_OHP: {'bootstrap': True, 'criterion': 'gini', 'max_depth': None, 'max_features': None, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 50}
RF_PP_FFS_OHP: {'bootstrap': False, 'criterion': 'entropy', 'max_depth': None, 'max_features': 'sqrt', 'min_samples_leaf': 6, 'min_samples_split': 2, 'n_estimators': 10}	RF_PP_FFS_OHP: {'bootstrap': False, 'criterion': 'entropy', 'max_depth': None, 'max_features': 'sqrt', 'min_samples_leaf': 10, 'min_samples_split': 2, 'n_estimators': 200}	RF_PP_FFS_OHP: {'bootstrap': True, 'criterion': 'gini', 'max_depth': None, 'max_features': None, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 20}
RF_PP_RES_OHP: {'bootstrap': True, 'criterion': 'entropy', 'max_depth': None, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 5, 'n_estimators': 100}	RF_PP_RES_OHP: {'bootstrap': False, 'criterion': 'gini', 'max_depth': None, 'max_features': 'log2', 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 100}	RF_PP_RES_OHP: {'bootstrap': False, 'criterion': 'entropy', 'max_depth': 20, 'max_features': 'log2', 'min_samples_leaf': 1, 'min_samples_split': 5, 'n_estimators': 10}

Continuação da Tabela E.1 – Parâmetros ótimos encontrados na *grid search* para os modelos RF

<i>Australian</i>	<i>German</i>	<i>Japanese</i>
RF_PP_RES_WFS_OHP: {'bootstrap': True, 'criterion': 'gini', 'max_depth': None, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 20, 'n_estimators': 50}	RF_PP_RES_WFS_OHP: {'bootstrap': True, 'criterion': 'entropy', 'max_depth': 20, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 15, 'n_estimators': 50}	RF_PP_RES_WFS_OHP: {'bootstrap': True, 'criterion': 'gini', 'max_depth': None, 'max_features': None, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 10}
RF_RES_FFS_OHP: {'bootstrap': True, 'criterion': 'entropy', 'max_depth': 5, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 5, 'n_estimators': 10}	RF_RES_FFS_OHP: {'bootstrap': True, 'criterion': 'entropy', 'max_depth': None, 'max_features': 'sqrt', 'min_samples_leaf': 4, 'min_samples_split': 10, 'n_estimators': 20}	RF_RES_FFS_OHP: {'bootstrap': True, 'criterion': 'gini', 'max_depth': None, 'max_features': None, 'min_samples_leaf': 1, 'min_samples_split': 5, 'n_estimators': 10}
RF_PP_RES_OHP: {'bootstrap': True, 'criterion': 'gini', 'max_depth': None, 'max_features': None, 'min_samples_leaf': 10, 'min_samples_split': 2, 'n_estimators': 10}	RF_PP_RES_OHP: {'bootstrap': False, 'criterion': 'entropy', 'max_depth': 10, 'max_features': 'sqrt', 'min_samples_leaf': 8, 'min_samples_split': 20, 'n_estimators': 20}	RF_PP_RES_OHP: {'bootstrap': True, 'criterion': 'gini', 'max_depth': None, 'max_features': None, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 20}

Fonte: Autoria própria (2023)

APÊNDICE - Artigos Publicados em Congressos e Periódicos

Artigos submetidos e publicados em anais de congressos:

- MONTEVECHI, A. A.; MIRANDA, R. C.; MEDEIROS, A. L.; AMARAL, J.V.S. Machine Learning na análise do risco de crédito: uma aplicação da regressão logística e algoritmos de resampling. In: XLII Encontro Nacional de Engenharia de Produção, *Anais...* Online. 2022.
- MONTEVECHI, A. A.; MIRANDA, R. C.; MEDEIROS, A. L.; AMARAL, J.V.S. Modelagem de um classificador SVM para análise do risco de crédito. In: LIV Simpósio Brasileiro de Pesquisa Operacional, *Anais...* Online. 2022.
- MONTEVECHI, A. A.; MIRANDA, R. C.; MEDEIROS, A. L. O papel da seleção de variáveis na modelagem do risco de crédito. Submetido para: XLIII Encontro Nacional de Engenharia de Produção. Data da submissão: 23/05/2023.
- MONTEVECHI, A. A.; MIRANDA, R. C.; MEDEIROS, A. L.; LÚCIO, J. S. Proposta de frameworks para a modelagem de regressão logística na classificação do risco de crédito. Submetido para: LV Simpósio Brasileiro de Pesquisa Operacional. Data da submissão: 11/07/2023.

Artigos submetidos para publicação em periódicos:

- MONTEVECHI, A. A.; MIRANDA, R. C.; MEDEIROS, A. L.; MONTEVECHI, J. A. B. Machine Learning in consumer credit risk modelling: a systematic literature review. Artigo submetido a revista: *International Journal of Production Research*. Data da submissão: 26/07/2023.