UNIVERSIDADE FEDERAL DE ITAJUBÁ - UNIFEI
PROGRAMA DE PÓS-GRADUAÇÃO EM
CIÊNCIA E TECNOLOGIA DA COMPUTAÇÃO

Detecção de Insatisfação de Servidores Públicos com Inteligência Artificial.

André Luiz Alves Dias

Itajubá, 21 de dezembro de 2023

UNIVERSIDADE FEDERAL DE ITAJUBÁ - UNIFEI
PROGRAMA DE PÓS-GRADUAÇÃO EM
CIÊNCIA E TECNOLOGIA DA COMPUTAÇÃO

André Luiz Alves Dias

Detecção de Insatisfação de Servidores Públicos com Inteligência Artificial.

Dissertação submetida ao Programa de Pós-Graduação em Ciência e Tecnologia da Computação como parte dos requisitos para obtenção do Título de Mestre em Ciência e Tecnologia da Computação.

Área de Concentração: Matemática da Computação

Orientador: Prof. Dr. Carlos Henrique Valério de Moraes

21 de dezembro de 2023
Itajubá

UNIVERSIDADE FEDERAL DE ITAJUBÁ - UNIFEI
PROGRAMA DE PÓS-GRADUAÇÃO EM
CIÊNCIA E TECNOLOGIA DA COMPUTAÇÃO

Detecção de Insatisfação de Servidores Públicos com Inteligência Artificial.

André Luiz Alves Dias

Banca Examinadora:
Prof. Dr. Ahmed Ali Abdalla Esmin
Prof. Dr. João Paulo Reus Rodrigues Leite

Itajubá
2023

André Luiz Alves Dias Detecção de Insatisfação de Servidores Públicos com Inteligência Artificial. / André Luiz Alves Dias– Itajubá-MG, 2023.
107 f.
Orientador: Prof. Dr. Carlos Henrique Valério de Moraes
Dissertação (Mestrado)- Universidade Federal de Itajubá - UNIFEI, Programa de pós-graduação em CIÊNCIA E TECNOLOGIA DA COMPUTAÇÃO, 2023.
1. Insatisfação. 2. Servidor Público. 3. Inteligência Artificial. I. Prof. Dr. Carlos Henrique Valério de Moraes, orient. II. Detecção de Insatisfação de Servidores Públicos com Inteligência Artificial.
A474d

Agradecimentos

Aos meus pais, por terem me dado as oportunidades para eu chegar até aqui, por sempre terem acreditado que seria possível e me incentivado a continuar.

A minha namorada Daniella e a família dela, por terem se privado de tempo comigo para que fosse possível a realização dos estudos e desenvolvimento da dissertação.

A Deus, que me provê com saúde e intelecto para poder estudar, além de ter me trazido por todo este caminho até aqui.

Aos meus colegas da Diretoria de Desenvolvimento de Pessoal que sempre me apoiaram na realização de atividades e sempre estiveram dispostos a colaborar com ideias para a execução da pesquisa.

Ao meu orientador, Prof. Dr. Carlos Henrique por todo o incentivo, contribuição e cujo apoio, principalmente nos momentos em que me encontrava perdido, foi fundamental durante a execução desse projeto. Acredito que ele seja um dos melhores orientadores que a UNIFEI possui em seus cursos.

"Não há nada a temer na vida, apenas tratar de compreender." — Marie Curie

Resumo

Este trabalho destaca uma pesquisa abrangente sobre a aplicação da Inteligência Artificial (I.A.) na gestão de recursos humanos, com um foco específico na identificação da insatisfação dos funcionários por meio de abordagens de aprendizado de máquina. A investigação incluiu uma revisão de artigos científicos que discutiam tanto a implementação da I.A. no contexto de recursos humanos quanto o uso de técnicas de aprendizado de máquina para detectar casos de *turnover/attrition*, além da relação de insatisfação e os casos de *turnover/attrition*. Para avaliar essas abordagens, foram selecionadas quatro bases de dados públicas validadas. Três delas continham dados fictícios de funcionários e uma continha dados reais de *turnover* de funcionários. Cada base de dados passou por um processo de fatorização de campos textuais, seguido por análises para destacar as distribuições dos dados em cada conjunto. Na condução da pesquisa, diferentes abordagens de aprendizado de máquina foram aplicadas a cada uma das bases, com o objetivo de verificar a viabilidade de identificar a insatisfação por meio da I.A. As técnicas utilizadas incluíram detecção de anomalias ou novidades, classificadores e conjuntos de classificadores otimizados. Os resultados foram quantificados, revelando pontuações promissoras, com desempenhos superiores a 90%. Esses resultados destacam a eficácia geral do aprendizado de máquina na identificação da insatisfação dos funcionários, demonstrando seu potencial para aplicações práticas no ambiente de recursos humanos.

Palavras-chaves: Inteligência Artificial, *turnover*, *attrition*, aprendizado de máquina, insatisfação de funcionário.

Abstract

This work highlights a comprehensive investigation into the application of Artificial Intelligence (A.I.) in human resources management, with a specific focus on identifying employee dissatisfaction through machine learning approaches. The research included a review of scientific articles discussing both the implementation of A.I. in the context of human resources and the use of machine learning techniques to detect cases of turnover/attrition, along with the relationship between dissatisfaction and turnover/attrition cases. To assess these approaches, four validated public databases were selected. Three of them contained fictional employee data, and one contained real employee turnover data. Each database underwent a process of textual field factorization, followed by analyses to highlight the data distributions in each set. In conducting the research, different machine learning approaches were applied to each of the databases, aiming to verify the feasibility of identifying dissatisfaction through A.I. The techniques used included anomaly or novelty detection, classifiers, and optimized sets of classifiers. The results were quantified, revealing promising scores, with performances exceeding 90%. These results emphasize the overall effectiveness of machine learning in identifying employee dissatisfaction, demonstrating its potential for practical applications in the human resources environment.

Key-words: Artificial Inteligence, turnover, attrition, machine learning, employee dissatisfaction.

Lista de ilustrações

Lista de tabelas

Sumário

1 Introdução

O serviço público brasileiro tem como objetivo básico o atendimento à sociedade no campo em que está inserido, fornecendo serviços como educação, saúde, assistência social, segurança pública, entre outros. Nas últimas décadas, mudanças vindas da globalização e redemocratização do estado brasileiro têm buscado um modelo de gestão que gerasse ganho de eficiência e maior participação da sociedade em sua gestão, conforme mencionado por Ferreira et al.(2).

Ademais, de acordo com Affonso e Martins(3), a missão básica de qualquer organização, principalmente as públicas, é o pleno atendimento à sociedade na qual ela está inserida, sendo necessária uma busca pela melhoria da qualidade e produtividade nas organizações. Para que essas melhorias sejam atingidas, os servidores públicos devem ser considerados e analisados como parte essencial desses objetivos, pois é a produtividade e satisfação destes que proporcionarão o atendimento à sociedade.

Os servidores públicos são admitidos mediante concurso público, um rigoroso processo de seleção, que busca selecionar os melhores candidatos para os cargos e funções públicas. Mas, apesar de todo esse processo, a maioria dos órgãos não aproveita o potencial de seus colaboradores, comprometendo assim a qualidade dos serviços prestados. Segundo Carvalho e Silva(4) isso acabou gerando a figura estereotipada e generalista do servidor público em nossa cultura, como alguém que trabalha pouco e ganha muito. Como forma de reverter essa situação, a administração pública deve dar uma atenção maior para a motivação dos seus servidores, pois dentro das teorias da motivação, a satisfação no trabalho está relacionada ao quão motivado e satisfeito o colaborador está dentro da organização, do qual o bom desempenho pode resultar no crescimento profissional, conforme Silva, Diniz e Pellizzoni(5).

A motivação do servidor público não é uma tarefa fácil de ser alcançada, principalmente quando o contexto das organizações públicas é considerado, pois a distribuição de cargos e tarefas é, em alguns casos, feita baseada em favoritismo político segundo Carvalho e Silva(4). Portanto foi identificado que a motivação do servidor está relacionada diretamente à sua satisfação com o seu ambiente de trabalho. Segundo Carvalho, Falce e Guimarães(6), a motivação de serviço público é um tipo de motivação multidimensional, orientada para o outro, e consiste na vontade e desejo de contribuir para o interesse público e a sociedade em geral.

Andrade(7) cita que os indivíduos se sentem atraídos por uma organização na expectativa de que seus valores pessoais combinem com os valores da organização, porém, eles podem descobrir, depois de algum tempo, que não se encaixam na organização e

experimentam o que é chamado de atrito (*attrition*). Esse desacordo pode resultar na falta de satisfação no trabalho, influenciando assim seu engajamento, resultando em demissão por parte da organização ou por iniciativa própria. A questão é que no serviço público, devido à estabilidade garantida na legislação vigente, o indivíduo aprovado em concurso público tende a permanecer na mesma instituição até a sua aposentadoria, mesmo quando experimenta o atrito, o que acaba por impactar seu engajamento e, consequentemente, a qualidade do serviço prestado por ele.

Já em empresas privadas, essa falta de satisfação pode gerar o que é chamado de rotatividade de funcionários (*employee turnover*), que é o fato do funcionário trocar de trabalho, gerando prejuízo para a organização pois o tempo e os custos de repor esta vaga são altos, além de prejudicar a produtividade da empresa. No entanto, no serviço público essa insatisfação pode reduzir o desempenho do servidor e impactar o ambiente de trabalho, sobrecarregando os demais servidores do setor e diminuindo a qualidade da prestação de serviços à sociedade. Isso demonstra que a satisfação no trabalho também pode ser atrelada ao bem estar e saúde do trabalhador, enquanto que a insatisfação do servidor pode trazer malefícios físicos, mentais e sociais, o que propicia problemas organizacionais e ao ambiente de trabalho.

Nesse cenário, a importância de uma eficiente gestão de pessoas em um ambiente organizacional, público ou privado, é notória, pois se trata de um instrumento capaz de melhorar a produtividade e a qualidade de um serviço, conforme Carvalho e Silva(4). Atualmente, a gestão de pessoas pode se beneficiar do avanço da tecnologia no serviço público brasileiro, no qual um dos pontos de destaque é a inteligência artificial (I.A.) desempenhando um papel cada vez mais importante, em que as tecnologias de I.A. são utilizadas para automatizar tarefas de rotina, analisar grandes quantidades de dados e fornecer *insights* que podem auxiliar na tomada de decisões.

A automação de processos robóticos (RPA - *Robotic Process Automation*) e suas tecnologias associadas, como a Inteligência Artificial (I.A.) e o Aprendizado de Máquina (ML), são essenciais para a transformação digital dentro do serviço público, que inclui uma gama de atividades desde a gestão de recursos humanos até a prestação de serviços essenciais. Essas tecnologias podem oferecer soluções significativas para o problema da rotatividade de funcionários ao analisar dados de desempenho e satisfação dos servidores, identificando padrões que antecedem a saída de um empregado e alertando a gestão para possíveis insatisfações e reduções na produtividade.

Considerado um ramo da I.A., o aprendizado de máquina (ML - *Machine Learning*) estuda formas de fazer com que computadores melhorem sua performance com base na experiência, sendo capazes de detectar padrões em dados de forma automática, e posteriormente usar esses padrões para prever dados futuros ou desempenhar outras formas de tomada de decisão, conforme Desordi e Bona(8).

O aprendizado de máquina se aproxima de ensinar as máquinas a chegarem a um resultado, mostrando-lhes muitos exemplos de resultados - chamados de "treinamento". Outra abordagem é os humanos definirem um conjunto de amplas regras que geralmente permitem que a máquina aprenda por conta própria por tentativa e erro, de acordo com Cedraz, Arraes e TCU(9).

Os tipos de aprendizagem de máquina existentes são a aprendizagem supervisionada que é quando a máquina, para conseguir fazer previsões a partir de informações de entrada, precisa ter sido treinada com exemplos das informações de entrada e informações de saída. A aprendizagem não supervisionada que é voltada para a identificação de padrões nos dados sem a necessidade de uma supervisão humana, exemplo desse tipo de aprendizagem incluem os agrupamentos, a detecção de anomalias e outros que têm por objetivo extrair algum grau útil de informação a partir dos dados, conforme citado em Cedraz, Arraes e TCU(9). Por fim, existe a aprendizagem por reforço que é aquela que não tem instruções sobre como atingir o objetivo, e o sistema de I.A. tem que observar o estado do ambiente ao seu redor, agir sobre ele e colher a recompensa sobre sua ação para que possa aprender de forma interativa a descobrir qual a melhor maneira para atingir seu objetivo.

Com as técnicas de aprendizado de máquina mais comumente usadas, o presente trabalho pretende fazer um comparativo entre as técnicas na identificação da rotatividade de funcionários utilizando bancos de dados públicos e associar as melhores técnicas para detectar a insatisfação de servidores em uma universidade pública.

A pesquisa foi iniciada com o levantamento de trabalhos científicos que indicam a aplicação de aprendizado de máquina na gestão de recursos humanos e dentre as tarefas que estão tendo aplicação da I.A., houve destaque no tema de *turnover/attrition* por ser algo que gera custos para as organizações e preocupação para os gestores. Além disso, a pesquisa tem o objetivo de aplicar as técnicas de aprendizado de máquina com melhores resultados no serviço público, para identificar os servidores insatisfeitos e com baixa produtividade.

Dentre os trabalhos identificados, vale destacar o de Chowdhury et al.(10), que apresentaram as possibilidades de aplicação da I.A. na gestão de recursos humanos e propôs um *framework* para analisar se a organização está preparada para a implantação da I.A. e para preparar estratégias para adotar e implantar práticas e processos de I.A. na gestão de recursos humanos.

Já Birzniece et al.(11) sugeriu um sistema de predição de rotatividade voluntária de funcionários que aplicou a clusterização nos dados por não conhecerem, inicialmente, a estrutura dos dados e o algoritmo de árvore de decisão CART (Classification and Regression Tree) a fim de identificar os funcionários com maior chance de rotatividade voluntária e desenvolveu uma interface gráfica para os resultados ficarem mais amigáveis aos gestores

possibilitando uma melhor análise.

Neste sentido da rotatividade, Sadana e Munnuru(12) explicou o básico de um *pipeline* para prever o atrito na força de trabalho de uma empresa de T.I. Já Shankar et al.(13) aplicou diferentes técnicas de aprendizado de máquina em um banco de dados público da IBM para analisar qual das técnicas apresentou melhor resultado, sendo que um classificador *Multi Layer Perceptron(MLP)* foi o que gerou a melhor generalização.

Enquanto que Jin Jiaxing Shang e Qiang(14) propuseram um algoritmo de um modelo híbrido que combina análise de sobrevivência com conjuntos de aprendizado para a predição do comportamento de rotatividade, baseado em uma perspectiva centrada em evento.

De forma a facilitar a implantação da I.A., Johnson, Coggburn e Llorens(15) apresentaram considerações importantes para a pesquisa e a prática da implantação da I.A. na gestão de recursos humanos na área pública, como em quais tarefas de *Human Resources Management* (HRM) a I.A. poderia ser utilizada, quais os impactos de sua implantação nos trabalhadores e no trabalho, além de questões que envolvem o papel da tomada de decisão sugerida pela I.A., destacando as diversas formas que a tecnologia poderia ser utilizada para atingir os objetivos estratégicos da gestão de recursos humanos.

Considerando a área pública, em 2021, o Tribunal de Contas da União, órgão de controle externo do governo federal, realizou um levantamento sobre o estágio atual da utilização de tecnologias de I.A. por diversas organizações que compõem a Administração Pública, sob vários aspectos, conforme apresentado por Cedraz, Arraes e TCU(9). Esse levantamento foi feito por meio de formulário eletrônico encaminhado a 293 organizações abrangidas no escopo, sobre como a I.A. é utilizada para resolver problemas de negócio e agregar valor nos serviços prestados aos cidadãos.

De maneira semelhante, Felipe(16) realizou um levantamento sobre o estágio de desenvolvimento de iniciativas do uso de I.A. para o setor jurídico e destacou que existe a prevalência de metodologias e algoritmos bastante utilizados fora do contexto jurídico, especialmente comum na mineração de textos e classificação de documentos.

A integração de *Robotic Process Automation*(RPA) inteligente nas práticas de gestão de pessoas no serviço público pode mitigar os impactos negativos da rotatividade de funcionários. Sistemas de I.A., por exemplo, podem ser treinados para identificar sinais de descontentamento ou estresse entre os servidores, permitindo intervenções proativas para melhorar o engajamento e a satisfação no trabalho. Além disso, RPA pode assumir tarefas rotineiras e repetitivas, liberando os servidores para funções mais estratégicas e gratificantes, o que potencialmente aumentaria a satisfação no trabalho e diminuiria a rotatividade.

Por meio do uso dessas tecnologias, a administração pública pode evoluir para um

ambiente de trabalho onde a automação não apenas aumenta a eficiência, mas também contribui para um clima organizacional mais satisfatório e engajador, fundamental para a retenção de talentos e para uma prestação de serviço público de alta qualidade.

1.1 Motivação

Por se tratar de uma pesquisa que tem por objetivo buscar uma aplicação na área de gestão de pessoas de uma universidade pública, houve o interesse de identificar quais utilizações da I.A. na gestão de recursos humanos estavam ocorrendo e quais os respectivos resultados. A universidade pública em questão possui 881 servidores ativos e nos últimos anos tem perdido seus servidores por motivos diversos, tais como exoneração à pedido, redistribuição, vacância para posse em outro cargo, aposentadoria. Nos casos onde a reposição dessa força de trabalho é possível, a instituição leva de três a oito meses para conseguir executar todo o processo e preencher o cargo vago. Foi observado que a existência da insatisfação pelos empregados pode levar à ocorrência de *turnover/attrition*, situação em que o funcionário deixa a empresa.

A pesquisa realizada possui o intuito de comparar as abordagens de I.A. que produzem um bom resultado na identificação dos casos de *turnover/attrition* e tentar, com estas técnicas, identificar a insatisfação dos servidores da universidade para que seja possível agir preventivamente, antes desse servidor se desligar da instituição.

Em muitos casos, devido à estabilidade, o servidor mesmo insatisfeito não se desliga do quadro da universidade e pode acabar gerando problemas para a instituição e para os usuários de seus serviços, por isso as técnicas de I.A. que serão levantadas na pesquisa serão avaliadas e os seus resultados serão considerados. As melhores técnicas serão aplicadas nos dados dos servidores da universidade pública para que seja possível identificar a insatisfação e tratá-la, seja com acompanhamento do servidor, ou até mesmo uma tentativa de remoção para que ele possa ter mais satisfação em fazer parte da força de trabalho da universidade.

1.2 Objetivos

O objetivo dessa pesquisa consiste em adquirir conhecimento acerca da aplicação da Inteligência Artificial nas estratégias de aprendizado de máquina para aprimorar a gestão de recursos humanos, especialmente concentrando-se na detecção de sinais de insatisfação entre os colaboradores, fatores estes que frequentemente resultam em casos de *turnover/attrition*. As questões de pesquisa que guiaram o estudo foram:

QP01 - Quais as possíveis aplicações da I.A. na gestão de recursos humanos?

QP02 - Existem aplicações que identificam casos de *turnover/attrition* de um funcionário?

QP03 - A aplicação da I.A. pode ser feita em uma universidade pública?

Com base nessas questões será possível pesquisar as diferentes possibilidades que a aplicação da I.A. tem fornecido à gestão de Recursos Humanos.

Os objetivos específicos desta pesquisa são os seguintes:

- Realizar busca por bases de dados públicas e validadas em artigos que abordem a detecção de rotatividade (*turnover*) ou *attrition* em empresas reais ou sintéticas;
- Aplicar abordagens diversas de aprendizado de máquina para a extração de padrões e identificação de sinais de *turnover/attrition* entre os funcionários;
  - Investigar a eficácia de detectores de anomalias ou novidades devido à sua capacidade de identificar padrões de informação sem a necessidade de tutor no treinamento;
  - Empregar técnicas de aprendizado classificador clássico devido às suas características simples e eficientes, tornando-as adequadas para aplicações em computadores pessoais;
  - Propor uma estratégia de otimização em conjuntos de aprendizado classificador;
  - Avaliar a viabilidade do uso de conjuntos de aprendizado classificador com otimização por algoritmo genético, considerando o elevado tempo computacional envolvido.

Cabe ressaltar que não será adotado o aprendizado profundo nesta pesquisa devido ao alto custo computacional, tornando-se impraticável para o ambiente comum dos órgãos de gestão de pessoas.

1.3 Organização do Documento

A pesquisa está organizada neste documento da seguinte forma:

- **Capítulo 2 - Referencial Teórico:** Apresenta como foi pesquisado e quais as pesquisas utilizadas como referencial teórico deste documento.
- **Capítulo 3 - Metodologia:** Apresenta a metodologia de execução da pesquisa, desde a seleção das bases de dados até as métricas utilizadas, passando pelas análises de dados e apresentação das técnicas utilizadas na pesquisa.

- **Capítulo 4 - Resultados:** Apresenta os resultados obtidos com a execução de cada técnica utilizada.
- **Capítulo 5 - Conclusão:** Apresenta as conclusões obtidas com o trabalho e indica os possíveis trabalhos futuros a serem executados.

2 Referencial Teórico

Por envolver um estudo destinado a ampliar conhecimentos aplicáveis à pró-reitoria de gestão de pessoas em uma universidade pública, com foco na linha de pesquisa em Inteligência Artificial (I.A.), a investigação teve início com uma busca sobre a aplicação de técnicas de I.A. na gestão de recursos humanos. Diante dos resultados, houve destaque entre os temas, a aplicação dessa tecnologia na gestão da rotatividade de funcionários, um desafio recorrente no quadro da universidade anualmente. Além disso, foi considerada também a insatisfação no serviço público, visando identificar as consequências desse fenômeno no ambiente de trabalho.

Inicialmente, foi realizada a busca no *Google Scholar* com o termo *"Artificial Intelligence in Human Resources"*, limitando os trabalhos publicados até 2018. Foram escolhidos artigos que atendiam aos critérios presentes na Tabela 1. De acordo com os artigos obtidos, foi possível perceber uma relação entre a insatisfação no trabalho e os casos de *turnover/attrition*. Posteriormente, foi feita uma busca adicional no *Google Scholar* com o termo "Insatisfação no serviço público", também limitada a publicações até 2018, e foram selecionados outros artigos que atendiam aos critérios específicos sobre insatisfação. Em ambas as pesquisas, alguns artigos anteriores à data limite foram incluídos por serem considerados relevantes para a pesquisa.

Uma vez realizada a seleção dos trabalhos que seriam utilizados na pesquisa, houve a identificação do fato de que diversas técnicas de aprendizado de máquina são utilizadas para prever *turnover/attrition* em empresas privadas e, em alguns casos, também são utilizadas no serviço público. Na sequência serão apresentados os resultados da seleção dos artigos que contribuíram para o tema de pesquisa nas seções seguintes.

2.0.1 Possibilidades da I.A. na gestão de recursos humanos

Em um contexto mais amplo, Desordi e Bona(8) pesquisou sobre o uso da I.A. na Administração Pública, citando os casos do Tribunal de Contas da União que utiliza de sistemas inteligentes para aumentar a sua produtividade, como as robôs Alice, Sofia e Monica, que fazem uma varredura em contratações federais, buscando possíveis irregula-

Tabela 1 – Critérios de seleção de artigos

Critérios de seleção	
Critérios de seleção	Critérios de exclusão
CI01 - Artigo fala sobre I.A. no RH	CE01 - Artigo não fala sobre I.A. mas somente sobre RH
CI02 - Artigo fala sobre aprendizado de máquina no RH	CE02 - Artigo fala sobre I.A. mas não no RH
CI03 - Artigo fala sobre I.A. no RH de setor público	CE03 - Artigo menciona apenas políticas públicas e inovação
CI04 - Artigo fala sobre insatisfação no setor público	CE04 - Artigo não menciona insatisfação no setor público

ridades em editais de licitações e atas de registro de preços, sendo utilizadas não somente pelo TCU, mas também pela Controladoria Geral da União, o Ministério Público Federal, a Polícia Federal e Tribunais de Contas estaduais. Desordi ainda cita casos estaduais de desenvolvimento de sistemas que fazem uso de I.A., além de mencionar aplicações no contexto de controle público, como da Operação Serenata de Amor que faz uma varredura nos gastos públicos reembolsados pela Cota para Exercício de Atividade Parlamentar (CEAP) de deputados federais e senadores e destaca que a utilização da I.A. pode auxiliar no aumento da eficiência da Administração Pública, uma vez que dinamizam, modernizam e desburocratizam a atividade pública.

Já Silva, Diniz e Pellizzoni(5) realizou um estudo sobre a tendência do uso da I.A. no setor público e como essa tecnologia pode complementar, requalificar ou substituir a força de trabalho humano. Também foram identificadas algumas áreas de aplicação da I.A. no setor público, como interfaces de usuário de sistemas de software, interação humano-computador para tarefas repetitivas como entrada de dados, agentes virtuais (chatbots e avatares), análise preditiva com aprendizagem de máquina e visualização de dados, que são *softwares* combinados com *big data* de análise avançada de dados. Outro ponto identificado pelo autor foi o fato de que pode haver resistência por parte das organizações e dos empregados pela possibilidade de diminuição de empregos, mas é interessante frisar que com o advento do uso da tecnologia novos empregos serão criados e que o setor deve buscar um ajuste entre a força de trabalho e a I.A. com a busca de ganhos da sua aplicação e seus efeitos junto aos empregados.

No contexto de gestão de recursos humanos, Johnson, Coggburn e Llorens(15) apresentou em quais áreas da gestão de recursos humanos a I.A. poderia ser aplicada com destaque para aquisição de talentos, desenvolvimento de pessoal, gestão de desempenho, compensação e *turnover* e retenção, explicando sua aplicação nessas áreas e as questões que devem ser observadas. Chowdhury et al.(10) propôs um *framework* que demonstra a capacidade de integrar a I.A. nos processos e práticas de recursos humanos no contexto da empresa. De maneira semelhante, Kumar et al.(17) cita áreas e benefícios da aplicação de I.A. e aprendizado de máquina, como em práticas de gestão para monitoramento do engajamento dos empregados e nas avaliações de desempenho dos empregados, concluindo que pode ocorrer uma melhora na acurácia das decisões e na eficiência operacional com a utilização de I.A. e ML.

Pampouktsi et al.(18) realizou uma pesquisa no serviço público grego, por meio de um *survey* dividido em duas partes, na qual a primeira levantou a qualificação desejada para alguns cargos e a segunda, levantou as qualificações dos empregados públicos e suas qualificações. Com essas informações, foi possível testar o desempenho de algoritmos de aprendizado de máquina supervisionados para realizar a seleção e o posicionamento dos recursos humanos no serviço público. De forma semelhante Mathew, Chacko e Udhaya-

kumar(19) utilizou um banco de dados público presente na plataforma *Kaggle* (20) para comparar a acurácia de diversos algoritmos de classificação de acordo com a quantidade de características e classes presentes na base de dados. O autor menciona que a escolha do algoritmo deve ser feita com base nas características da base para que seja possível alcançar a maior acurácia possível do algoritmo selecionado.

Vivek e Yawalkar(21) realizou um estudo onde identificou que a I.A. poderia ser aplicada em diversas áreas do departamento de recursos humanos de indústrias, sendo utilizada para automatizar etapas do processo de recrutamento e seleção, gerando uma maior transparência por realizar a seleção de currículos baseado na descrição do trabalho, além de mencionar que a I.A. é aplicada para automatizar tarefas repetitivas, aumentando assim a eficiência no ambiente de trabalho. No entanto, apesar dos benefícios, também são citados os desafios da implantação da I.A., entre os quais estão o conjunto de habilidades necessárias por parte dos empregados para a utilização da inteligência artificial no departamento de recursos humanos e a preocupação que esta implantação pode trazer para a mente dos empregados. Embora existam dificuldades, a maioria dos pesquisadores e especialistas recomendam que as indústrias utilizem ferramentas de inteligência artificial.

2.0.2 Turnover, Attrition e Satisfação

Turnover ou rotatividade está relacionada com o fato do empregado mudar de emprego, normalmente por escolha própria, o que acaba gerando problemas para as empresas, como investimento em treinamento de um substituto, sobrecarga do pessoal que permaneceu na empresa, necessidade de recrutar um novo funcionário para a vaga desocupada. De forma semelhante ocorre o *Attrition* ou atrito, que é o termo que se refere ao fato de desocupar uma vaga, seja pela aposentadoria do ocupante, seja pela reestruturação da empresa, sendo que sua diferença é que essa vaga normalmente não é reposta. Artigos que tratam sobre esses dois assuntos foram considerados, nessa pesquisa que se baseou nas técnicas de I.A. apresentadas para prever a possibilidade das situações ocorrerem tomando como base os dados dos funcionários das empresas.

Satisfação e *turnover/attrition* estão relacionados, conforme citado em Sadana e Munnuru(12), Al-Darraji et al.(22), Shankar et al.(13), Qutub et al.(23), Srivastava e Eachempati(24), Kang, Croft e Bichelmeyer(25), Yuan(26), Judrups et al.(27), Chai, Qian e Wang(28), Punnoose e Xavier(29), Kang et al.(30) e Andrade(7). As bases de dados utilizadas pela maioria deles tinham como característica considerada a satisfação, além disso os autores citam que a motivação e a satisfação são consideradas pelos funcionários antes de ocorrer o *turnover/attrition*.

Como a motivação inicial foi identificar o que poderia ser aplicado de I.A. em uma universidade pública, foram levantados trabalhos que trouxeram uma visão da gestão de pessoas no setor público, como Ferreira et al.(2) que realizou uma pesquisa em um

órgão público do poder judiciário e identificou que a insatisfação dos funcionários tende a aumentar com o tempo de serviço e essa insatisfação pode causar conflitos no ambiente de trabalho. De forma semelhante, Affonso e Martins(3) buscou identificar os fatores que geravam insatisfação em uma equipe de saúde municipal, com o objetivo de minimizar o alto índice de absenteísmo que ocorria na entidade, conflitos interpessoais, melhoria de processos de trabalho com um melhor aproveitamento do tempo. A pesquisa realizada gerou uma listagem de sugestões e considerações fornecidas pelos servidores e o autor acreditou que seria possível resgatar o comprometimento dos servidores com a organização de saúde municipal por meio de um conjunto de ações baseada na listagem identificada.

Carvalho, Falce e Guimarães(6) e Carvalho e Silva(4) relacionam motivação com satisfação no serviço público, sendo que Silva e Carvalho apresentaram o conceito de Motivação e as Teorias Motivacionais e concluem que a motivação está relacionada com a satisfação do servidor público no ambiente de trabalho, além de mencionar que os gestores devem ter uma percepção do perfil dos servidores com o objetivo de promover a motivação e satisfação destes. Já Carvalho, Falce e Guimarães(6) realizaram um estudo sobre motivação e satisfação em uma universidade pública e identificaram que baixos níveis de motivação tendem a diminuírem os níveis de satisfação também e, no caso de uma órgão público, podem comprometer a produtividade dos servidores, comprometer os processos da organização e aumentar o número de remoções de servidores.

2.0.3 Aplicação prática da I.A. na gestão de recursos humanos

As aplicações da I.A. na gestão de recursos humanos (RH) são diversas, dentre as quais vale destacar a aplicação no processo de recrutamento/seleção e triagem, a avaliação de desempenho dos empregados, a detecção de *turnover/attrition*, foco da pesquisa. As aplicações normalmente utilizam dados obtidos por meio de *surveys, scraping* em redes sociais, no caso de recrutamento e seleção, bancos de dados de sistemas de recursos humanos e utilizam técnicas de aprendizado de máquina e aprendizado profundo (DL - *Deep Learning*) para apontarem uma situação que pode ser utilizada na tomada de decisão do gestor de RH.

Por serem diversas as aplicações e pesquisas que envolvem a aplicação da I.A. no setor público, foram identificados três trabalhos que merecem destaque, sendo eles: Mehr(31) que aponta em seu estudo áreas do serviço público ao redor do mundo tem se beneficiado com o uso da I.A., e cita exemplo de aplicação no atendimento e requisições de cidadãos, onde os sistemas são capazes de responder questões, preencher e buscar documentos, etc. Nesses casos, ele cita que o uso da I.A. irá liberar os servidores para atender de uma forma melhor os cidadãos, melhorando a relação entre eles. Além disso, o autor destaca cuidados que devem ser tomados para que a implantação da tecnologia ocorra da forma correta, reconhecendo a I.A. como uma boa ferramenta para aumentar a

eficiência dos serviços públicos.

Anastasopoulos e Whitford(32) também utilizou algoritmos de ML para realizar uma pesquisa na administração pública americana sobre as postagens realizadas no Twitter por diversos órgãos e comparou o desempenho obtido com o algoritmo *Gradient Boosted Tree*. A pesquisa identificou também a reputação organizacional dos órgãos públicos por meio das postagens e o autor concluiu que o algoritmo teve bons resultados e cita que o aprendizado profundo irá revolucionar como os governos coletam, processam e interpretam dados.

Já Young, Bullock e Lecy(33) analisou as dificuldades e responsabilidades de implantar a I.A. no ambiente público, bem como os impactos que podem ser causados com a execução exclusiva por uma I.A. O autor traz os termos discricionaridade humana e discricionaridade artificial, esta última gerada por uma I.A. ao analisar o problema e decidir. Ele cita que os sistemas de discricionaridade artificiais tem como vantagem com relação à escalabilidade e custo, mas com relação à qualidade, deve ser feita uma avaliação mais criteriosa. Os autores ofereceram um *framework* de discricionaridade artificial capaz de entender e antecipar a difusão da I.A. no governo e avaliar seus impactos sob várias dimensões.

Considerando que o estudo em questão é a identificação do *turnover* e como a I.A. pode ser utilizada para identificá-lo antes que ocorra, serão apresentados algumas publicações onde os pesquisadores utilizaram algoritmos de aprendizado de máquina para analisar uma base de dados que continha informações que poderiam auxiliar na identificação do *turnover*.

Valle e Ruz(34) fez um estudo com um banco de dados de uma empresa de *call center* onde utilizou os algoritmos de classificação *Naïve Bayes (NB)* e *Random Forest (RF)* para prever o turnover dos funcionários utilizando os dados de um mês de trabalho dos funcionários, com o objetivo de perceber o turnover no segundo mês e com dados de dois meses de trabalho, para prever o turnover no terceiro mês. O autor conseguiu identificar funcionários com baixa produtividade para poder sugerir uma atuação da empresa para evitar o turnover ou mesmo identificar funcionários que não permaneceriam no trabalho.

Já Punnoose e Xavier(29) tinha como objetivo criar modelos de identificação de turnover e, para isso, utilizou dados de funcionários de uma empresa global que representavam dados de 18 meses extraídos de um sistema de recursos humanos, e fez um comparativo de diversos algoritmos de classificação - *Logistic Regression (LR), Naive Bayes(NB), Random Forest(RF), K-Nearest Neighbors (KNN), Linear Discriminant Analysis (LDA), Support Vector Machine (SVM) e Extreme Gradient Boosting (XGBoost)*. Com os resultados obtidos, identificou o *XGBoost* como o melhor dos algoritmos testados, capaz de prever o *turnover* dos funcionários com uma alta acurácia, recomendando seu uso para prever essa situação dos funcionários.

Namrata et al.(35) utilizou o banco de dados *"IBM HR Employee Analytics Attrition and Performance"*, banco de dados disponível na plataforma *Kaggle(20)* para comparar o desempenho de alguns algoritmos de aprendizado de máquina que foram *Decision Tree (DT), SVM, KNN, RF e NB*. O autor mediu o desempenho destes algoritmos usando as métricas *Accuracy*, Matriz de Confusão e curva ROC, apontando também uma comparação do resultado de cada algoritmo em duas situações, onde havia feito a seleção de características e onde não havia feito essa seleção, e apontou que essa etapa é muito importante para o desempenho satisfatório dos algoritmos.

Jhaver, Gupta e Mishra(36) utilizou um banco de dados obtido em um *Data Camp*, atualmente também disponível no *Kaggle(20)*, fez uma análise exploratória dos dados e aplicou os algoritmos de aprendizado de máquina *Logistic Regression (LR), RF, SVM, Gradient Boosting (GB) e Artificial Neural Networks (ANN)*, utilizando acurácia como avaliador de desempenho e teve como melhor resultado o GB. O autor apontou que essa predição pode auxiliar a empresa a manter o empregado por meio de ações que evitem os gastos que envolvem o *turnover*.

Jin Jiaxing Shang e Qiang(14) propôs a detecção de *turnover* utilizando a combinação de RF com análise de sobrevivência, utilizou dados da maior plataforma profissional da China e comparou o resultado do algoritmo proposto por eles com as técnicas clássicas de aprendizado de máquina *NB, LR, DT, XGBoost e RF*. Além disso combinou as técnicas clássicas com o modelo de risco proporcional de COX e também as técnicas clássicas em combinação com o modelo de *Random Survival Forest*. O autor utilizou como métricas acurácia, precisão, recall, F1-score e AUC e o modelo combinado proposto pelo autor teve bons resultados, sendo melhor que os demais algoritmos comparados em quatro das cinco métricas.

Sadana e Munnuru(12) aplicou dois *surveys* em uma empresa de T.I., com um funcionários e outro com ex-funcionários, para levantamento de dados que iria utilizar com os algoritmos de aprendizado de máquina, realizou uma análise destes dados, balanceamento dos dados e aplicou os algoritmos LR, RF, DT e GB como forma de demonstrar um *pipeline* básico de predição de turnover em empresas de T.I.

Shankar et al.(13) fez uma análise utilizando o banco de dados *IBM HR Employee Analytics Attrition and Performance* e fez um comparativo entre as técnicas *Multi-layer Perceptron*, construído conforme proposições do autor, e as técnicas GB, RF e *TabNet*, usando como métricas acurácia, *Log Loss, Jaccard Score e Hamming Loss*, onde o proposto pelo autor se mostrou o melhor resultado.

Além dos trabalhos apresentados, a Tabela 2 apresenta os outros trabalhos que utilizaram a I.A. na gestão de recursos humanos, sendo a grande maioria trabalhando com *turnover/attrition*.

Tabela 2 – Artigos levantados sobre I.A. na gestão de RH

Ref.	Ano	Citações	Problema	Solução
(37)	2019	41	Avaliações de desempenho e contratação	Decision Tree com ID3 e C45 com Naive Bayes
(24)	2021	34	Turnover/Attrition	RF, GB, DNN, Multiple Linear Regression Feature Selection
(23)	2021	34	Turnover/Attrition	DT, Adaboost, RF, LR, GB e Ensemble - DT+LR, AB+RF, SG+GB
(25)	2021	22	Turnover	CART e XGBoost
(38)	2018	16	Promoção	KNN, LR, SVC, DT, RF e Adaboost
(22)	2021	13	Turnover	Deep Learning
(35)	2019	12	Attrition	DT, SVM, KNN, RF, NB
(19)	2018	9	Reposição de mão de obra adequada	Adaboost, DT, GB, KNN, LDA, LR, MLP, NB, RF, SGD, SVM
(36)	2019	8	Turnover	LR, SVM, RF, GB, ANN
(12)	2021	6	Attrition	LR, RF, DT, GB para features
(18)	2021	6	Seleção e alocação	J48, NB, RF, SMO(melhoria do SVM)
(30)	2021	3	Desempenho organizacional percebido	GUIDE
(26)	2021	3	Turnover	SVM, RF, NN, DT, LR
(39)	2022	3	Promoção	DT
(27)	2021	2	Turnover	Clustering e CART
(40)	2022	2	Promoção	SVM, RF, ANN(MLP) com e sem ROS ou SMOTE
(13)	2021	1	Attrition e Desempenho	RF, TABNET,GB e MLP
(41)	2021	1	Turnover	B-LSTM
(11)	2022	1	Turnover	K-means (cluster) e CART, ANN
(42)	2022	0	Turnover	XGBoost e LR
(28)	2022	0	Turnover	DT, GBDT, XGBoost

Como o principal foco da trabalho é determinar uma ferramenta de auxílio ao RH com o uso de I.A., as técnicas serão sucintamente citadas apenas como entendimento de sua capacidade para as análises da metodologia.

Além dos trabalhos informados, devem ser citados os algoritmos de aprendizagem de máquina que foram utilizados na pesquisa. Inicialmente foram testados algoritmos de detecção de anomalias, sendo que foi testado o *Elliptic Envelope*, definido por Rousseeuw e Driessen(43), que é um algoritmo que visa encontrar o centro e a forma de uma nuvem de dados de uma forma que seja resistente a valores discrepantes, usando uma abordagem de divisão e conquista para dividir os dados em subconjuntos menores, que são então analisados para encontrar uma estimativa de *Minimum Covariance Determinant (MCD)*.

Além do *Elliptic Envelope*, ainda foi testado o *One Class SVM* que é um algoritmo de detecção de anomalias que usa uma SVM para aprender um limite de decisão que abrange as instâncias de dados normais. O objetivo é separar os dados normais dos possíveis *outliers* ou anomalias, identificando efetivamente as instâncias que estão longe do limite aprendido. Também foi testado o desempenho do *SGD One Class SVM*, onde ele se difere por utilizar o *Stochastic Gradient Descent*.

O *Isolation Forest*, definido por Liu, Ting e Zhou(44) propõe uma nova abordagem baseada em modelo para detecção de anomalias chamada *iForest*. Este algoritmo isola anomalias em vez de traçar o perfil de pontos normais e possui uma complexidade de tempo linear com uma constante baixa. O *iForest* funciona bem em grandes conjuntos de dados e problemas de alta dimensão, e pode detectar anomalias mesmo em situações onde o conjunto de treinamento não contém nenhuma anomalia. O autor fornece evidências empíricas de que o *iForest* tem um desempenho favorável a outros detectores de anomalias de última geração em termos de tempo de processamento e desempenho.

Já Breunig et al.(45) definiu o *Local Outlier Factor(LOF)*, algoritmo que, ao contrário dos métodos tradicionais de detecção de *outliers*, que rotulam um objeto como discrepante ou não, o LOF atribui um grau de discrepância a cada objeto com base em seu isolamento dentro da vizinhança circundante. Segundo os autores, o algoritmo possui eficiência na localização de valores discrepantes significativos que podem ter sido perdidos por outros métodos que utilizam conjuntos de dados do mundo real. Eles também realizam uma avaliação cuidadosa do desempenho do algoritmo e mostram que ele é prático para encontrar valores discrepantes locais.

Além de técnicas de aprendizado não supervisionado, foram testadas as técnicas de aprendizado supervisionado nas bases de dados de RH:

O *Ridge Classifier* é uma extensão do classificador linear padrão (também conhecido como Regressão Logística para classificação binária), definido por Hoerl e Kennard(46) . Esse algoritmo incorpora um termo de regularização à função de custo do classificador linear padrão. O termo de regularização ajuda a evitar *overfitting* e melhora a generalização do modelo.

O algoritmo de aprendizado supervisionado *Logistic Regression* pode ser usado para classificação e funciona calculando a probabilidade de uma instância pertencer a uma classe, dado um conjunto de características. A classe com a maior probabilidade é então escolhida como a classificação da instância. Para evitar que ocorra *overfitting*, podem ser aplicadas penalidades aos pesos dos modelos.

Já o *Dummy Classifier* faz previsões usando regras muito simples e na verdade não aprende com os dados, sendo utilizado muitas vezes como base para verificar se um classificador mais avançado fornece algum poder preditivo real além do que uma abordagem simples baseada em regras pode alcançar.

A *Linear Discriminant Analysis (LDA)*, envolve encontrar a combinação linear de recursos que melhor separa as classes, mas o foco está em fazer previsões precisas sobre dados novos e invisíveis, em vez de extração de recursos ou redução de dimensionalidade. O algoritmo estima médias específicas de classe e matrizes de covariância e, em seguida, usa essas estimativas para calcular limites de decisão que podem ser usados para classificar novos pontos de dados.

Quadratic Discriminant Analysis (QDA) é um algoritmo de classificação semelhante à LDA, mas difere pois considera que cada classe possui sua própria matriz de covariância. Essa flexibilidade adicional permite que o QDA capture relacionamentos mais complexos entre recursos e classes, tornando-o útil quando as classes têm estruturas de covariância diferentes. No entanto, podem ser necessários uma quantidade maior de dados para estimar as matrizes de covariância com precisão.

Ada Boost Classifier (Adaptive Boosting), de autoria de Freund e Schapire(47) é

uma técnica de conjunto que combina vários modelos simples para criar um classificador robusto. Isso é feito atribuindo pesos mais altos a instâncias classificadas incorretamente, para que os modelos simples subsequentes se concentrem mais em acertar essas instâncias. A previsão final é uma soma ponderada das previsões individuais dos modelos simples.

Já Breiman(48) definiu o *Bagging (Bootstrap Aggregating) Classifier* que é uma abordagem de conjunto onde vários modelos básicos (geralmente árvores de decisão) são treinados em diferentes amostras de *bootstrap* dos dados de treinamento. Suas previsões são então calculadas em média ou votadas pela maioria para produzir a previsão final. O agrupamento reduz o *overfitting* e melhora a estabilidade dos modelos.

O *Gradient Boosting (GB)*, definido por H.(49), é uma técnica de conjunto que constrói modelos sequencialmente. Cada modelo é treinado para corrigir os erros do modelo anterior. A previsão final é a soma das previsões de todos os modelos individuais. O GB é excelente em precisão preditiva, mas pode estar sujeito a *overfitting* se não for ajustado com cuidado.

O *Hist Gradient Boosting Classifier* é semelhante ao *Gradient Boosting*, por construir modelos sequencialmente, no entanto ele usa técnicas baseadas em histograma para discretizar o espaço de recursos, tornando-o mais eficiente em termos de memória e mais rápido para grandes conjuntos de dados.

Bernoulli Naive Bayes e Categorical Naive Bayes são variantes do classificador *Naive Bayes* para dados binários e categóricos, respectivamente. Ambos são baseados no teorema de *Bayes* e na suposição de independência de recursos dada à classe. *Bernoulli NB* é usado para dados binários, enquanto *Categorical NB* é usado para dados categóricos com mais de duas categorias e é particularmente útil para classificação de texto onde os recursos representam contagens de palavras ou presençaausência de certas palavras.

Complement Naive Bayes foi projetado para combater o problema de distribuições de classes desequilibradas, considerando o "complemento" da distribuição de classes e é especialmente eficaz ao lidar com conjuntos de dados desbalanceados, onde uma classe é mais prevalente. O *Complement NB* ajusta as probabilidades dos recursos para a classe complementar, o que ajuda a melhorar o desempenho da classificação em situações de desbalanceamento.

Multinomial Naive Bayes é usado para dados discretos com múltiplas categorias, como contagens de palavras em dados de texto, pois estima as probabilidades de ocorrência de recursos dentro de cada classe e usa o teorema de *Bayes* para fazer previsões. Entre suas aplicações, é frequentemente usado em tarefas de classificação de texto, como detecção de spam ou análise de sentimentos.

Gaussian Naive Bayes assume que os recursos seguem uma distribuição gaussiana (normal) dentro de cada classe. É adequado para recursos numéricos contínuos e funciona

bem quando a distribuição dos dados dentro de cada classe é aproximadamente gaussiana. Apesar de sua suposição "ingênua" (naive) de independência de recursos, o *Gaussian NB* pode ter um desempenho surpreendentemente bom na prática.

K-Nearest Neighbors atribui um rótulo de classe a um ponto de dados com base nos rótulos de classe majoritários de seus k-vizinhos mais próximos nos dados de treinamento, sendo que a escolha de k afeta a suavidade do limite de decisão. Valores menores de k levam a limites mais complexos e potencialmente ruidosos. O algoritmo *KNeighborsClassifier* é baseado em instâncias e não paramétrico, o que significa que não faz suposições fortes sobre a distribuição de dados.

Já o *RadiusNeighbors* é semelhante a *K-Nearest Neighbors*, mas atribui rótulos de classe com base em um raio fixo ao redor do ponto de dados, em vez de um número fixo de vizinhos. É útil quando a densidade dos pontos de dados varia no espaço de recursos, pois se adapta à densidade de dados local.

O *Nearest Centroid* classifica os pontos de dados com base no centroide de classe mais próximo (média) no espaço de recursos e é um algoritmo de classificação simples que assume que cada classe possui um centroide representativo. Funciona bem quando as classes são separáveis por centroides, mas pode ter dificuldades quando as classes têm formas complexas.

O algoritmo *Multi-Layer Perceptron (MLP)*, definido por Hinton(50), atualmente é uma rede neural com múltiplas camadas, incluindo camadas de entrada, ocultas e de saída. É capaz de aprender relações complexas em dados ajustando os pesos e tendências de seus neurônios por meio de retropropagação. O *MLPClassifier* é adequado para uma ampla gama de tarefas, mas seu desempenho depende muito da arquitetura e do ajuste de hiperparâmetros.

As técnicas *Linear SVC, Nu SVC, SVC*, que possuem referências em C.(51) e Chang e Lin(52), são variantes de *Support Vector Machine (SVM)* para classificação, onde o *Linear SVC* usa um kernel linear e é eficaz para dados linearmente separáveis, o *Nu SVC* permite controlar o número de vetores de suporte com o parâmetro “nu” e o SVC é um SVM geral com diferentes opções de *kernel* (linear, polinomial, função de base radial, etc.). Todas as técnicas visam encontrar o hiperplano que melhor separa as classes, maximizando a margem entre os vetores de suporte.

A *Decision Tree (DT)*, definida por Breiman et al.(53), é um algoritmo de aprendizado de máquina supervisionado usado para tarefas de classificação e regressão. Ele divide os dados em subconjuntos com base em recursos, formando uma estrutura semelhante a uma árvore, onde seleciona recursos para dividir dados, cria nós para cada divisão e atribui rótulos ou valores a nós folha. A divisão recursiva continua até que um critério de parada, como profundidade ou amostras mínimas, seja atendido. Essa estrutura permite

a previsão percorrendo a árvore da raiz à folha, utilizando valores de recursos para caminhos de decisão. As árvores de decisão são interpretáveis e lidam com vários tipos de recursos, mas pode ocorrer *overfitting*. Para evitar este problema, técnicas como métodos de poda e *ensemble* podem ser utilizadas.

Também definido por Breiman(54), o *Random Forest Classifier* é um algoritmo conjunto que cria múltiplas árvores de decisão durante o treinamento e combina suas previsões por meio de votação ou média. Reduz o *overfitting* e fornece bom desempenho. Cada árvore de decisão é treinada em um subconjunto inicializado de dados e toma decisões de forma independente.

O *Extra Trees Classifier* é uma extensão do *Random Forest*. Ele introduz aleatoriedade extra usando limites aleatórios para cada recurso ao dividir nós em árvores de decisão. Essa aleatoriedade adicional torna o *ExtraTrees* mais robusto para dados ruidosos e pode levar a uma melhor generalização.

Chen e Guestrin(55) definiu *Extreme Gradient Boosting(XGBoost)* que é uma estrutura de aumento de gradiente que usa árvores de decisão como modelos básicos.O *XGBRFClassifier* é uma variante que usa *Random Forests* como modelos base. O *XGBoost* constrói modelos sequencialmente, minimizando a função de perda ao adicionar árvores de decisão que corrigem os erros das árvores anteriores.

Light GBM, definido em Ke et al.(56) é uma estrutura de aumento de gradiente com técnicas eficientes de treinamento baseadas em histograma. É conhecido por sua velocidade, escalabilidade e desempenho em grandes conjuntos de dados. *LightGBM* é amplamente utilizado para uma variedade de tarefas devido à sua eficiência e forte capacidade preditiva. Já o *DaskLGBMClassifier* é uma implementação do classificador *LightGBM* compatível com *Dask*, permitindo computação distribuída. É útil para lidar com eficiência com grandes conjuntos de dados que não cabem na memória.

3 Metodologia

A pesquisa foi iniciada com a seleção dos bancos de dados, com as bases selecionadas foi feita a fatoração dos campos categóricos e então foram feitas as análises iniciais dos dados presentes nas bases. Após esse processo, foram executados algoritmos de detecção de anomalias para verificar se era possível a identificação dos dados sem uma classe tutora e os resultados não foram tão interessantes.

Durante a análise dos dados, foi possível identificar que os dados estavam desbalanceados e com o objetivo de obter o resultado correto das técnicas de aprendizado de máquina foi feito o balanceamento utilizando o SMOTE, proposto por Chawla et al.(57), foi utilizado como técnica de *oversampling* para as bases de dados por ser considerado o mais popular para o balanceamento de dados e também por ter sido utilizado por Sadana e Munnuru(12), Sahinbas(40) e Namrata et al.(35).

Com os dados balanceados, foi feita a classificação das bases utilizando os algoritmos de classificação clássicos e depois foram feitas a utilização de duas técnicas de classificação do módulo ensemble do *Scikit-Learn* - o *StackingClassifier* (58) e o *VotingClassifier*.

Além das técnicas clássicas e os métodos ensemble também foi utilizado o *Tree-based Pipeline Optimization Tool (TPOT)*, proposto por Olson, Edu e Moore(59), que constrói um pipeline de operadores otimizados utilizando algoritmos genéticos.

Os resultados obtidos dos algoritmos foram medidos com as métricas acurácia balanceada, precisão, recuperação e f1. A visão geral do fluxo da metodologia pode ser visto Figura 1.

As técnicas que apresentarem os resultados mais promissores serão posteriormente aplicadas no contexto da base de dados do sistema integrado de gestão da universidade. Esta base abrange uma variedade de dados, incluindo informações de frequência, registros de ponto, históricos de acessos, emissões de documentos e outros dados relevantes sobre os servidores técnico-administrativos da instituição. Além disso, o setor de gestão de pessoas possui dados que podem ser combinados com os dados de sistema para gerar uma nova base de dados.

A riqueza de dados contida nesse conjunto proporciona uma oportunidade única para aprimorar a eficiência da gestão de recursos humanos. Dentre as características dessa base de dados, destaca-se a inclusão de informações relacionadas ao desligamento de servidores. A existência desses dados possibilita a construção de classificadores capazes de identificar perfis de servidores que podem estar enfrentando insatisfações no ambiente

de trabalho, que pode gerar uma situação de *turnover/attrition* por parte do servidor. Embora sejam ambientes diferentes, os dados dos funcionários são comuns muitas vezes, o que pode auxiliar na utilização das ferramentas de aprendizado de máquina para definir um perfil.

Ao treinar tais classificadores, será possível identificar padrões e sinais de alerta precoce associados à insatisfação dos servidores. Este processo não apenas permite a detecção proativa de potenciais problemas de satisfação, mas também oferece uma abordagem preventiva para abordar essas questões antes que se tornem mais complexas. Enquanto a insatisfação no setor privado leva à situação de *turnover/attrition*, no setor público, além da possibilidade de ocorrer tal situação, o servidor pode ficar desmotivado e prejudicar a prestação do órgão público.

Uma vez que os perfis de servidores insatisfeitos sejam identificados, a equipe de gestão de pessoas estará capacitada a intervir de maneira proativa. Isso envolverá uma abordagem personalizada para cada servidor, onde a equipe poderá verificar, de maneira colaborativa, se a situação identificada pela técnica de classificação reflete precisamente a realidade do servidor. Além disso, será possível oferecer soluções e oportunidades personalizadas que visam aumentar a satisfação do servidor dentro do ambiente de trabalho, gerando oportunidades para o servidor que possam trazer mais satisfação.

Essa estratégia integrada, unindo técnicas avançadas de análise de dados com uma abordagem proativa da equipe de gestão de pessoas, representa um passo significativo em direção à melhoria contínua do ambiente laboral. Ao compreender e antecipar as necessidades dos servidores, a universidade demonstra um compromisso com a promoção do bem-estar e satisfação de sua equipe, contribuindo, assim, para um ambiente de trabalho mais saudável e produtivo.

Toda a metodologia foi feita utilizando a linguagem de programação *python* em um *Jupyter Notebook* e a ferramenta gerada está disponível em Dias e Moraes(60). O repositório citado possui a ferramenta de detecção de *turnover/attrition*, esta dissertação de mestrado, o artigo publicado durante o programa e a imagem apresentada com a metodologia proposta.

3.1 Análises das Bases de Dados

As bases de dados foram buscadas no site Kaggle(20), na parte de *datasets*. A busca foi feita por dados que tinham algum foco em gestão de recursos humanos e possíveis aplicações da I.A.. Dentre as diversas bases de diferentes temas, foi identificado o tema de *turnover/attrition* que seria propício para ser utilizado, pela possibilidade de aplicação em uma instituição pública para identificar insatisfação de servidores e com isso auxiliar a gestão na atuação junto aos insatisfeitos. Diante disso foram selecionados quatro bases

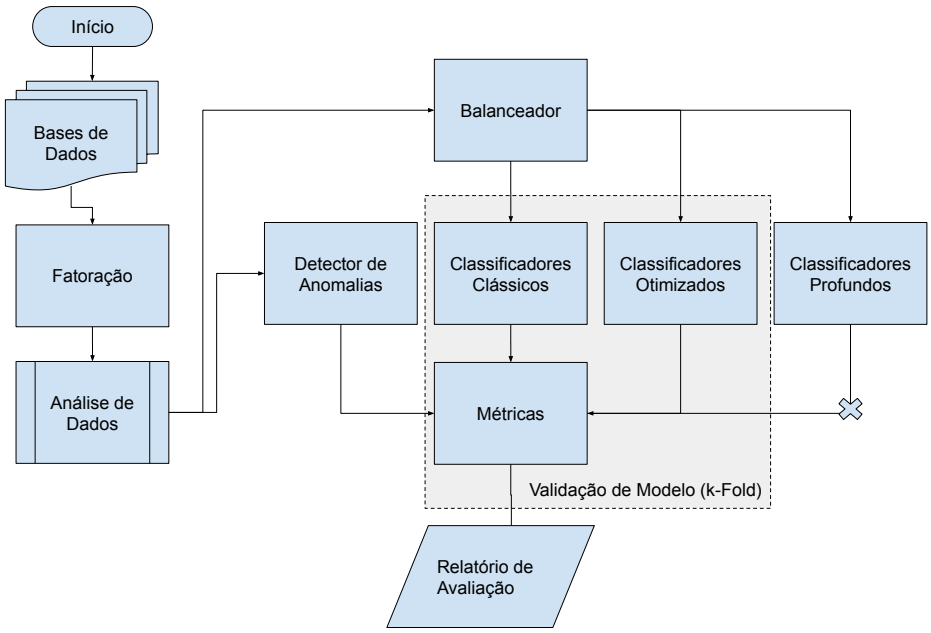

Figura 1 – Fluxograma de execução da metodologia

de dados sobre o tema.

Para cada uma das bases de dados foi realizada a fatoração dos atributos texto utilizando a função *factorize* do pandas como forma de pré-processamento dos dados. Após esta etapa, a análise de dados foi feita com a matriz de correlação para cada base de dados, para que fosse possível identificar a relação das características de cada uma delas, depois foram selecionados as características mais significativas com o *SelectKbest* do *scikit-learn* e foi feito um histograma para as características selecionadas pelo algoritmo.

Além das análises feitas, foi feita a redução de dimensionalidade das bases utilizando as duas técnicas mais populares, o *Principal Component Analysis (PCA)* que é uma técnica linear que faz uma redução de dimensionalidade que preserva a máxima variância nos dados e o *t-Distributed Stochastic Neighbor Embedding (t-SNE)* que é uma outra técnica de redução de dimensionalidade, mas diferente do PCA, é uma técnica não-linear.

Cada uma das bases de dados e seus respectivos gráficos serão apresentados nas subseções específicas.

3.1.1 HR Comma Sep

Para a análise inicial, foi utilizada a base de dados denominada HR Comma Sep (61), divulgada em 2017 pelo usuário do Kaggle(20) denominado Kakisama. Esta base é composta por 10 colunas e abrange um total de 14.999 registros.

As características da base de dados em questão são apresentadas na lista a seguir, onde as características numéricas(real e inteiro) possuem informação quanto ao valor

mínimo, valor máximo e distribuição. Para determinar a distribuição foi utilizada a biblioteca *fitter* (62) do python para determinar a distribuição com base nas distribuições mais comuns apresentadas pela biblioteca, que são: *'cauchy', 'chi2', 'expon', 'exponpow', 'gamma', 'lognorm', 'norm', 'powerlaw', 'rayleigh' e 'uniform'.*

1. *satisfaction_level*
    - descrição: nível de satisfação do funcionário
    - tipo: real
    - valor mínimo: 0.09
    - valor máximo: 1.00
    - média: 0.61
    - distribuição: *exponpow*
2. *last_evaluation*
    - descrição: pontuação obtida na última avaliação
    - tipo: real
    - valor mínimo: 0.36
    - valor máximo: 1.00
    - média: 0.71
    - distribuição: *powerlaw*
3. *number_project*
    - descrição: número de projetos do funcionário
    - tipo: inteiro
    - valor mínimo: 2
    - valor máximo: 7
    - média: 3.80
    - distribuição: *powerlaw*
4. *average_montly_hours*
    - descrição: média de horas mensais trabalhadas
    - tipo: inteiro
    - valor mínimo: 96
    - valor máximo: 310

- média: 201.05
- distribuição: *rayleigh*

5. *time_spend_company*
    - descrição: tempo gasto na empresa
    - tipo: inteiro
    - valor mínimo: 2
    - valor máximo: 10
    - média: 3.49
    - distribuição: *powerlaw*
6. *work_accident*
    - descrição: registro de acidente com o funcionário
    - tipo: categorizado
    - categoria 1: 0 - não houve acidente
    - categoria 2: 1 - houve acidente
7. *class*
    - descrição: registro se o funcionário deixou a empresa
    - tipo: categorizado
    - categoria 1: 0 - não deixou a empresa
    - categoria 2: 1 - deixou a empresa
8. *promotion_last_5years*
    - descrição: registro se o funcionário teve promoção nos últimos cinco anos
    - tipo: categorizado
    - categoria 1: 0 - não houve promoção
    - categoria 2: 1 - houve promoção
9. *department*
    - descrição: departamento do funcionário
    - tipo: texto
10. *salary*
    - descrição: salário do funcionário

- tipo: categorizado
- categoria 1: 1 - *low*
- categoria 2: 2 - *medium*
- categoria 3: 3 - *high*

Com o intuito de visualizar a inter-relação entre as características, o mapa de correlação é apresentado na figura 2. Adicionalmente, pode ser verificado um panorama visual das características mais relevantes por meio do histograma, disponibilizado na figura 3 e como pode ser visto, as características mais revelantes identificadas pelo algoritmo *SelectKBest* foram *satisfaction_level, average_montly_hours, time_spend_company, work_accident, promotion_last_5years*, além do tutor.

Esta seleção estratégica da base de dados visa proporcionar uma compreensão inicial da estrutura subjacente e das características distintivas presentes nos dados. A análise da correlação destaca possíveis relações entre variáveis, enquanto o histograma oferece uma representação gráfica das distribuições das principais características. Essas ferramentas visuais fornecem uma base sólida para orientar futuras etapas de exploração e modelagem dos dados, preparando o terreno para análises mais aprofundadas e conclusões significativas.

3.1.2 IBM Employee Attrition

A base de dados IBM Employee Attrition (63) constitui-se de um conjunto fictício de dados desenvolvido por cientistas de dados da IBM, sendo divulgado na plataforma Kaggle(20) em 2016 pelo usuário Pavansubhash. Composta por 1470 registros e 35 colunas, essa base foi concebida para explorar e identificar potenciais causas de atrito (*attrition*) no ambiente de trabalho.

De forma semelhante à apresentada na seção anterior, as características da base de dados IBM são apresentadas na lista a seguir, onde as características numéricas(real e inteiro) possuem informação quanto ao valor mínimo, valor máximo e distribuição. Para determinar a distribuição foi utilizada a biblioteca *fitter* (62) do python para determinar a distribuição com base nas distribuições mais comuns apresentadas pela biblioteca, que são: *'cauchy', 'chi2', 'expon', 'exponpow', 'gamma', 'lognorm', 'norm', 'powerlaw', 'rayleigh' e 'uniform'*.

1. *Age*
   - descrição: idade do funcionário
   - tipo: inteiro
   - valor mínimo: 18

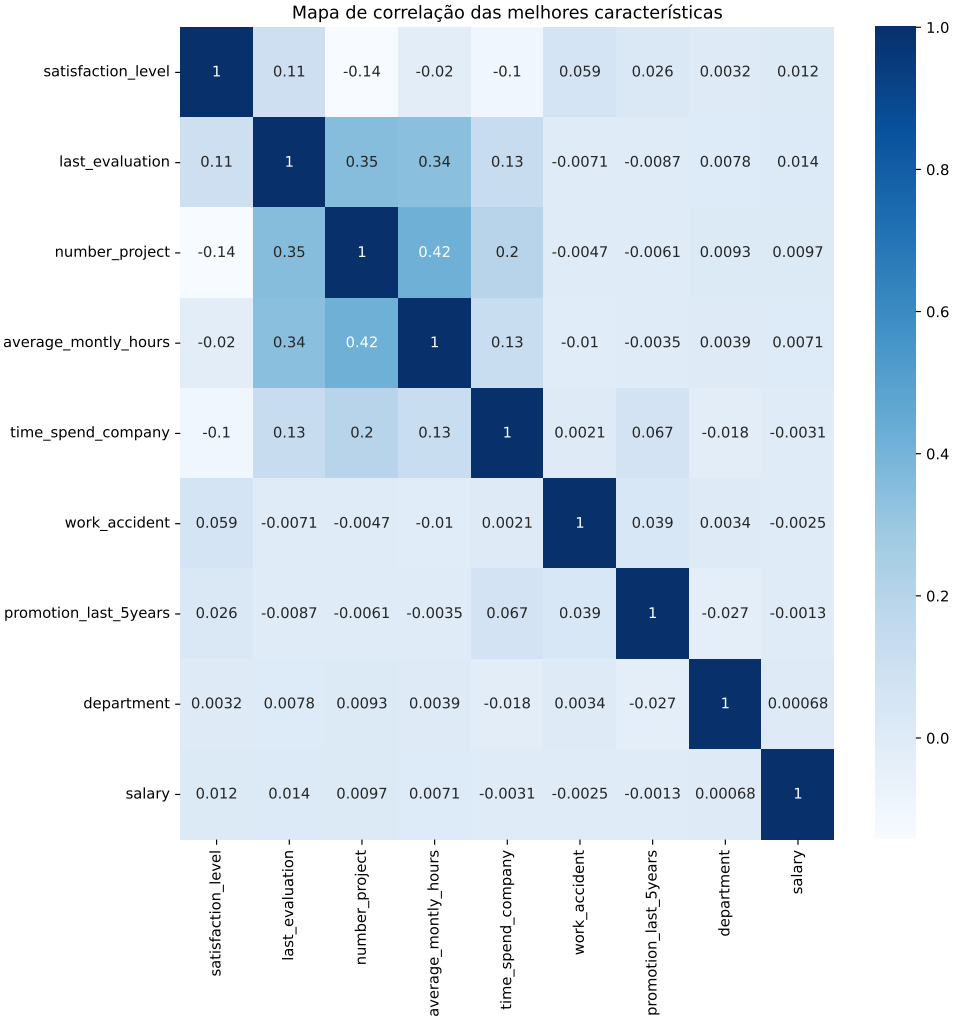

Figura 2 – Mapa de correlação das características da base de dados HR Comma Sep

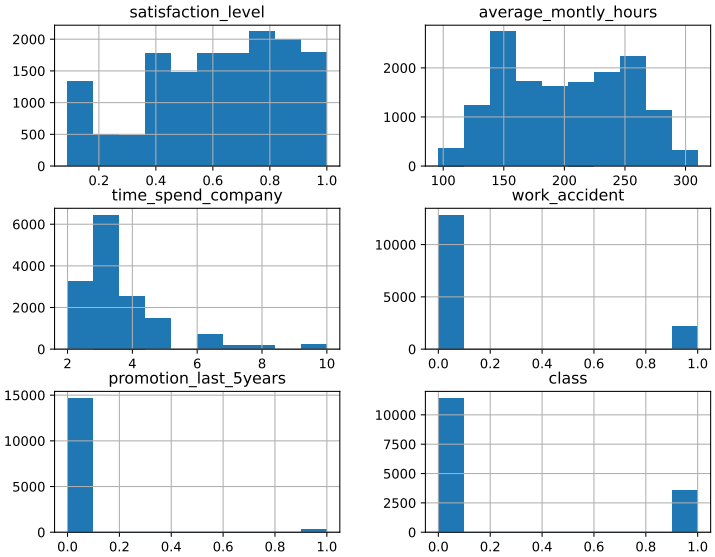

Figura 3 – Histograma das melhores características da base de dados HR Comma Sep

- valor máximo: 60

- média: 36.92
- distribuição: *gamma*

2. *class*
    - descrição: registra se um funcionário deixou a empresa ou não
    - tipo: categorizado
    - categoria 1: Yes
    - categoria 2: No
3. *BusinessTravel*
    - descrição: quantidade de viagens a negócios
    - tipo: categorizado
    - categoria 1: *No Travel*
    - categoria 2: *Travel Frequently*
    - categoria 3: *Travel Rarely*
4. *DailyRate*
    - descrição: nível de salário
    - tipo: inteiro
    - valor mínimo: 102
    - valor máximo: 1499
    - média: 802.48
    - distribuição: *uniform*
5. *Department*
    - descrição: departamento do funcionário
    - tipo: categorizado
    - categoria 1: HR
    - categoria 2: R&D
    - categoria 3: Sales
6. *DistanceFromHome*
    - descrição: distância da casa ao trabalho
    - tipo: inteiro

- valor mínimo: 1
- valor máximo: 29
- média: 9.19
- distribuição: *powerlaw*

7. *Education*
    - descrição: nível de educação
    - tipo: categorizado
    - categoria 1: *Below College*
    - categoria 2: *College*
    - categoria 3: *Bachelor*
    - categoria 4: *Master*
    - categoria 5: *Doctor*
8. *EducationField*
    - descrição: Campo de formação educacional do funcionário
    - tipo: categorizado
    - categoria 1: HR
    - categoria 2: *Life Sciences*
    - categoria 3: Marketing
    - categoria 4: *Medical Sciences*
    - categoria 5: *Other*
    - categoria 6: *Technical*
9. *EmployeeCount*
    - descrição: contagem de registro do funcionário na empresa
    - tipo: inteiro
    - valor mínimo: 1
    - valor máximo: 1
10. *EmployeeNumber*
    - descrição: identificação do funcionário
    - tipo: inteiro
    - valor mínimo: 1

- valor máximo: 2068
- média: 1024.86
- distribuição: *powerlaw*

11. *EnvironmentSatisfaction*

- descrição: satisfação com o ambiente
- tipo: categorizado
- categoria 1: 1 - *Low*
- categoria 2: 2 - *Medium*
- categoria 3: 3 - *High*
- categoria 4: 4 - *Very High*

12. *Gender*

- descrição: gênero do funcionário
- tipo: categorizado
- categoria 1: *Female*
- categoria 2: *Male*

13. *HourlyRate*

- descrição: salário por hora
- tipo: inteiro
- valor mínimo: 30
- valor máximo: 100
- média: 65.89
- distribuição: *uniform*

14. *JobInvolvement*

- descrição: envolvimento com o trabalho
- tipo: categorizado
- categoria 1: 1 - *Low*
- categoria 2: 2 - *Medium*
- categoria 3: 3 - *High*
- categoria 4: 4 - *Very High*

15. *JobLevel*

- descrição: nível do trabalho
- tipo: inteiro
- valor mínimo: 1
- valor máximo: 5
- média: 2.06
- distribuição: *gamma*

16. *JobRole*

- descrição: cargo
- tipo: categorizado
- categoria 1: *HC Rep*
- categoria 2: *HR*
- categoria 3: *Lab Technician*
- categoria 4: *Manager*
- categoria 5: *Managing Director*
- categoria 6: *Research Director*
- categoria 7: *Research Scientist*
- categoria 8: *Sales Executive*
- categoria 9: *Sales Representative*
- distribuição:

17. *JobSatisfaction*

- descrição: satisfação com o trabalho
- tipo: categorizado
- categoria 1: 1 - *Low*
- categoria 2: 2 - *Medium*
- categoria 3: 3 - *High*
- categoria 4: 4 - *Very High*

18. *MaritalStatus*

- descrição: estado civil
- tipo: categorizado
- categoria 1: *Single*

- categoria 2: *Married*
- categoria 3: *Divorced*

19. *MonthlyIncome*

- descrição: salário mensal
- tipo: inteiro
- valor mínimo: 1009
- valor máximo: 19999
- média: 6502.93
- distribuição: *lognorm*

20. *MonthlyRate*

- descrição: ganho mensal
- tipo: inteiro
- valor mínimo: 2094
- valor máximo: 26999
- média: 14313.10
- distribuição: *uniform*

21. *NumCompaniesWorked*

- descrição: número de empresas que o funcionário trabalhou
- tipo: inteiro
- valor mínimo: 0
- valor máximo: 9
- média: 2.69
- distribuição: *exponpow*

22. *Over18*

- descrição: maior de 18
- tipo: categorizado
- categoria 1: *Yes*
- categoria 2: *No*

23. *Overtime*

- descrição: realização de hora extra
- tipo: categorizado
- categoria 1: *Yes*
- categoria 2: *No*

24. *PercentSalaryHike*

- descrição: porcentagem de aumento no salário do funcionário
- tipo: inteiro
- valor mínimo: 11
- valor máximo: 25
- média: 15.20
- distribuição: *powerlaw*

25. *PerformanceRating*

- descrição: classificação de desempenho do funcionário
- tipo: categorizado
- categoria 1: 1 - *Low*
- categoria 2: 2 - *Good*
- categoria 3: 3 - *Excellent*
- categoria 4: 4 - *Outstanding*

26. *RelationshipSatisfaction*

- descrição: satisfação com relacionamentos
- tipo: categorizado
- categoria 1: 1 - *Low*
- categoria 2: 2 - *Medium*
- categoria 3: 3 - *High*
- categoria 4: 4 - *Very High*

27. *StandardHours*

- descrição: quantidade de horas de trabalho padrão
- tipo: inteiro
- valor mínimo: 80
- valor máximo: 80

28. *StockOptionLevel*
    - descrição: nível de opção de ação
    - tipo: inteiro
    - valor mínimo: 0
    - valor máximo: 3
    - média: 0.79
    - distribuição: *gamma*
29. *TotalWorkingYears*
    - descrição: quantidade total de anos de trabalho
    - tipo: inteiro
    - valor mínimo: 0
    - valor máximo: 40
    - média: 11.27
    - distribuição: *cauchy*
30. *TrainingTimesLastYear*
    - descrição: quantidade de treinamentos no último ano
    - tipo: inteiro
    - valor mínimo: 0
    - valor máximo: 6
    - média: 2.79
    - distribuição: *chi2*
31. *WorkLifeBalance*
    - descrição: equilíbrio entre vida pessoal e profissional
    - tipo: categorizado
    - categoria 1: 1 - *Bad*
    - categoria 2: 2 - *Good*
    - categoria 3: 3 - *Better*
    - categoria 4: 4 - *Best*
32. *YearsAtCompany*
    - descrição: quantidade de anos na empresa

- tipo: inteiro
- valor mínimo: 0
- valor máximo: 40
- média: 7.00
- distribuição: *gamma*

33. *YearsInCurrentRole*

- descrição: quantidade de anos no cargo atual
- tipo: inteiro
- valor mínimo: 0
- valor máximo: 18
- média: 4.22
- distribuição: *exponpow*

34. *YearsSinceLastPromotion*

- descrição: quantidade de anos desde a última promoção
- tipo: inteiro
- valor mínimo: 0
- valor máximo: 15
- média: 2.18
- distribuição: *chi2*

35. *YearsWitCurrManager*

- descrição: quantidade de anos com o gerente atual
- tipo: inteiro
- valor mínimo: 0
- valor máximo: 17
- média: 4.12
- distribuição: *exponpow*

Para fornecer uma visão inicial da estrutura e das características essenciais dos dados, são apresentados o mapa de correlação na figura 4 e o histograma das características mais importantes na figura 5, como pode ser visto, nessa base de dados, a execução do algoritmo *SelectKBest* identificou as características *JobLevel, MaritalStatus, Overtime,*

TotalWorkingYears e YearsInCurrentRole, além do tutor, como características importantes para a ocorrência de *turnover*. É possível perceber que o algoritmo não identificou os campos relacionados à satisfação não foram identificados pelo algoritmo como algo impactante para esta base de dados.

Essa escolha estratégica de base de dados visa não apenas explorar as nuances específicas do atrito no ambiente de trabalho, mas também proporcionar uma compreensão inicial das variáveis fundamentais que podem influenciar esse fenômeno. As representações visuais oferecidas por meio do mapa de correlação e do histograma constituem ferramentas essenciais para identificar padrões iniciais e orientar a análise subsequente de maneira mais aprofundada.

3.1.3 Turnover

A base de dados "Employee Turnover" (64) é uma fonte real, conforme atestado pelo autor, composta por 1129 registros distribuídos em 16 colunas. Destaca-se por ser a única base na pesquisa em que o número de funcionários que deixaram o emprego supera aqueles que permaneceram. Esta base foi divulgada pelo usuário Davin Wijaya no ano de 2020.

As características da base de dados *Turnover* constam na listagem abaixo. Embora seja uma base de dados real, possui algumas características específicas da Rússia, país onde ocorreu o levantamento dos dados. Além disso, apresentou a pontuação dos funcionários utilizando as características do teste de personalidade *Big 5 Personality Test*. Para determinar a distribuição dos campos numéricos foi utilizada a biblioteca *fitter* (62) do python para determinar a distribuição com base nas distribuições mais comuns apresentadas pela biblioteca, que são: *'cauchy', 'chi2', 'expon', 'exponpow', 'gamma', 'lognorm', 'norm', 'powerlaw', 'rayleigh' e 'uniform'*.

1. *stag*
    - descrição: tempo de experiência
    - tipo: real
    - valor mínimo: 0.39
    - valor máximo: 179.44
    - média: 36.62
    - distribuição: *gamma*
2. *class*
    - descrição: registro se o funcionário deixou a empresa

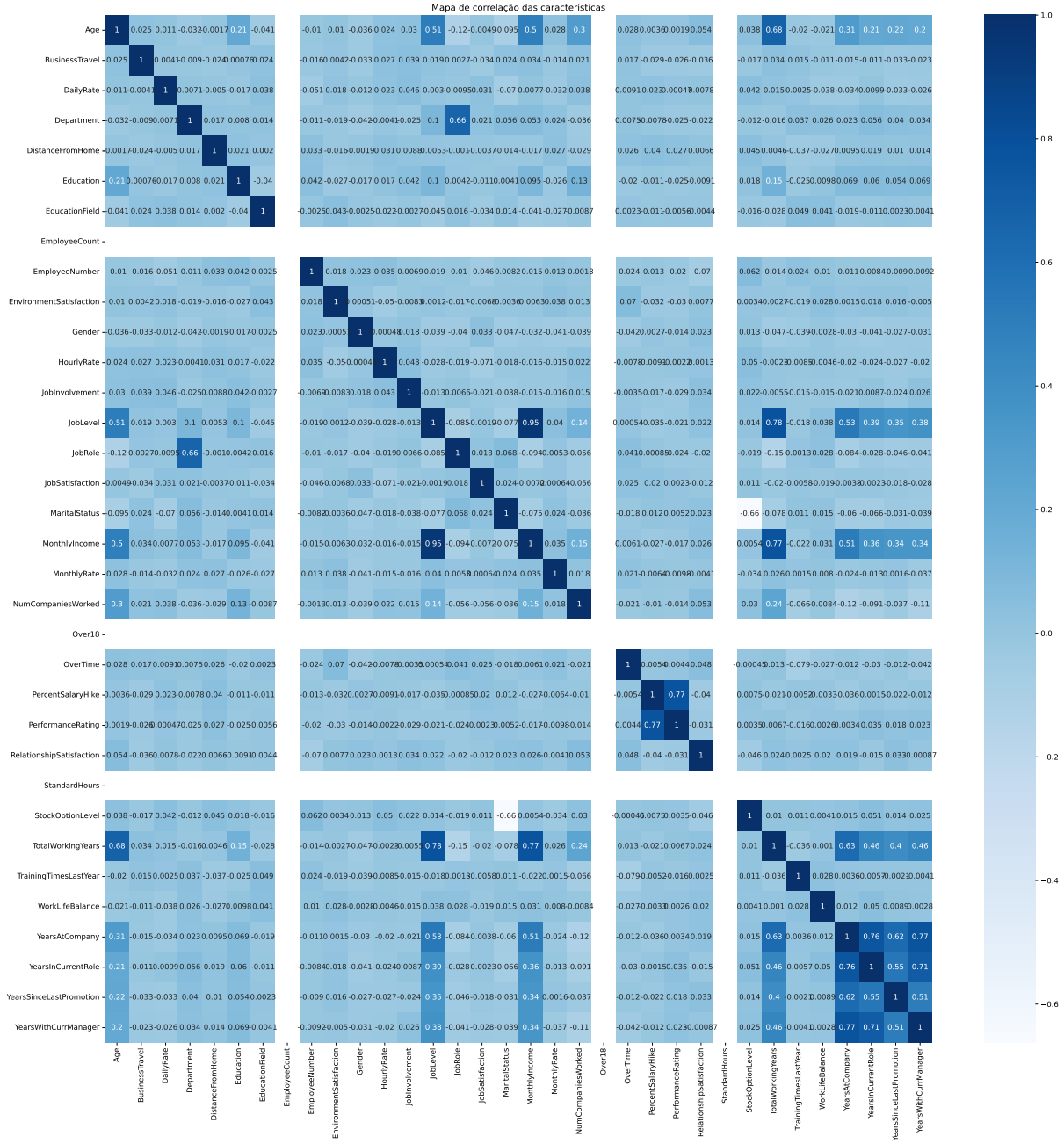

Figura 4 – Mapa de correlação das características da base de dados IBM

- tipo: categorizado
- categoria 1: 0 - não deixou a empresa
- categoria 2: 1 - deixou a empresa

3. *gender*

- descrição: gênero do funcionário
- tipo: categorizado
- categoria 1: m
- categoria 2: f

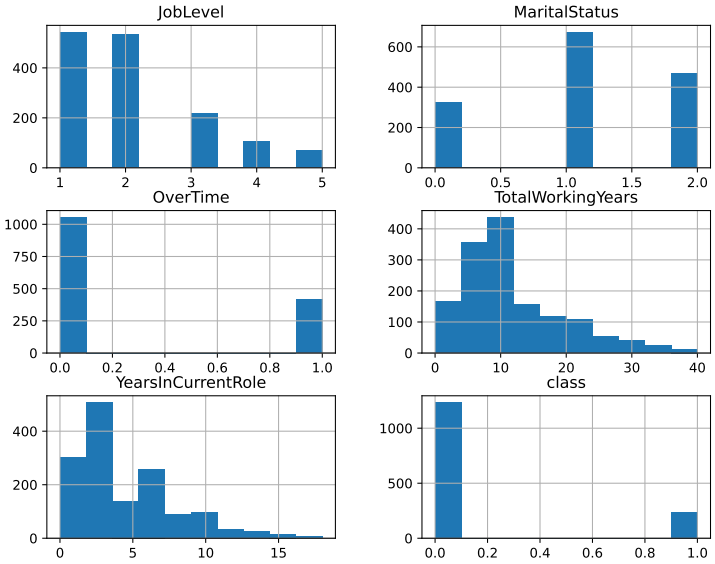

Figura 5 – Histograma das melhores características da base de dados IBM

4. *age*
    - descrição: idade do funcionário
    - tipo: real
    - valor mínimo: 18.00
    - valor máximo: 58.00
    - média: 31.06
    - distribuição: *gamma*
5. *industry*
    - descrição: campo de atuação da empresa do funcionário
    - tipo: texto
6. *profession*
    - descrição: profissão do funcionário
    - tipo: categorizado
    - categoria 1: HR
    - categoria 2: *Commercial*
    - categoria 3: Marketing
    - categoria 4: etc
    - categoria 5: *Sales*
    - categoria 6: *BusinessDevelopment*

- categoria 7: *Finanñe*
- categoria 8: *Teaching*
- categoria 9: *manage*
- categoria 10: *IT*
- categoria 11: *Law*
- categoria 12: *Consult*
- categoria 13: *Engineer*
- categoria 14: PR
- categoria 15: *Accounting*

7. *traffic*
    - descrição: forma de ingresso na empresa
    - tipo: categorizado
    - categoria 1: rabreNErab
    - categoria 2: empjs
    - categoria 3: youjs
    - categoria 4: referal
    - categoria 5: advert
    - categoria 6: KA
    - categoria 7: recNErab
    - categoria 8: friends
8. *coach*
    - descrição: presença de um treinador no período de teste
    - tipo: categorizado
    - categoria 1: 1 - *no*
    - categoria 2: 2 - *yes*
    - categoria 3: 3 - *my head*
9. *head_gender*
    - descrição: gênero do gerente do funcionário
    - tipo: categorizado
    - categoria 1: f

- categoria 2: m

10. *greywage*
    - descrição: o salário não é apresentado ao fisco
    - tipo: categorizado
    - categoria 1: *white*
    - categoria 2: *grey*
11. *way*
    - descrição: forma que o funcionário vai ao trabalho
    - tipo: categorizado
    - categoria 1: *bus*
    - categoria 2: *car*
    - categoria 2: *foot*
12. *extraversion*
    - descrição: pontuação nesse critério do *Big 5 Personality Test*
    - tipo: real
    - valor mínimo: 1.00
    - valor máximo: 10.00
    - média: 5.59
    - distribuição: *lognorm*
13. *independ*
    - descrição: pontuação nesse critério do *Big 5 Personality Test*
    - tipo: real
    - valor mínimo: 1.00
    - valor máximo: 10.00
    - média: 5.47
    - distribuição: *norm*
14. *selfcontrol*
    - descrição: pontuação nesse critério do *Big 5 Personality Test*
    - tipo: real

- valor mínimo: 1.00
- valor máximo: 10.00
- média: 5.59
- distribuição: *exponpow*

15. *anxiety*
    - descrição: pontuação nesse critério do *Big 5 Personality Test*
    - tipo: real
    - valor mínimo: 1.00
    - valor máximo: 10.00
    - média: 5.66
    - distribuição: *chi2*
16. *novator*
    - descrição: pontuação nesse critério do *Big 5 Personality Test*
    - tipo: real
    - valor mínimo: 1.00
    - valor máximo: 10.00
    - média: 5.87
    - distribuição: *exponpow*

A visualização da estrutura dos dados é facilitada através do mapa de correlação, apresentado na figura 6. Além disso, a execução do algoritmo *SelectKBest* resultou nas características *industry, coach, way, independ, anxiety*, além do tutor, como sendo as mais importantes para os casos de turnover. Estas características estão representadas no histograma da figura 7, essa visualização proporciona uma representação gráfica que demonstra as distribuições presentes nessas características.

Essa base singular, ao priorizar eventos de turnover, oferece uma oportunidade única de análise para identificar padrões e fatores subjacentes ao desligamento de colaboradores. As representações visuais fornecem uma entrada valiosa para investigações mais detalhadas, permitindo uma compreensão aprofundada das dinâmicas envolvidas e, consequentemente, facilitando a tomada de decisões informadas em ambientes corporativos.

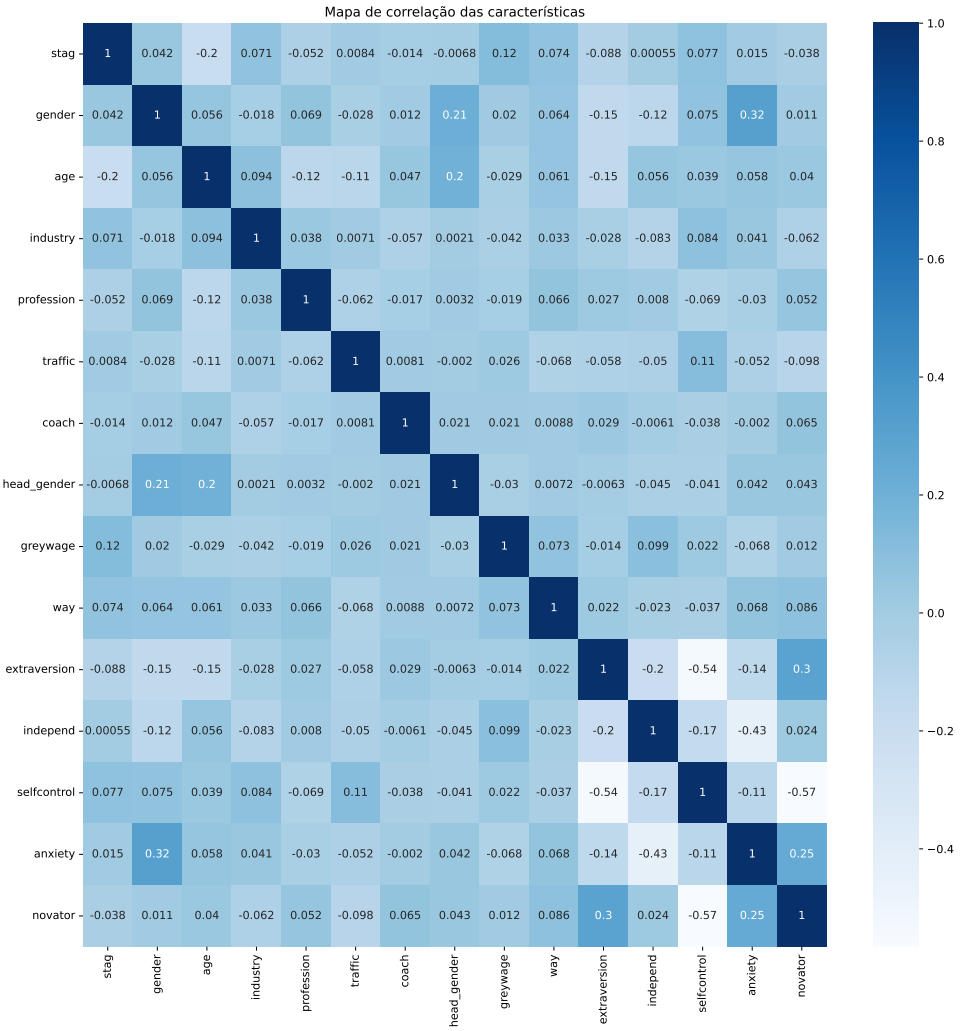

Figura 6 – Mapa de correlação das características da base de dados de Turnover

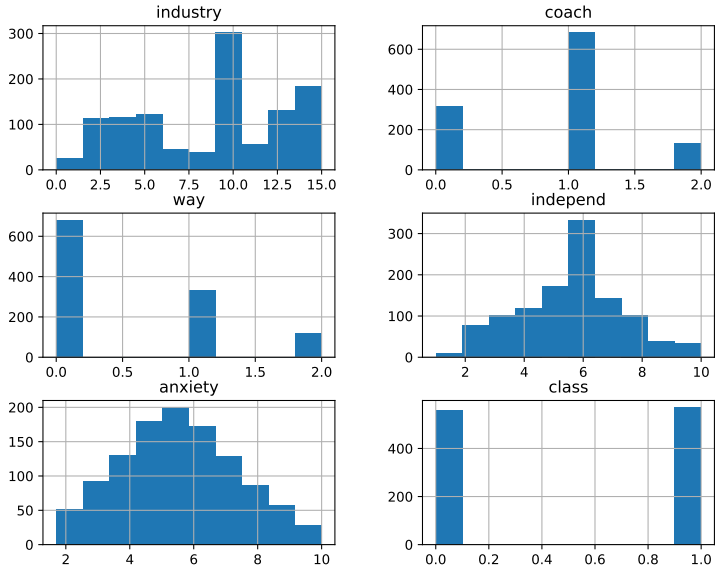

Figura 7 – Histograma das melhores características da base de dados Turnover

3.1.4 Attrition

A mais recente adição ao conjunto de bases analisadas é a *"Employee Attrition"* (65), uma base de dados fictícia que se destaca por seu tamanho significativo, contendo um total de 49.653 registros distribuídos em 18 colunas. Este recurso foi disponibilizado pela comunidade através do usuário *People HR Analytics Repository*, sendo sua última atualização registrada no ano de 2017.

As características detalhadas dessa base de dados são apresentadas na lista presente na sequência. O autor cita que as características contidas na base são, normalmente, as características contidas em sistemas de recursos humanos. Para determinar a distribuição das características numéricas foi utilizada a biblioteca *fitter* (62) do python para determinar a distribuição com base nas distribuições mais comuns apresentadas pela biblioteca, que são: *'cauchy', 'chi2', 'expon', 'exponpow', 'gamma', 'lognorm', 'norm', 'powerlaw', 'rayleigh' e 'uniform'*.

1. *EmployeeID*
    - descrição: identificador do funcionário
    - tipo: inteiro
    - valor mínimo: 1318
    - valor máximo: 8336
    - média: 4859.49
    - distribuição: *powerlaw*
2. *recorddate_key*
    - descrição: data em que foram obtidas as informações do funcionário
    - tipo: texto
3. *birthdate_key*
    - descrição: data de nascimento do funcionário
    - tipo: texto
4. *orighiredate_key*
    - descrição: data em que o funcionário ingressou na empresa
    - tipo: texto
5. *terminationdate_key*
    - descrição: data em que o funcionário deixou a empresa

- tipo: texto

6. *age*
    - descrição: idade do funcionário
    - tipo: inteiro
    - valor mínimo: 19
    - valor máximo: 65
    - média: 42.07
    - distribuição: *powerlaw*
7. *length_of_service*
    - descrição: tempo de serviço do funcionário
    - tipo: inteiro
    - valor mínimo: 0
    - valor máximo: 26
    - média: 10.43
    - distribuição: *rayleigh*
8. *city_name*
    - descrição: cidade onde o funcionário está localizado
    - tipo: texto
9. *department_name*
    - descrição: departamento do funcionário
    - tipo: texto
10. *job_title*
    - descrição: cargo do funcionário
    - tipo: texto
11. *store_name*
    - descrição: qual a loja que o funcionário trabalha
    - tipo: inteiro
    - valor mínimo: 1
    - valor máximo: 46

- média: 27.29
- distribuição: *uniform*

12. *gender_short*
    - descrição: gênero do funcionário abreviado
    - tipo: categorizado
    - categoria 1: M
    - categoria 2: F
13. *gender_full*
    - descrição: gênero do funcionário
    - tipo: categorizado
    - categoria 1: *Male*
    - categoria 2: *Female*
14. *termreason_desc*
    - descrição: motivo do desligamento
    - tipo: categorizado
    - categoria 1: *Not Applicable*
    - categoria 2: *Retirement*
    - categoria 3: *Resignation*
    - categoria 4: *Layoff*
15. *termtype_desc*
    - descrição: tipo de desligamento
    - tipo: categorizado
    - categoria 1: *Not Applicable*
    - categoria 2: *Voluntary*
    - categoria 3: *Involuntary*
16. *STATUS_YEAR*
    - descrição: qual era o ano dos dados
    - tipo: inteiro
    - valor mínimo: 2006

- valor máximo: 2015
- distribuição: *uniform*

17. *class*
    - descrição: informa se o funcionário está ativo ou inativo
    - tipo: categorizado
    - categoria 1: *ACTIVE*
    - categoria 2: *TERMINATED*
18. *BUSINESS_UNIT*
    - descrição: informa se o funcionário está alocado na sede ou em lojas
    - tipo: categorizado
    - categoria 1: *HEADOFFICE*
    - categoria 2: *STORES*

Explorar essa base ficcional oferece uma oportunidade única de investigar cenários amplos e diversificados relacionados à atração e retenção de talentos. A extensão substancial dessa base permite a análise detalhada de diferentes variáveis que podem influenciar o fenômeno de *turnover* no ambiente de trabalho.

Os resultados dessas análises são encapsulados visualmente nas figuras 8 e 9. O mapa de correlação, apresentado na figura 8, proporciona insights sobre as inter-relações entre as diversas características, enquanto o histograma, exibido na figura 9, destaca as distribuições individuais das características selecionadas pelo método *SelectKBest*, que nesse caso foram *recorddate_key, terminationdate_key, age, termreason_desc e termtype_desc*, além da classe tutora.

3.2 Abordagens de Aprendizados de Máquina

Nesta seção, serão exploradas abordagens diversas de aprendizado de máquina voltadas para a identificação de servidores potencialmente insatisfeitos em suas posições atuais. Para alcançar esse objetivo, serão empregadas técnicas de detecção de anomalias ou novidades, bem como classificadores e conjuntos de classificadores otimizados.

3.2.1 Detectores de Anomalias ou Novidades

A aplicação de algoritmos de I.A. para detecção de *turnoverattrition* foi iniciada com algoritmos de detecção de anomaliasnovidades, sendo eles o *Elliptic Envelope*, o *One*

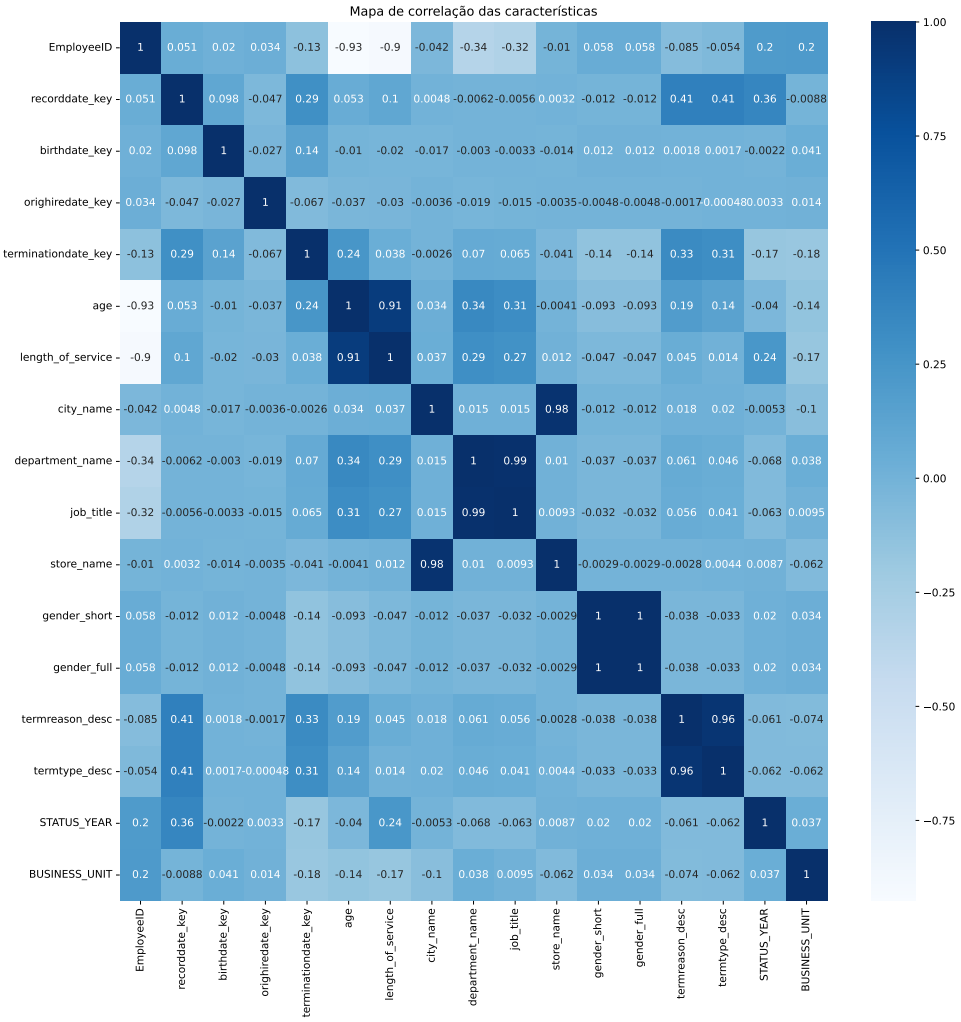

Figura 8 – Mapa de correlação das características da base de dados Attrition

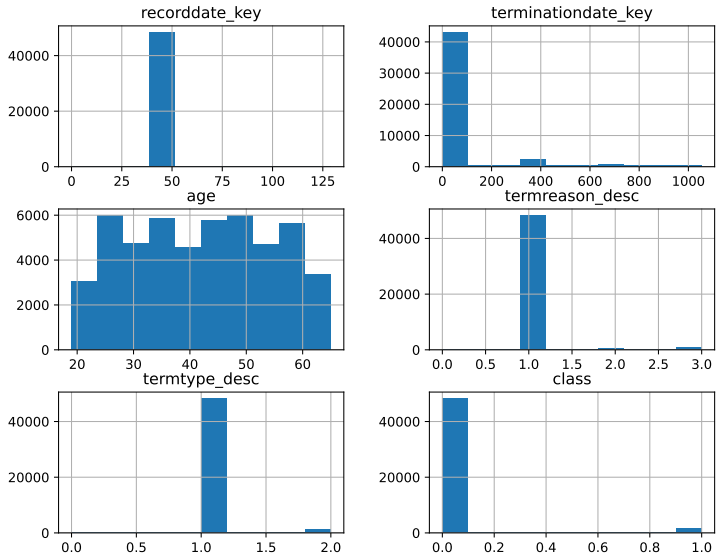

Figura 9 – Histograma das melhores características da base de dados Attrition

Class SVM, um *pipeline* utilizando *Nystroem* e *SGD One Class SVM*, o *Isolation Forest* e o *Local Outlier Factor* com cada uma das bases de dados. Nos testes realizados foram

executadas versões com alterações de configuração de parâmetros, sendo que foram três versões do *Elliptic Envelope*, três versões do *One Class SVM*, quatro versões do *pipeline - Nystroem e SGD One Class SVM*, quatro versões do *Isolation Forest* e seis versões do *Local Outlier Factor*. Os resultados obtidos serão apresentados no capítulo 04 do trabalho, mas é possível adiantar que embora tenha o tempo mais rápido de execução para todas as bases de dados, os resultados não foram surpreendentes. Onde em cada uma das técnicas foram utilizadas as métricas Acurácia Balanceada, Precisão, Recuperação e F1.

3.2.2 Classificadores

Dando continuidade com os testes de algoritmos de I.A. foram utilizadas três abordagens, sendo a primeira delas a que utilizou os classificadores clássicos e ocorreu a avaliação de cada uma das técnicas individuais com as métricas Acurácia Balanceada, *Precisão*, *Recuperação* e *F1*.

As técnicas avaliadas nesse caso foram *Ridge Classifier*, *Logistic Regression*, *Dummy Classifier*, *Linear Discriminant Analysis*, *Quadratic Discriminant Analysis*, *AdaBoost Classifier*, *Bagging Classifier*, *Extra Trees Classifier*, *Gradient Boosting Classifier*, *RandomForestClassifier*, *Hist Gradient Boosting Classifier*, *Bernoulli NB*, *Categorical NB*, *Complement NB*, *Multinomial NB*, *Gaussian NB*, *K Neighbors Classifier*, *Radius Neighbors Classifier*, *Nearest Centroid*, *MLPClassifier*, *Linear SVC*, *Nu SVC*, *SVC*, *Decision Tree Classifier*, *Extra Tree Classifier*, *XGB Classifier*, *XGB RF Classifier*, *DaskLGBMClassifier* e *LGBM Classifier*.

O *Grid Search* foi usado apenas para a execução controlada do *K-Fold*, com a divisão definida em 4 partes de forma a percorrer todos os dados de cada base de dados, e totalização de métricas, não sendo utilizado para teste exaustivo de parâmetros. A técnica *"cross_validate"* da mesma biblioteca não foi viável, como o *Grid Search*, devido a problemas de alto consumo de memória durante o treinamento das técnicas.

Após o processamento foi salvo o resultado que será apresentado na respectiva seção, mas já houve melhora nos resultados obtidos em comparação com as técnicas de detecção de anomalias/novidades.

3.2.3 Métodos Ensemble

Considerando a melhoria dos resultados obtidos pelas técnicas de classificação, foi feito um teste utilizando dois métodos *ensemble* para combinar técnicas e analisar seus resultados. Nesta situação foram testados o *Stacking Classifier* e o *Voting Classifier*.

O *Stacking Classifier*(58) faz o empilhamento de diversas técnicas e a predição delas é utilizada em um classificador final que analisa aquela predição, sendo que esse

último estimador é treinado com validação cruzada. O objetivo desse método é combinar os estimadores e obter o melhor resultado possível.

Já o *Voting Classifier*(66) combina diferentes classificadores e faz a classificação considerando os votos majoritários dos classificadores ou a médias das probabilidades preditas para realizar a classificação. Nos casos de votos majoritários, caso o número de resultados seja um empate, o *voting classifier* considera a ordem de classificação ascendente.

Nos testes realizados, foram retirados classificadores que consistiam em agrupar outras técnicas e foram utilizadas as seguintes técnicas de classificação *Ridge Classifier*, *Logistic Regression*, *Linear Discriminant Analysis*, *Quadratic Discriminant Analysis*, *Bernoulli NB*, *Gaussian NB*, *K Neighbors Classifier*, *MLPClassifier*, *Linear SVC*, *Nu SVC*, *SVC*, *Decision Tree Classifier*, *Extra Tree Classifier*, *XGB Classifier*, *XGB RF Classifier*, *DaskLGBMClassifier* e *LGBM Classifier*.

O *Grid Search* foi usado apenas para a execução controlada do *K-Fold* e totalização de métricas, não sendo utilizado para teste exaustivo de parâmetros, de forma semelhante ao utilizado com os classificadores isolados. A técnica *"cross_validate"* da mesma biblioteca não foi viável, como o *Grid Search*, devido a problemas de alto consumo de memória durante o treinamento das técnicas.

3.2.4 TPOT

O *Tree-based Pipeline Optimization Tool*(TPOT) é uma biblioteca de código aberto em python, definido por Olson, Edu e Moore(59), que utiliza algoritmos de otimização genética para sugerir um pipeline que possui a melhor combinação de técnicas de pré-processamento de dados e modelos de aprendizado de máquina para obter os melhores resultados. A geração automática dos pipelines ocorre da seguinte forma: para a base dados fornecida, o algoritmo genético cria 100 *pipelines* aleatórios e avalia a acurácia dos resultados da classificação desses pipelines, após essa etapa são selecionados os melhores 20 *pipelines* da população, cada um desses 20 melhores geram cinco cópias (filhos) na próxima geração da população. Ocorre então que 5% dos filhos gerados cruzam com outros filhos usando cruzamento de um ponto e 90% dos filhos restantes são alterados por um ponto, inserção ou mutação de encolhimento sendo 1/3 de chance de cada um ocorrer. Esse processo é repetido por 100 gerações, adicionando e otimizando os operadores do pipeline que melhoram a acurácia e os operadores de poda que degradam a acurácia da classificação, então o algoritmo seleciona o pipeline com a melhor acurácia da curva de Pareto.

3.2.5 Aprendizado Profundo

Com relação a aplicação das técnicas de aprendizado profundo houve uma tentativa de aplicação nas bases de dados selecionadas, no entanto o tempo de execução e os recursos computacionais necessários para execução destas técnicas fizeram com que elas não fossem consideradas nesta pesquisa, pois no ambiente de aplicação prático seriam necessários recursos muito elevados para implantação na universidade e o tempo para gerar um modelo treinado seria alto.

3.3 Métricas de Avaliação

As métricas são medidas importantes para que se possa avaliar os resultados de forma quantitativa. Como forma de calcular os resultados das métricas a matriz de confusão é utilizada, que é uma tabela que compara as previsões do modelo com as bases de teste conforme ilustrado na Figura 10.

		Valor Predito	
		Sim	**Não**
Valor Real	**Sim**	Verdadeiro Positivo (VP)	Falso Negativo (FN)
	Não	Falso Positivo (FP)	Verdadeiro Negativo (VN)

Figura 10 – Matriz de Confusão - adaptado de Scikit-learn (1)

Os elementos :

- **VP - Verdadeiro Positivo:** Representa a quantidade de acertos do modelo ou seja, quantas vezes o modelo foi capaz de classificar um dado como *turnover* quando de fato se tratava de um caso de *turnover*.
- **FP - Falso Positivo:** É a quantidade de vezes em que um dado foi classificado como *turnover* quando de fato era não *turnover*.
- **VN - Verdadeiro Negativo:** É a quantidade de vezes que o modelo classificou uma dado como não *turnover* e de fato se tratava de um caso de não *turnover*.
- **FN - Falso Negativo:** É a quantidade de vezes em que o modelo classificou um dado não *turnover* quando na verdade se tratava de *turnover*.

Com os valores obtidos pela matriz de confusão, é feito o cálculo das métricas que serão úteis em uma avaliação mais abrangente, e no caso da pesquisa são:

- ***Precisão***: A medida da precisão é o quociente da divisão de VP pela soma de VP com FP e é dada pela equação 3.1. Para o cenário de identificação de *turnover*, é uma medida importante para saber o percentual de acertos do modelo ao predizer de forma correta a classificação de uma situação de *turnover*.

$$\text{Precisão} = \frac{VP}{VP + FP} \tag{3.1}$$

- ***Recuperação*:** O *Recuperação* é uma taxa de verdadeiros positivos ou seja, o quociente entre VP pela soma dos valores de VP e FN, dada pela equação 3.2. Essa taxa mostra qual o percentual de acertos das predições corretas(VP) em relação à todas as corretas, representadas pelos acertos e pelos erros.

$$\text{Recuperação} = \frac{VP}{VP + FN} \tag{3.2}$$

- ***F1 Score*:** A métrica F1-Score é uma média harmônica entre precisão e recuperação, que mede, respectivamente, a proporção de predições positivas que são corretas e a proporção de exemplos positivos que são corretamente classificados. O F1-Score varia de 0 a 1, sendo 1 o melhor valor possível e 0 o pior.

$$\text{F1} = 2 \times \frac{\text{Precisão} \times \textit{Recuperação}}{\text{Precisão} + \textit{Recuperação}} \tag{3.3}$$

- ***Acurácia*:** A acurácia é a proporção de previsões corretas em relação ao número total de previsões. Ela mede o quão bem o modelo está acertando as previsões em geral. Ela é dada pela equação 3.4.

$$\text{Acurácia} = \frac{VP + VN}{VP + VN + FP + FN} \tag{3.4}$$

- ***Acurácia Balanceada:*** Quando existe um desbalanceamento dos dados, como é o caso dessa pesquisa, a métrica da acurácia pode não ser útil ou até mesmo fornecer uma medida equivocada, uma vez que os valores classificados como verdadeiros negativos podem mascarar as classificações baixas de verdadeiros positivos, fornecendo assim uma classificação não confiável.

  Dessa forma, é utilizada a acurácia balanceada, que consiste em calcular a taxa de verdadeiros positivos e verdadeiros negativos, como demonstrado na equação 3.5.

$$\text{Acurácia Balanceada} = \frac{1}{2}\left(\frac{VP}{VP + FN} + \frac{VN}{VN + FP}\right) \tag{3.5}$$

  Ao utilizar-se a acurácia balanceada, no qual é levada em conta os acertos de cada classe de forma igualitária, é mostrado um valor mais próximo do quanto o modelo

consegue acertar de cada classe, penalizando assim a métrica quando alguma classe tem uma alta taxa de erros.

- ***Tempo de Execução*:** O tempo de execução, apesar de não estar ligado a nenhuma métrica sobre os erros e acertos, deve ser levada em consideração, uma vez que na aplicação do modelo no ambiente real, o tempo tem um impacto significativo.

4 Resultados

Neste capítulo, são demonstrados os resultados obtidos ao aplicar cada técnica de aprendizado de máquina utilizada na pesquisa em cada base de dados. A análise destacará o desempenho de cada técnica em diferentes contextos de dados.

Para facilitar a transparência e a replicabilidade, foi disponibilizado o código-fonte da pesquisa em um repositório do GitHub (60). O código inclui todo o processo de processamento de dados e está disponível para qualquer pessoa examinar, entender e replicar.

Ao compartilhar o código-fonte da pesquisa no GitHub, fica demonstrado o compromisso da pesquisa com a transparência e o rigor científico. Isso permite que a comunidade acadêmica e científica avalie o trabalho e colabore com as descobertas resultantes.

4.1 Resultados Detectores de Anomalias/Novidades

Nesta seção, é apresentada uma análise dos resultados obtidos ao aplicar diferentes técnicas de detecção de anomalias a cada base. A análise destaca o desempenho de cada técnica em diferentes contextos de dados, incluindo as métricas de desempenho mencionadas na seção 3.3 e tempo de execução.

Os resultados da análise permitem identificar as técnicas que se destacam em termos de desempenho, bem como aquelas que podem requerer ajustes ou modificações. A análise do tempo de execução também ajuda a determinar a eficiência operacional de cada técnica.

A seção fornece um recurso valioso para orientar a seleção e otimização de técnicas de detecção de anomalias. Nesta seção serão apresentados os resultados dos detectores de anomalias para cada uma das bases de dados, onde é possível verificar o resultado de cada métrica utilizada e o tempo de execução de cada uma das técnicas utilizadas.

4.1.1 Hr Comma Sep

Na Tabela 3 são apresentados os resultados da aplicação das técnicas de aprendizado de máquinas para identificação de anomalias/novidades e o melhor resultado obtido pela técnica *One Class SVM* foi de apenas 57.88% para esta base de dados com o tempo de execução de 5.25 segundos, o que demonstra que mesmo a melhor técnica não foi precisa na identificação de casos de *turnover/attrition*.

Já as técnicas que foram executadas em maior velocidade que foram duas variações

do *Pipeline* obtiveram resultado de aproximadamente 50%, o que inviabiliza sua utilização, apesar do tempo.

Os resultados variaram de 44.34%, utilizando a técnica *Local Outlier Factor* a 57.89%, a técnica *One Class SVM* o que faz perceber que a melhor técnica não chegou a ter um resultado significativo para a detecção de anomalias nessa base de dados. No entanto a pontuação foi de 69.44% tanto nas métricas Precisão, capaz de identificar as previsões corretas do modelo, quanto na Recuperação, importante para o nosso modelo por identificar os casos de *turnover/attrition*. Esses resultados fazem com que a pontuação com a métrica F1 também seja bom.

Tabela 3 – Resultado das técnicas de detecção de anomalias/novidades para a base de dados HR-Comma-Sep

Técnica	**Acurácia Balanceada**	**Precisão**	**Recuperação**	**F1**	**Tempo**
OneClassSVM	0.57885	0.69443	0.69445	0.69442	5.249
EllipticEnvelope	0.53989	0.66616	0.66618	0.66615	2.102
EllipticEnvelope	0.53989	0.66616	0.66618	0.66615	2.315
EllipticEnvelope	0.53341	0.66144	0.66144	0.66144	2.744
Pipeline	0.50412	0.46811	0.4377	0.6408	0.066
OneClassSVM	0.50000	0.09157	0.23808	0.05668	3.061
LocalOutlierFactor	0.49959	0.65881	0.76085	0.61287	16.501
LocalOutlierFactor	0.49709	0.65693	0.75705	0.58875	17.221
Pipeline	0.48774	0.54789	0.5119	0.62862	0.057
Pipeline	0.48573	0.62464	0.62237	0.62697	0.527
Pipeline	0.48466	0.62648	0.62691	0.62604	0.636
IsolationForest	0.47993	0.62264	0.62264	0.62264	1.535
OneClassSVM	0.46538	0.612	0.61191	0.61209	13.302
LocalOutlierFactor	0.45928	0.60916	0.61097	0.60737	0.828
IsolationForest	0.45476	0.60437	0.60437	0.60437	0.996
IsolationForest	0.45407	0.60389	0.60391	0.60387	0.205
LocalOutlierFactor	0.45175	0.60646	0.61211	0.6011	3.68
LocalOutlierFactor	0.44665	0.60266	0.60831	0.5973	0.49
IsolationForest	0.44465	0.59704	0.59704	0.59704	0.215
LocalOutlierFactor	0.4434	0.59749	0.59924	0.59577	4.003

A técnica de detecção de anomalias para esta base de dados possui os seguintes parâmetros:

1. *OneClassSVM*
   - *kernel ’poly’*
   - *degree 3*
   - *gamma ’auto’*
   - *coef0 0.0*
   - *tol 0.001*

- *nu 0.23*
- *shrinking True*
- *cache_size 200*
- *shrinking True*
- *max_iter =1*

As demais métricas para o melhor resultado ...

4.1.2 IBM Employee Attrition

Os resultados da detecção de anomalias/novidades para a base de dados *IBM Employee Attrition* estão presentes na Tabela 4 e é possível verificar que o melhor resultado foi obtido pela técnica *Local Outlier Factor* com uma pontuação de 52.80% na métrica Acurácia Balanceada. No entanto a pontuação foi de 74.55% na métrica Precisão, que informa o quanto o modelo realmente acertou sobre os casos de *turnover/attrition*. Já a pontuação na métrica Recuperação foi de 75.57%, importante para o nosso modelo por identificar os casos de *turnover/attrition* registrados corretamente pelo modelo. Consequentemente a pontuação com a métrica F1 foi de 75.11%, demonstrando um bom desempenho do modelo, embora a pontuação da acurária tenha sido um pouco menor.

Tabela 4 – Resultado das técnicas de detecção de anomalias/novidades para a base de dados IBM Employee Attrition

Técnica	**Acurácia Balanceada**	**Precisão**	**Recuperação**	**F1**	**Tempo**
LocalOutlierFactor	0.52803	0.74551	0.75714	0.75112	0.183
LocalOutlierFactor	0.52535	0.74345	0.74694	0.74518	0.188
IsolationForest	0.52211	0.7415	0.7415	0.7415	0.438
IsolationForest	0.51959	0.74014	0.74014	0.74014	0.661
LocalOutlierFactor	0.51399	0.73726	0.74218	0.73969	0.109
OneClassSVM	0.50702	0.73333	0.73333	0.73333	0.067
LocalOutlierFactor	0.50661	0.73333	0.74694	0.73991	0.101
IsolationForest	0.50450	0.73197	0.73197	0.73197	0.073
OneClassSVM	0.50000	0.02599	0.16122	0.04477	0.043
Pipeline	0.50000	0.70354	0.83878	0.76523	0.015
Pipeline	0.50000	0.70354	0.83878	0.76523	0.034
Pipeline	0.50000	0.70354	0.83878	0.76523	0.009
Pipeline	0.50000	0.70354	0.83878	0.76523	0.032
IsolationForest	0.49696	0.72789	0.72789	0.72789	0.073
LocalOutlierFactor	0.49314	0.72556	0.7415	0.73327	0.412
OneClassSVM	0.49308	0.71248	0.22109	0.18184	0.614
EllipticEnvelope	0.47683	0.71701	0.71701	0.71701	0.953
EllipticEnvelope	0.47683	0.71701	0.71701	0.71701	0.906
EllipticEnvelope	0.47683	0.71701	0.71701	0.71701	0.889
LocalOutlierFactor	0.4628	0.70885	0.71633	0.71255	0.42

Conforme apresentado, pode ser observada uma variação de resultados entre 46.28%, uma variação do *Local Outlier Factor* e 52.80%, que foi outra variação da mesma técnica, com diferentes parâmetros.

A técnica de detecção de anomalias com melhor resultado para esta base de dados possui os seguintes parâmetros:

1. *LocalOutlierFactor*
    - *n_neighbors=20*
    - *algorithm='auto'*
    - *leaf_size=30*
    - *metric='braycurtis'*
    - *p=2*
    - *metric_params=None*
    - *contamination=0.16*
    - *novelty=True*
    - *n_jobs=None*

4.1.3 Turnover

Já os resultados para a base de dados *Turnover* podem ser verificados na Tabela 5. A peculiaridade dessa base é que ela é a única base que apresenta um número maior de casos de *turnover* do que dos casos de não *turnover*. Novamente o *Local Outlier Factor* foi o que apresentou melhor resultado, no entanto foi um resultado de 53.97% na identificação dos casos de *turnover/attrition*, o que não é interessante para a pesquisa por ser um valor próximo ao equilíbrio.

Das cinco melhores técnicas para essa base de dados, três são variações da técnica *Local Outlier Factor* e estas variaram entre 51.13% e 53.97%, o que representa um valor equilibrado e sem destaque, também sendo interessante para o objetivo pesquisa.

Os resultados das técnicas variaram entre 48.15%, sendo este o resultado com o valor mais baixo e 53.87%, sendo este último o resultado obtido pela técnica *Local Outlier Factor*. A pontuação um pouco mais baixa foi confirmada pelas métricas Precisão e Recuperação, que obtiveram pontuação de 53.97% e 53.94%, respectivamente, o que confirma o baixo desempenho do modelo sobre os casos de *turnover/attrition* para essa base de dados. De forma semelhante a pontuação com a métrica F1 foi de 53.91%, não demonstrando variação dos demais resultados.

Os parâmetros utilizados pelo melhor resultado estão apresentados na lista abaixo:

Tabela 5 – Resultado das técnicas de detecção de anomalias/novidades para a base de dados Turnover

Técnica	Acurácia Balanceada	Precisão	Recuperação	F1	Tempo
LocalOutlierFactor	0.5397	0.53988	0.53942	0.53916	0.066
LocalOutlierFactor	0.53775	0.53783	0.53764	0.53763	0.099
OneClassSVM	0.53498	0.53504	0.53499	0.535	0.057
EllipticEnvelope	0.53322	0.53329	0.53322	0.53323	0.704
LocalOutlierFactor	0.51139	0.5115	0.51107	0.51071	0.036
LocalOutlierFactor	0.51117	0.51124	0.51107	0.51105	0.056
LocalOutlierFactor	0.5058	0.50961	0.50221	0.44901	0.126
Pipeline	0.50492	0.50508	0.50576	0.50312	0.024
LocalOutlierFactor	0.50227	0.5026	0.50044	0.48753	0.148
OneClassSVM	0.50000	0.25579	0.50576	0.33975	0.048
IsolationForest	0.49956	0.49963	0.49956	0.49957	0.329
Pipeline	0.49799	0.4979	0.49956	0.49013	0.031
IsolationForest	0.49779	0.49786	0.49779	0.4978	0.07
Pipeline	0.49628	0.49631	0.4969	0.49547	0.009
IsolationForest	0.49602	0.49609	0.49601	0.49603	0.326
Pipeline	0.49154	0.4907	0.49336	0.48036	0.006
IsolationForest	0.48716	0.48723	0.48716	0.48717	0.067
EllipticEnvelope	0.48185	0.48191	0.48184	0.48186	0.701
EllipticEnvelope	0.48185	0.48191	0.48184	0.48186	0.701
OneClassSVM	0.48158	0.48141	0.48096	0.47943	0.157

1. *LocalOutlierFactor*
    - *n_neighbors¯20*
    - *algorithm¯'auto'*
    - *leaf_size¯30*
    - *metric¯'braycurtis'*
    - *p¯2*
    - *metric_params¯None*
    - *contamination¯0.5*
    - *novelty¯True*
    - *n_jobs=None*

4.1.4 Attrition

A Tabela 6 apresenta os resultados para a base de dados *Attrition*, maior base de dados utilizada na pesquisa, conforme apresentado em 3.1.4. Nos resultados obtidos pode ser verificado que a técnica *Isolation Forest* obteve uma pontuação de 92.15% na identificação dos casos de *turnover/attrition*, sendo o melhor resultado das técnicas de detecção de anomalias/novidades dentre as execuções na pesquisa. O bom resultado das

técnicas é confirmado pela pontuação nas demais métricas que obtiveram a pontuação de 99.09% na métrica Precisão, que informa o quanto o modelo realmente acertou sobre os casos de *turnover/attrition* e também de 99.09% na métrica Recuperação, como já mencionado, esta é uma métrica importante para a pesquisa pois é capaz de identificar os casos de *turnover/attrition* registrados corretamente pelo modelo. Os bons resultados também foram obtidos na métrica F1, com a pontuação de 99.09% para o melhor resultado, demonstrando um bom desempenho do modelo, embora a pontuação da acurária tenha sido um pouco menor.

As cinco melhores ocorrências foram compostas por uma técnica *One Class SVM* e quatro variações da técnica *Isolation Forest*, sendo que todas estas tiveram resultado superior a 80%.

Tabela 6 – Resultado das técnicas de detecção de anomalias/novidades para a base de dados Attrition

Técnica	**Acurácia Balanceada**	**Precisão**	**Recuperação**	**F1**	**Tempo**
IsolationForest	0.92156	0.9909	0.9909	0.9909	3.048
IsolationForest	0.92087	0.99082	0.99082	0.99082	2.563
IsolationForest	0.81258	0.97825	0.97825	0.97825	0.486
IsolationForest	0.81223	0.97821	0.97821	0.97821	0.479
OneClassSVM	0.66561	0.9579	0.62663	0.74452	496.872
EllipticEnvelope	0.60745	0.95444	0.95444	0.95444	5.09
EllipticEnvelope	0.60745	0.95444	0.95444	0.95444	6.506
EllipticEnvelope	0.60745	0.95444	0.95444	0.95444	4.053
LocalOutlierFactor	0.553	0.94827	0.94947	0.94887	141.822
LocalOutlierFactor	0.54173	0.94723	0.95102	0.9491	146.213
LocalOutlierFactor	0.53554	0.94623	0.94788	0.94705	67.042
LocalOutlierFactor	0.53159	0.94594	0.94971	0.9478	67.594
OneClassSVM	0.50956	0.94308	0.94306	0.94307	6.891
LocalOutlierFactor	0.50513	0.94479	0.96737	0.95472	330.502
LocalOutlierFactor	0.50443	0.94731	0.96919	0.95548	332.01
Pipeline	0.5	0.94108	0.97009	0.95537	0.546
Pipeline	0.5	0.94108	0.97009	0.95537	0.105
Pipeline	0.5	0.94108	0.97009	0.95537	0.509
Pipeline	0.5	0.94108	0.97009	0.95537	0.106
OneClassSVM	0.5	0.00089	0.02991	0.00174	5.458

Os parâmetros utilizados pelo melhor resultado para esta base de dados estão apresentados na lista abaixo:

1. *IsolationForest*
    - *n_estimators=100*
    - *max_samples='auto'*
    - *contamination='auto'*
    - *max_features=1.0*

- *bootstrap=False*
- *n_jobs=None*
- *random_state=42*
- *warm_start=False*

De maneira geral a abordagem de detecção de anomalias/novidades não apresenta resultados interessantes, no entanto vale ressaltar que os resultados apresentados na base de dados *Attrition* indicam um potencial que deve ser analisado de forma mais profunda em trabalhos futuros. Para o tema em questão e, de acordo com as bases disponíveis, não é viável a utilização da abordagem de detecção de anomalias/novidades.

4.2 Resultados Classificadores

Nesta seção, será feita uma análise dos resultados obtidos por meio da aplicação individual de cada técnica de classificação em todos os conjuntos de dados (*datasets*) abordados nesta pesquisa. A avaliação desses resultados se dará por meio das métricas previamente discutidas na seção 3.3, garantindo uma abordagem consistente e compreensível.

A fim de fornecer uma visualização clara e acessível, os resultados estão organizados em tabelas que apresentam o desempenho de cada técnica em diferentes contextos de dados. Cada tabela será estruturada de maneira a destacar as métricas relevantes, tais como acurácia balanceada, precisão, recuperação, F1, entre outras.

A apresentação desses resultados em formato tabular não apenas simplifica a compreensão, mas também permite uma comparação direta entre as técnicas avaliadas. Isso proporciona uma visão abrangente do desempenho relativo de cada abordagem em relação aos diversos conjuntos de dados considerados, facilitando a identificação de padrões e tendências.

Essa análise minuciosa dos resultados individuais de cada técnica contribuirá significativamente para a tomada de decisões informadas em relação à escolha e otimização de métodos de classificação, enriquecendo assim o entendimento sobre a eficácia de cada abordagem no contexto específico desta pesquisa.

4.2.1 HR Comma Sep

Para a base de dados *HR Comma Sep* é possível verificar pelos resultados apresentados na Tabela 7 que a técnica que obteve melhores resultados foi a técnica *Random Forest Classifier* com uma pontuação de 98.12% na Acurácia Balanceada. Este resultado também foi superior à 98% nas métricas Precisão - 98.15%, Recuperação - 98.12% e F1 -

98.12%, o que demonstra que a técnica de classificação com melhor resultado na Acurácia também foi capaz de gerar bons resultados nas demais métricas, indicando que existe precisão nas previsões positivas e existe a capacidade de identificar todas as instâncias positivas.

As cinco técnicas que tiveram melhor resultado foram *Random Forest Classifier, Extra Trees Classifier, XGB Classifier, Hist Gradient Boosting Classifier e LGBM Classifier*, todas apresentando um resultado acima de 97.45%, sendo que destas técnicas, a que teve menor tempo de execução foi a técnica *Hist Gradient Boosting Classifier*.

Esse resultado demonstra que o desempenho das técnicas de classificação para esta base de dados foram melhores em comparação com os resultados obtidos com as técnicas de detecção de anomalias/novidades.

Tabela 7 – Resultado das técnicas de classificação para a base de dados HR Comma Sep

Técnica	**Acurácia Balanceada**	**Precisão**	**Recuperação**	**F1**	**Tempo**
RandomForestClassifier	0.98119	0.98148	0.98119	0.98118	0.38411
ExtraTreesClassifier	0.9804	0.98057	0.9804	0.9804	0.6672
XGBClassifier	0.97826	0.97831	0.97826	0.97825	0.02542
HistGradientBoostingClassifier	0.97506	0.97521	0.97506	0.97506	0.0129
LGBMClassifier	0.9748	0.97506	0.9748	0.9748	0.02242
BaggingClassifier	0.97397	0.97411	0.97397	0.97397	0.04626
GradientBoostingClassifier	0.9541	0.95456	0.9541	0.95409	0.04849
ExtraTreeClassifier	0.94785	0.94797	0.94785	0.94784	0.00116
XGBRFClassifier	0.9426	0.94264	0.9426	0.9426	0.01014
DecisionTreeClassifier	0.9398	0.93982	0.9398	0.9398	0.00205
AdaBoostClassifier	0.9377	0.93773	0.9377	0.9377	0.10184
KNeighborsClassifier	0.93463	0.93885	0.93463	0.93447	0.00109
MLPClassifier	0.90615	0.90828	0.90615	0.90601	6.32255
QuadraticDiscriminantAnalysis	0.89504	0.89614	0.89504	0.89496	0.01155
NuSVC	0.86546	0.87115	0.86546	0.86462	1.51713
LogisticRegression	0.76553	0.77133	0.76553	0.76432	0.08707
RidgeClassifier	0.75354	0.7605	0.75354	0.75207	0.00386
LinearDiscriminantAnalysis	0.7535	0.76046	0.7535	0.75203	0.00947
ComplementNB	0.71452	0.78074	0.71452	0.69661	0.00313
MultinomialNB	0.71452	0.78074	0.71452	0.69661	0.00068
SVC	0.69343	0.69406	0.69343	0.69312	4.15655
GaussianNB	0.6779	0.75399	0.6779	0.64893	0.00056
BernoulliNB	0.6103	0.70065	0.6103	0.56012	0.0016
LinearSVC	0.59437	0.60972	0.59437	0.52619	0.0039
NearestCentroid	0.53395	0.53396	0.53395	0.53393	0.0004
DummyClassifier	0.5	0.25	0.5	0.33333	0.00415

As técnicas *CategoricalNB*, *RadiusNeighborsClassifier* e *DaskLGBMClassifier* não conseguiram obter resultados, pois os dados tabulares necessários para estas técnicas não foram alcançados.

Os parâmetros utilizados pela técnica com melhor resultado para esta base de dados estão apresentados na lista abaixo:

1. *RandomForestClassifier*

- *n_estimators=100*
- *criterion='gini'*
- *max_depth=None*
- *min_samples_split=2*
- *min_samples_leaf=1*
- *min_weight_fraction_leaf=0.0*
- *max_features='sqrt'*
- *max_leaf_nodes=None*
- *min_impurity_decrease=0.0*
- *bootstrap=True*
- *oob_score=False*
- *n_jobs=None*
- *random_state=42*
- *warm_start=False*
- *class_weight=None*
- *ccp_alpha=0.0*
- *max_samples=None*

4.2.2 IBM Employee Attrition

Para a base de dados *IBM Employee Attrition* os resultados podem ser verificados na Tabela 8 com o melhor resultado obtido com a técnica *Extra Trees Classifier* com 90.03% de pontuação na Acurácia Balanceada, além de 90.80% na Precisão, 90.03% na Recuperação e 89.92% na F1.

Os resultados variaram de 49.82% a 90.03%, sendo que as cinco técnicas que tiveram um melhor resultado foram *Extra Trees Classifier, Quadratic Discriminant Analysis, Ridge Classifier, Linear Discriminant Analysis e Random Forest Classifier*. Embora não tenha o melhor tempo de execução entre as cinco melhores técnicas, a técnica *Extra Trees Classifier* gerou um resultado aproximadamente 10% melhor em comparação com a segunda melhor técnica em todas as métricas.

As técnicas *CategoricalNB*, *RadiusNeighborsClassifier* e *DaskLGBMClassifier* não conseguiram obter resultados, pois os dados tabulares necessários para estas técnicas não foram alcançados.

Os parâmetros utilizados pela técnica com melhor resultado, *Extra Trees Classifier*, para esta base de dados estão apresentados na lista abaixo:

Tabela 8 – Resultado das técnicas de classificação para a base de dados IBM Employee Attrition

Técnica	Acurácia Balanceada	Precisão	Recuperação	F1	Tempo
ExtraTreesClassifier	0.90033	0.90795	0.90027	0.89928	0.01068
QuadraticDiscriminantAnalysis	0.79977	0.81255	0.7997	0.79725	0.00615
RidgeClassifier	0.78111	0.79	0.78105	0.77914	0.00684
LinearDiscriminantAnalysis	0.7807	0.78966	0.78065	0.77872	0.00265
RandomForestClassifier	0.74329	0.7659	0.74329	0.71409	0.10631
ExtraTreeClassifier	0.73566	0.74228	0.73561	0.73372	0.00012
KNeighborsClassifier	0.70804	0.75707	0.70803	0.69331	0.00063
GaussianNB	0.70244	0.7191	0.70238	0.6975	0.00109
BernoulliNB	0.69792	0.6981	0.6979	0.69782	0.00166
XGBRFClassifier	0.68048	0.68954	0.68043	0.6529	0.82013
BaggingClassifier	0.67763	0.70927	0.67758	0.65503	0.01418
DecisionTreeClassifier	0.63951	0.64671	0.63949	0.62199	0.00188
LogisticRegression	0.59975	0.602	0.5998	0.59142	0.02024
NearestCentroid	0.59164	0.60193	0.59165	0.58072	0.00396
MLPClassifier	0.58891	0.66512	0.58884	0.52919	0.10034
XGBClassifier	0.57595	0.64568	0.5758	0.5031	0.00895
SVC	0.55918	0.56419	0.55921	0.54589	0.00476
NuSVC	0.55235	0.55581	0.55234	0.53976	0.02282
GradientBoostingClassifier	0.54716	0.58854	0.54703	0.47767	0.13649
MultinomialNB	0.53973	0.54085	0.53974	0.53545	0.00044
ComplementNB	0.53973	0.54085	0.53974	0.53545	0.0011
HistGradientBoostingClassifier	0.53658	0.5292	0.53652	0.46111	0.01996
LGBMClassifier	0.52725	0.57368	0.5272	0.44986	0.00229
AdaBoostClassifier	0.50575	0.47475	0.50573	0.43668	0.01221
DummyClassifier	0.5	0.2496	0.49959	0.33288	0.00611
LinearSVC	0.49826	0.5296	0.49841	0.40596	0.00996

1. *ExtraTreesClassifier*

   - *n_estimators=100*
   - *criterion='gini'*
   - *max_depth=None*
   - *min_samples_split=2*
   - *min_samples_leaf=1*
   - *min_weight_fraction_leaf=0.0*
   - *max_features='sqrt'*
   - *max_leaf_nodes=None*
   - *min_impurity_decrease=0.0*
   - *bootstrap=False*
   - *oob_score=False*
   - *n_jobs=None*
   - *random_state=42*
   - *warm_start=False*

- *class_weight=None*
- *ccp_alpha=0.0*
- *max_samples=None*

4.2.3 Turnover

A Tabela 9 apresenta os resultados para a base de dados *Turnover* onde é possível verificar que a melhor técnica de classificação foi o *Hist Gradient Boosting Classifier* com uma pontuação de 59.45% na Acurácia Balanceada, 59.51% na Precisão, 59.45% na Recuperação e 59.39% na F1. Com esses resultados é possível perceber que não houve variação dos resultados entre as métricas e o resultado geral permaneceu em torno de 59%.

Tabela 9 – Resultado das técnicas de classificação para a base de dados Turnover

Técnica	Acurácia Balanceada	Precisão	Recuperação	F1	Tempo
HistGradientBoostingClassifier	0.59459	0.59516	0.59455	0.59396	0.00995
BaggingClassifier	0.59283	0.59649	0.59279	0.58928	0.00524
RandomForestClassifier	0.59277	0.59286	0.59278	0.59268	0.00659
LGBMClassifier	0.58751	0.58772	0.58753	0.58726	0.00195
XGBClassifier	0.58401	0.58415	0.58402	0.5838	0.23921
GradientBoostingClassifier	0.57084	0.57143	0.5709	0.56978	0.00411
ExtraTreesClassifier	0.56999	0.57008	0.57003	0.56982	0.00625
XGBRFClassifier	0.56389	0.56499	0.5639	0.56192	0.00438
AdaBoostClassifier	0.55855	0.55863	0.5586	0.55809	0.01004
NuSVC	0.55332	0.5542	0.55338	0.55173	0.01088
GaussianNB	0.54981	0.55015	0.54989	0.5486	0.00133
DecisionTreeClassifier	0.5448	0.5512	0.54461	0.53565	0.00208
MLPClassifier	0.54209	0.54344	0.54199	0.54032	0.06445
ExtraTreeClassifier	0.5385	0.53908	0.5385	0.53753	0.00012
RidgeClassifier	0.5349	0.53534	0.53498	0.53411	0.00311
LinearDiscriminantAnalysis	0.5349	0.53534	0.53498	0.53411	0.00377
QuadraticDiscriminantAnalysis	0.53317	0.53328	0.53326	0.53248	0.00157
LogisticRegression	0.5288	0.52918	0.52884	0.52822	0.00347
NearestCentroid	0.52172	0.52368	0.52182	0.51389	8e-05
KNeighborsClassifier	0.51665	0.51666	0.5166	0.51544	0.00122
SVC	0.51477	0.51496	0.51487	0.51363	0.00095
BernoulliNB	0.50867	0.50891	0.50875	0.50697	0.0015
ComplementNB	0.506	0.50528	0.50607	0.50119	0.00154
MultinomialNB	0.506	0.50528	0.50607	0.50119	0.00017
DummyClassifier	0.5	0.24912	0.49912	0.33236	0.00047
LinearSVC	0.48425	0.35609	0.48339	0.35536	0.02336

As variação entre os resultados foi de 48.42% e 59.45%, sendo este último o melhor resultado. Já as cinco melhores técnicas foram *Hist Gradient Boosting Classifier, Bagging Classifier, Random Forest Classifier, LGBM Classifier e XGB Classifier*, sendo que a variação de resultados entre elas foi baixo.

Esta é a única base de dados real e o fato de ser o que apresenta mais ocorrências de *turnover* do que não *turnover* traz uma peculiaridade que pode ter refletida nos resultados dos classificadores.

As técnicas *CategoricalNB*, *RadiusNeighborsClassifier* e *DaskLGBMClassifier* não conseguiram obter resultados, pois os dados tabulares necessários para estas técnicas não foram alcançados.

Os parâmetros utilizados pela técnica com melhor resultado para esta base de dados estão apresentados na lista abaixo:

1. *HistGradientBoostingClassifier*
    - *loss='log_loss'*
    - *learning_rate=0.1*
    - *max_iter=100*
    - *max_leaf_nodes=31*
    - *max_depth=None*
    - *min_samples_leaf=20*
    - *l2_regularization=0.0*
    - *max_bins=255*
    - *categorical_features=None*
    - *monotonic_cst=None*
    - *interaction_cst=None*
    - *warm_start=False*
    - *early_stopping='auto'*
    - *scoring='loss'*
    - *validation_fraction=0.1*
    - *n_iter_no_change=10*
    - *tol=1e07*
    - *random_state=42*
    - *class_weight=None*

4.2.4 Attrition

As técnicas de classificação aplicadas na base de dados *Attrition* obtiveram um resultado bom de forma geral, conforme Tabela 10, onde a técnica *Decision Tree Classifier* obteve uma pontuação de 99.87% na métrica Acurácia Balanceada, indicando uma melhora no resultado em comparação com o resultado da detecção de anomalias/novidades,

Tabela 10 – Resultado das técnicas de classificação para a base de dados Attrition

Técnica	Acurácia Balanceada	Precisão	Recuperação	F1	Tempo
DecisionTreeClassifier	0.99869	0.9987	0.99869	0.99869	0.00061
GradientBoostingClassifier	0.99826	0.99827	0.99826	0.99826	0.36376
HistGradientBoostingClassifier	0.99787	0.9979	0.99787	0.99787	0.79199
XGBClassifier	0.99752	0.99754	0.99752	0.99752	0.63068
LGBMClassifier	0.98726	0.98838	0.98726	0.98723	0.00344
XGBRFClassifier	0.98616	0.98724	0.98616	0.98613	0.0096
BaggingClassifier	0.97977	0.98133	0.97977	0.97973	0.0294
AdaBoostClassifier	0.97298	0.97741	0.97298	0.9727	0.36625
GaussianNB	0.9709	0.97184	0.9709	0.97088	0.00652
ExtraTreeClassifier	0.9648	0.9666	0.9648	0.96475	0.00164
LogisticRegression	0.93146	0.93506	0.93146	0.93128	0.02583
ExtraTreesClassifier	0.92097	0.94009	0.92097	0.91877	0.3256
LinearDiscriminantAnalysis	0.91168	0.91852	0.91168	0.91099	0.08204
RidgeClassifier	0.91166	0.91849	0.91166	0.91097	0.0027
BernoulliNB	0.91	0.91663	0.91	0.90967	0.00568
LinearSVC	0.90922	0.92107	0.90922	0.908	1.79701
RandomForestClassifier	0.90917	0.9343	0.90917	0.9053	0.4545
MLPClassifier	0.86636	0.89916	0.86636	0.8606	6.19948
SVC	0.83126	0.85979	0.83126	0.82493	13.85032
ComplementNB	0.809	0.81631	0.809	0.80505	0.00412
MultinomialNB	0.809	0.81631	0.809	0.80505	0.00044
NuSVC	0.77308	0.78664	0.77308	0.76672	17.01578
KNeighborsClassifier	0.58847	0.7292	0.58847	0.51118	0.00434
QuadraticDiscriminantAnalysis	0.5	0.25	0.5	0.33333	0.00956
DummyClassifier	0.5	0.25	0.5	0.33333	0.00125
NearestCentroid	0.46608	0.41852	0.46608	0.42299	0.00348

confirmada pelos resultados das métricas Precisão, Recuperação e F1, que tiveram os resultados 99.87%, 99.87% e 99.87%, respectivamente. Estes resultados confirmam que o modelo identifica com uma alta taxa de acerto os casos de *turnover/attrition*.

Os cinco melhores resultados foram das técnicas *Decision Tree Classifier, Gradient Boosting Classifier, Hist Gradient Boosting Classifier, XGB Classifier e LGBM Classifier*, sendo que esta última teve um resultado de 98.72% na Acurácia Balanceada.

Os resultados dos classificadores variaram entre 46.60% e 99.87%, sendo que a técnica com melhor resultado também teve um dos melhores tempos de execução.

Os parâmetros utilizados pela técnica com melhor resultado para esta base de dados estão apresentados na lista abaixo:

1. *DecisionTreeClassifier*
   - *criterion='gini'*
   - *splitter='best'*
   - *max_depth=5*
   - *min_samples_split=2*
   - *min_samples_leaf=1*

- *min_weight_fraction_leaf=0.0*
- *max_features=None*
- *random_state=42*
- *max_leaf_nodes=None*
- *min_impurity_decrease=0.0*
- *class_weight=None*
- *ccp_alpha=0.0*

4.3 Resultados Conjuntos de Classificadores Otimizados

Nesta seção, serão apresentados e discutidos os resultados obtidos por meio da aplicação dos métodos *ensemble*, *Stacking* e *Voting* em cada um dos conjuntos de dados utilizados nesta pesquisa. Esta análise abrangente visa fornecer uma compreensão aprofundada do desempenho dessas abordagens em diferentes conjuntos de dados, possibilitando uma avaliação comparativa e, assim, permitindo *insights* valiosos sobre a eficácia dessas técnicas em contextos específicos. A exploração meticulosa dos resultados proporcionará uma visão mais clara e abrangente das contribuições e limitações de cada método, contribuindo para uma compreensão mais refinada do impacto dessas estratégias na resolução dos problemas abordados pelas bases de dados em questão.

4.3.1 HR Comma Sep

Nas Tabelas 11 e 12 são apresentados os resultados obtidos com a aplicação do Stacking e do Voting, respectivamente. No caso do Stacking, é possível perceber que a combinação das técnicas *Linear SVC,Ridge Classifier,XGB Classifier,Extra Tree Classifier* utilizando a técnica de saída *Linear SVC* gerou uma pontuação de 97.83% na Acurácia Balanceada, 97.85% na Precisão, 97.84% na Recuperação e na F1.

A variação entre os agrupamentos de técnicas utilizando o *Stacking Classifier* foi de 69.34% a 97.84%. Vale destacar também que os cinco melhores resultados do *Stacking* ficaram em 97%, variando em casas decimais, confirmados por todas as métricas.

Os parâmetros utilizados pelo conjunto de técnicas utilizando o *Stacking Classifier* com melhor resultado utilizando para esta base de dados estão apresentados na lista abaixo:

1. Conjunto de técnicas:
   a) *LinearSVC*
      - *penalty='l2'*

Tabela 11 – Resultado do conjunto de classificadores Stacking para a base de dados HR Comma Sep

Técnicas	Técnica de saída	Acurácia Balanceada	Precisão	Recuperação	F1	Tempo
LinearSVC,RidgeClassifier,XGBClassifier,ExtraTreeClassifier	LinearSVC	0.97839	0.97846	0.97839	0.97839	0.82992
XGBClassifier,GaussianNB,SVC,QuadraticDiscriminantAnalysis	RidgeClassifier	0.97817	0.97822	0.97817	0.97817	0.59555
LinearSVC,SVC,KNeighborsClassifier,LGBMClassifier	QuadraticDiscriminantAnalysis	0.9776	0.97781	0.9776	0.9776	0.57659
QuadraticDiscriminantAnalysis,RidgeClassifier,LogisticRegression,XGBClassifier	LGBMClassifier	0.9769	0.97696	0.9769	0.9769	0.80831
LGBMClassifier,GaussianNB,NuSVC,QuadraticDiscriminantAnalysis,MLPClassifier	LogisticRegression	0.97506	0.97516	0.97506	0.97506	6.98913
LGBMClassifier,MLPClassifier,LGBMClassifier	ExtraTreeClassifier	0.96754	0.96759	0.96754	0.96754	6.19627
RidgeClassifier,LinearDiscriminantAnalysis,LGBMClassifier	ExtraTreeClassifier	0.96474	0.96493	0.96474	0.96473	0.36301
XGBRFClassifier,ExtraTreeClassifier,RidgeClassifier,ExtraTreeClassifier	RidgeClassifier	0.9601	0.96042	0.9601	0.96009	0.7409
LinearDiscriminantAnalysis,KNeighborsClassifier,GaussianNB	LGBMClassifier	0.95874	0.9589	0.95874	0.95874	0.28756
XGBRFClassifier,NuSVC	LGBMClassifier	0.9552	0.95623	0.9552	0.95517	6.71864
MLPClassifier,NuSVC,NuSVC,QuadraticDiscriminantAnalysis	LinearDiscriminantAnalysis	0.90637	0.90796	0.90637	0.90627	12.39344
LinearSVC,QuadraticDiscriminantAnalysis	DecisionTreeClassifier	0.89858	0.90982	0.89858	0.89724	0.40369
BernoulliNB,ExtraTreeClassifier,MLPClassifier	BernoulliNB	0.86481	0.88566	0.86481	0.86294	6.43574
SVC,GaussianNB	ExtraTreeClassifier	0.80338	0.8068	0.80338	0.80289	0.74222
RidgeClassifier,BernoulliNB	XGBRFClassifier	0.79095	0.79982	0.79095	0.78925	0.01304
LogisticRegression,LinearSVC,RidgeClassifier	BernoulliNB	0.7535	0.76046	0.7535	0.75203	0.36707
LGBMClassifier,QuadraticDiscriminantAnalysis,XGBClassifier,SVC,LinearDiscriminantAnalysis	BernoulliNB	0.69343	0.69406	0.69343	0.69312	0.96229

- *loss='squared_hinge'*
- *dual='warn'*
- *tol=0.0001*
- *C=1.0*
- *multi_class='ovr'*
- *fit_intercept=True*
- *intercept_scaling=1*
- *class_weight=None*
- *random_state=42*
- *max_iter=1000*

b) *RidgeClassifier*

- *alpha=1.0*
- *fit_intercept=True*
- *copy_X=True*
- *max_iter=None*
- *tol=0.0001*
- *class_weight=None*
- *solver='auto'*
- *positive=False*
- *random_state=42*

c) *XGBClassifier*

- *objective='binary logistic'*
- *max_depth=3*
- *learning_rate=0.1*
- *n_estimators=100*
- *silent=True*

- *booster='gbtree'*
- *n_jobs=1*
- *nthread=None*
- *gamma=0*
- *min_child_weight=1*
- *max_delta_step=0*
- *subsample=1*
- *colsample_bytree=1*
- *colsample_bylevel=1*
- *reg_alpha=0*
- *reg_lambda=1*
- *scale_pos_weight=1*
- *base_score=0.5*
- *random_state=42*
- *seed=None*
- *missing=None*

d) *ExtraTreeClassifier*

- *criterion='gini'*
- *splitter='random'*
- *max_depth=None*
- *min_samples_split=2*
- *min_samples_leaf=1*
- *min_weight_fraction_leaf=0.0*
- *max_features='sqrt'*
- *random_state=42*
- *max_leaf_nodes=None*
- *min_impurity_decrease=0.0,*
- *class_weight=None*
- *ccp_alpha=0.0*

2. Técnica de saída:

a) *LinearSVC*

- *penalty='l2'*
- *loss='squared_hinge'*
- *dual='warn'*

- *tol=0.0001*
- *C=1.0*
- *multi_class='ovr'*
- *fit_intercept=True*
- *intercept_scaling=1*
- *class_weight=None*
- *random_state=42*
- *max_iter=1000*

Já na utilização do Voting a combinação de técnicas mais bem pontuadas foi a combinação *Extra Tree Classifier,LGBM Classifier,Linear Discriminant Analysis* que apresentou uma pontuação de 95.91% na Acurácia Balanceada, confirmada pelas métricas Precisão, Recuperação e F1, que obtiveram, respectivamente, 95.92%, 95.91% e 95.91%.

A variação dos resultados com os agrupamentos do *Voting Classifier* foram de 67.43% a 95.91%, sendo que os cinco melhores agrupamentos tiveram resultados acima de 94.28%.

Tabela 12 – Resultado do método ensemble Voting de classificação para a base de dados HR Comma Sep

Técnicas	Acurácia Balanceada	Precisão	Recuperação	F1	Tempo
ExtraTreeClassifier,LGBMClassifier,LinearDiscriminantAnalysis	0.95914	0.95922	0.95914	0.95913	0.06516
GaussianNB,LGBMClassifier	0.95472	0.95684	0.95472	0.95466	0.00364
LGBMClassifier,KNeighborsClassifier,BernoulliNB	0.94706	0.9496	0.94706	0.94698	0.0035
MLPClassifier,KNeighborsClassifier	0.94658	0.94714	0.94658	0.94656	5.02955
KNeighborsClassifier,KNeighborsClassifier,GaussianNB,DecisionTreeClassifier	0.9429	0.94524	0.9429	0.94283	0.00109
DecisionTreeClassifier,KNeighborsClassifier,DecisionTreeClassifier,MLPClassifier,GaussianNB	0.9426	0.9432	0.9426	0.94258	4.96135
DecisionTreeClassifier,QuadraticDiscriminantAnalysis,NuSVC	0.93275	0.93284	0.93275	0.93275	1.21252
MLPClassifier,QuadraticDiscriminantAnalysis,LinearDiscriminantAnalysis,ExtraTreeClassifier	0.92291	0.9231	0.92291	0.9229	4.93453
MLPClassifier,MLPClassifier	0.90615	0.90828	0.90615	0.90601	10.73338
LinearDiscriminantAnalysis,LGBMClassifier	0.88213	0.90207	0.88213	0.88041	0.00398
NuSVC,LinearSVC,SVC,LGBMClassifier	0.86441	0.8806	0.86441	0.86284	1.9583
XGBClassifier,LinearDiscriminantAnalysis,RidgeClassifier,BernoulliNB	0.78207	0.78501	0.78207	0.78167	0.00916
LinearDiscriminantAnalysis,GaussianNB,LinearDiscriminantAnalysis,GaussianNB	0.76571	0.76932	0.76571	0.76514	0.01202
RidgeClassifier,LinearDiscriminantAnalysis	0.75354	0.7605	0.75354	0.75207	0.00362
LinearSVC,LinearSVC,SVC,BernoulliNB	0.72121	0.7427	0.72121	0.71437	0.15333
QuadraticDiscriminantAnalysis,GaussianNB,GaussianNB	0.6779	0.75399	0.6779	0.64893	0.00302
LinearSVC,LinearSVC	0.67431	0.72976	0.67431	0.64625	0.19136

Os parâmetros utilizados pelo conjunto de técnicas utilizando o *Voting Classifier* com melhor resultado utilizando para esta base de dados estão apresentados na lista abaixo:

1. Conjunto de técnicas:

   a) *ExtraTreeClassifier*

   - *criterion='gini'*
   - *splitter='random'*
   - *max_depth=None*
   - *min_samples_split=2*

- *min_samples_leaf=1*
- *min_weight_fraction_leaf=0.0*
- *max_features='sqrt'*
- *random_state=42*
- *max_leaf_nodes=None*
- *min_impurity_decrease=0.0,*
- *class_weight=None*
- *ccp_alpha=0.0*

b) *LGBMClassifier*

- *boosting_type='gbdt'*
- *num_leaves=31*
- *max_depth=-5*
- *learning_rate=0.09*
- *n_estimators=100*
- *subsample_for_bin=200000*
- *objective=None*
- *class_weight=None*
- *min_split_gain=0.0*
- *min_child_weight=0.001*
- *min_child_samples=20*
- *subsample=1.0*
- *subsample_freq=0*
- *colsample_bytree=1.0*
- *reg_alpha=0.0*
- *reg_lambda=0.0*
- *random_state=42*
- *n_jobs=None*
- *importance_type='split'*

c) *LinearDiscriminantAnalysis*

- *solver='svd'*
- *shrinkage=None*
- *priors=None*
- *n_components=None*
- *store_covariance=False*
- *tol=0.0001*

Tabela 13 – Resultado do método ensemble Stacking de classificação para a base de dados IBM Employee Attrition

Técnicas	Técnica de saída	Acurácia Balanceada	Precisão	Recuperação	F1	Tempo
XGBClassifier,RidgeClassifier,LinearDiscriminantAnalysis,BernoulliNB,GaussianNB	DecisionTreeClassifier	0.78963	0.7964	0.78957	0.78638	0.07002
DecisionTreeClassifier,LinearDiscriminantAnalysis	LinearSVC	0.78193	0.79064	0.78187	0.77979	0.04328
KNeighborsClassifier,XGBClassifier,GaussianNB,LinearSVC,DecisionTreeClassifier	QuadraticDiscriminantAnalysis	0.75802	0.77483	0.75792	0.75367	0.89134
BernoulliNB,ExtraTreeClassifier,LogisticRegression,NuSVC	SVC	0.73769	0.74289	0.73764	0.73608	0.37094
GaussianNB,XGBRFClassifier,SVC	LinearSVC	0.72437	0.741	0.72427	0.71882	0.072
XGBRFClassifier,DecisionTreeClassifier	QuadraticDiscriminantAnalysis	0.72145	0.73616	0.72138	0.70395	0.83825
LGBMClassifier,DecisionTreeClassifier,DecisionTreeClassifier,ExtraTreeClassifier,BernoulliNB	RidgeClassifier	0.72026	0.72898	0.72023	0.71769	0.25976
KNeighborsClassifier,DecisionTreeClassifier	LogisticRegression	0.72021	0.72843	0.7202	0.71753	0.00249
MLPClassifier,GaussianNB	LinearDiscriminantAnalysis	0.7146	0.72601	0.71454	0.71149	0.65051
DecisionTreeClassifier,GaussianNB	GaussianNB	0.71423	0.72631	0.71414	0.70904	0.00935
NuSVC,LinearSVC,DecisionTreeClassifier,BernoulliNB	DecisionTreeClassifier	0.68106	0.69552	0.68092	0.67084	0.31702
BernoulliNB,LGBMClassifier	LGBMClassifier	0.67195	0.67791	0.67197	0.66936	0.01456
LogisticRegression,LinearSVC,XGBRFClassifier	XGBClassifier	0.66748	0.68085	0.66749	0.66118	0.65442
ExtraTreeClassifier,LGBMClassifier	ExtraTreeClassifier	0.66183	0.66461	0.66181	0.65971	0.26256
XGBRFClassifier,BernoulliNB,MLPClassifier,BernoulliNB	ExtraTreeClassifier	0.60463	0.60892	0.60461	0.59932	1.74552

- *covariance_estimator=None*

É possível perceber que, nesse caso, o *Stacking Classifier* teve um resultado melhor em comparação com o *Voting Classifier*.

4.3.2 IBM Employee Attrition

Os resultados obtidos pelo *Stacking* pode ser verificado na Tabela 13 com a combinação de técnicas *XGB Classifier, Ridge Classifier, Linear Discriminant Analysis, Bernoulli NB, Gaussian NB* com a técnica de saída *Decision Tree Classifier* teve uma pontuação de 78.96% na Acurácia Balanceada, 79.64% na Precisão, 78.96% na Recuperação e 78.64% na F1.

A variação dos resultados dos agrupamentos do *Stacking Classifier* foi de 60.46% a 78.96%. Já os cinco melhores conjuntos variaram de 72.43% a 78.96%.

Os parâmetros utilizados pelo conjunto de técnicas utilizando o *Stacking Classifier* com melhor resultado utilizando para esta base de dados estão apresentados na lista abaixo:

1. Conjunto de técnicas:

   a) *XGBClassifier*

   - *objective='binary logistic'*
   - *max_depth=3*
   - *learning_rate=0.1*
   - *n_estimators=100*
   - *silent=True*
   - *booster='gbtree'*
   - *n_jobs=1*
   - *nthread=None*
   - *gamma=0*

- *min_child_weight=1*
- *max_delta_step=0*
- *subsample=1*
- *colsample_bytree=1*
- *colsample_bylevel=1*
- *reg_alpha=0*
- *reg_lambda=1*
- *scale_pos_weight=1*
- *base_score=0.5*
- *random_state=42*
- *seed=None*
- *missing=None*

b) *RidgeClassifier*

- *alpha=1.0*
- *fit_intercept=True*
- *copy_X=True*
- *max_iter=None*
- *tol=0.0001*
- *class_weight=None*
- *solver='auto'*
- *positive=False*
- *random_state=42*

c) *LinearDiscriminantAnalysis*

- *solver='svd'*
- *shrinkage=None*
- *priors=None*
- *n_components=None*
- *store_covariance=False*
- *tol=0.0001*
- *covariance_estimator=None*

d) *BernoulliNB*

- *alpha=1.0*
- *force_alpha='warn'*
- *binarize=0.0*

- *class_prior=None*

e) *GaussianNB*

- *priors=None*
- *var_smoothing=1e09*

2. Técnica de saída:

a) *DecisionTreeClassifier*

- *criterion='gini'*
- *splitter='best'*
- *max_depth=5*
- *min_samples_split=2*
- *min_samples_leaf=1*
- *min_weight_fraction_leaf=0.0*
- *max_features=None*
- *random_state=42*
- *max_leaf_nodes=None*
- *min_impurity_decrease=0.0*
- *class_weight=None*
- *ccp_alpha=0.0*

A Tabela 14 apresenta os resultados do *Voting* com a presente base de dados e é possível verificar que a combinação de técnicas *Linear Discriminant Analysis,Quadratic Discriminant Analysis,Linear Discriminant Analysis,Decision Tree Classifier* apresentou uma pontuação de 79.65% na Acurácia Balanceada, 80.36% na Precisão, 79.65% na Recuperação e 79.43 na F1.

No caso do *Voting Classifier*, a variação de resultados foi de 47.84% a 79.65%. Já os cinco melhores conjuntos de técnicas foram de 75.88% a 79.65%, resultados confirmados pelas demais métricas, confirmando que o conjunto de técnicas é capaz de identificar casos de *turnover/attrition*.

Os parâmetros utilizados pelo conjunto de técnicas utilizando o *Voting Classifier* com melhor resultado utilizando para esta base de dados estão apresentados na lista abaixo:

1. Conjunto de técnicas:

a) *LinearDiscriminantAnalysis*

- *solver='svd'*

Tabela 14 – Resultado do método ensemble Voting de classificação para a base de dados IBM Employee Attrition

Técnicas	Acurácia Balanceada	Precisão	Recuperação	F1	Tempo
LinearDiscriminantAnalysis,QuadraticDiscriminantAnalysis,LinearDiscriminantAnalysis,DecisionTreeClassifier	0.79654	0.8036	0.79647	0.7943	0.04127
ExtraTreeClassifier,RidgeClassifier,LinearDiscriminantAnalysis,XGBRFClassifier,ExtraTreeClassifier	0.78312	0.80532	0.78306	0.77776	0.76316
RidgeClassifier,LinearDiscriminantAnalysis	0.78111	0.79	0.78105	0.77914	0.01192
QuadraticDiscriminantAnalysis,GaussianNB,KNeighborsClassifier	0.7657	0.7863	0.76564	0.76193	0.00076
ExtraTreeClassifier,GaussianNB	0.7588	0.7699	0.75875	0.75436	0.00333
BernoulliNB,RidgeClassifier	0.74907	0.7711	0.74901	0.7413	0.0042
XGBRFClassifier,GaussianNB	0.7361	0.74452	0.73603	0.73192	0.0125
RidgeClassifier,KNeighborsClassifier,LogisticRegression	0.7263	0.74656	0.72631	0.72018	0.00343
RidgeClassifier,KNeighborsClassifier,NuSVC	0.71214	0.74159	0.71211	0.70358	0.01722
QuadraticDiscriminantAnalysis,SVC,XGBRFClassifier,XGBClassifier,BernoulliNB	0.70524	0.74084	0.70518	0.68778	0.0549
DecisionTreeClassifier,BernoulliNB	0.7024	0.71435	0.70237	0.69781	0.01237
KNeighborsClassifier,GaussianNB,XGBClassifier	0.67972	0.74859	0.67964	0.65609	0.00959
LGBMClassifier,XGBRFClassifier,ExtraTreeClassifier	0.65617	0.69643	0.65612	0.62063	0.01104
LinearSVC,LogisticRegression,XGBClassifier	0.58798	0.6077	0.58802	0.55649	0.01271
LogisticRegression,NuSVC,LGBMClassifier	0.58278	0.61173	0.58277	0.55645	0.01445
DecisionTreeClassifier,LinearSVC	0.47846	0.46091	0.47849	0.35828	0.01048

- *shrinkage=None*
- *priors=None*
- *n_components=None*
- *store_covariance=False*
- *tol=0.0001*
- *covariance_estimator=None*

b) *QuadraticDiscriminantAnalysis*

- *priors=None*
- *reg_param=0.0*
- *store_covariance=False*
- *tol=0.0001*

c) *LinearDiscriminantAnalysis*

- *solver='svd'*
- *shrinkage=None*
- *priors=None*
- *n_components=None*
- *store_covariance=False*
- *tol=0.0001*
- *covariance_estimator=None*

d) *DecisionTreeClassifier*

- *criterion='gini'*
- *splitter='best'*
- *max_depth=5*
- *min_samples_split=2*
- *min_samples_leaf=1*
- *min_weight_fraction_leaf=0.0*

Tabela 15 – Resultado do método ensemble Stacking de classificação para a base de dados Turnover

Técnicas	Técnica de saída	Acurácia Balanceada	Precisão	Recuperação	F1	Tempo
XGBClassifier,QuadraticDiscriminantAnalysis,SVC,LGBMClassifier	LinearSVC	0.57967	0.57999	0.57966	0.57928	0.0801
XGBRFClassifier,DecisionTreeClassifier,LGBMClassifier	MLPClassifier	0.57967	0.57997	0.57965	0.57933	0.36991
LGBMClassifier,LogisticRegression,GaussianNB	SVC	0.57614	0.57749	0.57616	0.57501	0.24433
XGBClassifier,GaussianNB,LogisticRegression,RidgeClassifier	SVC	0.57083	0.57104	0.57087	0.57056	0.78845
LinearSVC,LogisticRegression,ExtraTreeClassifier,NuSVC,LogisticRegression	LinearSVC	0.55354	0.55453	0.55337	0.55102	0.02526
XGBRFClassifier,NuSVC	LGBMClassifier	0.55083	0.55093	0.55078	0.55052	0.04958
XGBRFClassifier,GaussianNB,DecisionTreeClassifier,BernoulliNB	LinearDiscriminantAnalysis	0.53851	0.5389	0.53854	0.53717	0.04079
RidgeClassifier,MLPClassifier,XGBClassifier,GaussianNB	XGBClassifier	0.53333	0.53353	0.53325	0.53253	0.92645
GaussianNB,LinearSVC,ExtraTreeClassifier	GaussianNB	0.53241	0.53317	0.53237	0.52944	0.07471
KNeighborsClassifier,DecisionTreeClassifier,LinearSVC	RidgeClassifier	0.53235	0.53279	0.53234	0.53024	0.01503
ExtraTreeClassifier,SVC	MLPClassifier	0.52441	0.52435	0.52446	0.52286	0.46962
QuadraticDiscriminantAnalysis,LinearDiscriminantAnalysis	SVC	0.50633	0.50905	0.50613	0.49655	0.01785
LGBMClassifier,LinearDiscriminantAnalysis	BernoulliNB	0.5	0.24912	0.49912	0.33236	0.02201

- *max_features=None*
- *random_state=42*
- *max_leaf_nodes=None*
- *min_impurity_decrease=0.0*
- *class_weight=None*
- *ccp_alpha=0.0*

Nessa aplicação, a classificação realizada pelo *Voting Classifier* foi pouco melhor que a classificação realizada pelo *Stacking Classifier*.

4.3.3 Turnover

Os resultados do *Stacking* na base de dados *Turnover*, apresentados na Tabela 15, foram diferentes dos resultados obtidos com o *Stacking* nos demais conjuntos de dados. A combinação das técnicas *XGB Classifier, Quadratic Discriminant Analysis, SVC,LGBM Classifier* com a técnica de saída *Linear SVC* apresentou uma pontuação de 57.96% na Acurácia Balanceada, 58% na Precisão, 57.97% na Recuperação e 57.93% na F1. Novamente os resultados não foram tão significativos se comparados com a aplicação das mesmas técnicas nas demais bases de dados.

A variação de resultados dos conjuntos de técnicas do *Stacking Classifier* foi de 50% a 57.96%, sendo que das cinco primeiras técnicas, as quatro primeiras ficaram em 57%, com variações apenas nas casas decimais e a quinta melhor técnica ficou em 55.35%.

Os parâmetros utilizados pelo conjunto de técnicas utilizando o *Stacking Classifier* com melhor resultado utilizando para esta base de dados estão apresentados na lista abaixo:

1. Conjunto de técnicas:

   a) *XGBClassifier*

   - *objective='binary logistic'*

- *max_depth=3*
- *learning_rate=0.1*
- *n_estimators=100*
- *silent=True*
- *booster='gbtree'*
- *n_jobs=1*
- *nthread=None*
- *gamma=0*
- *min_child_weight=1*
- *max_delta_step=0*
- *subsample=1*
- *colsample_bytree=1*
- *colsample_bylevel=1*
- *reg_alpha=0*
- *reg_lambda=1*
- *scale_pos_weight=1*
- *base_score=0.5*
- *random_state=42*
- *seed=None*
- *missing=None*

b) *QuadraticDiscriminantAnalysis*

- *priors=None*
- *reg_param=0.0*
- *store_covariance=False*
- *tol=0.0001*

c) *SVC*

- *C=1.0*
- *kernel='rbf'*
- *degree=3*
- *gamma='scale'*
- *coef0=0.0*
- *shrinking=True*
- *probability=False*
- *tol=0.001*
- *cache_size=200*

- *class_weight=None*
- *max_iter=-1*
- *decision_function_shape='ovr'*
- *break_ties=False*
- *random_state=42*

d) *LGBMClassifier*

- *boosting_type='gbdt'*
- *num_leaves=31*
- *max_depth=-5*
- *learning_rate=0.09*
- *n_estimators=100*
- *subsample_for_bin=200000*
- *objective=None*
- *class_weight=None*
- *min_split_gain=0.0*
- *min_child_weight=0.001*
- *min_child_samples=20*
- *subsample=1.0*
- *subsample_freq=0*
- *colsample_bytree=1.0*
- *reg_alpha=0.0*
- *reg_lambda=0.0*
- *random_state=42*
- *n_jobs=None*
- *importance_type='split'*

2. Técnica de saída:

a) *LinearSVC*

- *penalty='l2'*
- *loss='squared_hinge'*
- *dual='warn'*
- *tol=0.0001*
- *C=1.0*
- *multi_class='ovr'*
- *fit_intercept=True*

- *intercept_scaling=1*
- *class_weight=None*
- *random_state=42*
- *max_iter=1000*

Já a Tabela 16 apresenta os resultados do *Voting* onde a combinação *XGB Classifier, XGB Classifier* teve uma pontuação de 58.40% na métrica Acurácia Balanceada, 58.42% na Precisão, 58.40% na Recuperação e 58.38% na F1.

Nesse caso, os resultados variaram entre 52.25% e 58.40%, sendo que os cinco conjuntos com melhores resultados ficaram entre 55.07% e 58.40%.

Tabela 16 – Resultado do método ensemble Voting de classificação para a base de dados Turnover

Técnicas	Acurácia Balanceada	Precisão	Recuperação	F1	Tempo
XGBClassifier,XGBClassifier	0.58401	0.58415	0.58402	0.5838	0.01528
LogisticRegression,LGBMClassifier,XGBRFClassifier	0.57788	0.57801	0.5779	0.57762	0.0154
XGBClassifier,RidgeClassifier,XGBClassifier,XGBRFClassifier,SVC	0.57436	0.5746	0.5744	0.57368	0.02326
KNeighborsClassifier,RidgeClassifier,LogisticRegression,XGBClassifier,LGBMClassifier	0.56477	0.56487	0.56477	0.56464	0.0088
LGBMClassifier,KNeighborsClassifier	0.55077	0.55978	0.55075	0.53237	0.01266
XGBClassifier,LogisticRegression,KNeighborsClassifier,KNeighborsClassifier	0.54638	0.54942	0.54635	0.53887	0.01068
GaussianNB,QuadraticDiscriminantAnalysis	0.54022	0.54262	0.54028	0.53185	0.00136
XGBRFClassifier,LogisticRegression,RidgeClassifier,XGBRFClassifier,SVC	0.54015	0.54016	0.54024	0.53965	0.02426
MLPClassifier,SVC,GaussianNB,LogisticRegression	0.53932	0.54013	0.53938	0.53382	0.32504
SVC,ExtraTreeClassifier,ExtraTreeClassifier	0.5385	0.53908	0.5385	0.53753	0.00153
LogisticRegression,GaussianNB,GaussianNB,LinearDiscriminantAnalysis	0.53578	0.53685	0.53589	0.52911	0.00448
RidgeClassifier,RidgeClassifier	0.5349	0.53534	0.53498	0.53411	0.00216
NuSVC,RidgeClassifier,LogisticRegression	0.5349	0.53533	0.53497	0.53418	0.00877
LinearDiscriminantAnalysis,XGBClassifier,LinearDiscriminantAnalysis	0.5349	0.53534	0.53498	0.53411	0.02787
LinearSVC,QuadraticDiscriminantAnalysis,DecisionTreeClassifier	0.53067	0.53755	0.53064	0.51419	0.02486
GaussianNB,BernoulliNB	0.52528	0.52886	0.52539	0.50676	0.00176

Os parâmetros utilizados pelo conjunto de técnicas utilizando o *Voting Classifier* com melhor resultado utilizando para esta base de dados estão apresentados na lista abaixo:

1. Conjunto de técnicas (2 técnicas idênticas):

   a) *XGBClassifier*

   - *objective='binary logistic'*
   - *max_depth=3*
   - *learning_rate=0.1*
   - *n_estimators=100*
   - *silent=True*
   - *booster='gbtree'*
   - *n_jobs=1*
   - *nthread=None*
   - *gamma=0*
   - *min_child_weight=1*

- *max_delta_step=0*
- *subsample=1*
- *colsample_bytree=1*
- *colsample_bylevel=1*
- *reg_alpha=0*
- *reg_lambda=1*
- *scale_pos_weight=1*
- *base_score=0.5*
- *random_state=42*
- *seed=None*
- *missing=None*

Embora o resultado tenha sido semelhante ao uso dos classificadores individuais apresentados na seção anterior, novamente o *Voting Classifier* apresentou um resultado pouco melhor que o *Stacking Classifier*.

4.3.4 Attrition

Para a base de dados *Attrition* foram obtidos os resultados apresentados na Tabela 17, onde é possível perceber que a combinação das técnicas *SVC, KNeighbors Classifier,Extra Tree Classifier, XGB Classifier, Bernoulli NB* com a técnica de saída sendo *Decision Tree Classifier* e foi obtida uma pontuação de 99.91% na métrica Acurácia Balanceada, 99.91% na métrica Precisão, 99.91% na métrica Recuperação, bem como o mesmo valor na métrica F1, o que confirma o bom resultado do conjunto de classificadores para esta base de dados.

Os resultados do *Stacking Classifier* variaram de 92.20% a 99.91% e os cinco melhores resultados obtiveram um resultado de 99%, variando apenas em casas decimais. Como desvantagem do *Stacking Classifier* nessa aplicação específica existe a questão do tempo de execução que é muito elevado pelo tamanho da base de dados.

Vale destacar o conjunto de técnicas *LGBM Classifier,LGBM Classifier,LGBM Classifier,KNeighbors Classifier,LGBM Classifier* com a técnica de saída sendo *KNeighbors Classifier* que teve um resultado de 99.84%, mas um tempo de execução bem inferior em comparação com o conjunto que apresentou o melhor resultado.

Os parâmetros utilizados pelo conjunto de técnicas utilizando o *Stacking Classifier* com melhor resultado utilizando para esta base de dados estão apresentados na lista abaixo:

1. Conjunto de técnicas:

Tabela 17 – Resultado do método ensemble Stacking de classificação para a base de dados Attrition

Técnicas	Técnica de saída	Acurácia Balanceada	Precisão	Recuperação	F1	Tempo
SVC,KNeighborsClassifier,ExtraTreeClassifier,XGBClassifier,BernoulliNB	DecisionTreeClassifier	0.99912	0.99912	0.99912	0.99912	62.72038
XGBClassifier,MLPClassifier,MLPClassifier	DecisionTreeClassifier	0.99852	0.99852	0.99852	0.99852	22.24524
LGBMClassifier,LGBMClassifier,LGBMClassifier,KNeighborsClassifier,LGBMClassifier	KNeighborsClassifier	0.99839	0.99839	0.99839	0.99839	1.50919
XGBRFClassifier,XGBClassifier,RidgeClassifier	SVC	0.99838	0.99839	0.99838	0.99838	12.16059
XGBClassifier,RidgeClassifier,KNeighborsClassifier	RidgeClassifier	0.99698	0.99702	0.99698	0.99698	2.20865
LinearDiscriminantAnalysis,LGBMClassifier,SVC,ExtraTreeClassifier,LogisticRegression	LinearSVC	0.99641	0.99648	0.99641	0.99641	68.77713
BernoulliNB,DecisionTreeClassifier,NuSVC	LGBMClassifier	0.9896	0.9899	0.9896	0.9896	51.81859
BernoulliNB,XGBRFClassifier	XGBRFClassifier	0.98937	0.98966	0.98937	0.98937	0.73751
ExtraTreeClassifier,LinearSVC	DecisionTreeClassifier	0.95968	0.96064	0.95968	0.95966	3.21001
KNeighborsClassifier,LogisticRegression	LogisticRegression	0.92834	0.9314	0.92834	0.92819	1.89529
NuSVC,LogisticRegression	KNeighborsClassifier	0.92478	0.92843	0.92478	0.9246	56.51826
SVC,LinearDiscriminantAnalysis,LogisticRegression	RidgeClassifier	0.92202	0.92707	0.92202	0.92167	54.01007

a) *SVC*

- *C=1.0*
- *kernel='rbf'*
- *degree=3*
- *gamma='scale'*
- *coef0=0.0*
- *shrinking=True*
- *probability=False*
- *tol=0.001*
- *cache_size=200*
- *class_weight=None*
- *max_iter=-1*
- *decision_function_shape='ovr'*
- *break_ties=False*
- *random_state=42*

b) *KNeighborsClassifier*

- *n_neighbors=5*
- *weights='uniform'*
- *algorithm='auto'*
- *leaf_size=30*
- *p=2*
- *metric='minkowski'*
- *metric_params=None*
- *random_state=42*
- *n_jobs=None*

c) *ExtraTreeClassifier*

- *criterion='gini'*

- *splitter='random'*
- *max_depth=None*
- *min_samples_split=2*
- *min_samples_leaf=1*
- *min_weight_fraction_leaf=0.0*
- *max_features='sqrt'*
- *random_state=42*
- *max_leaf_nodes=None*
- *min_impurity_decrease=0.0,*
- *class_weight=None*
- *ccp_alpha=0.0*

d) *XGBClassifier*

- *objective='binary logistic'*
- *max_depth=3*
- *learning_rate=0.1*
- *n_estimators=100*
- *silent=True*
- *booster='gbtree'*
- *n_jobs=1*
- *nthread=None*
- *gamma=0*
- *min_child_weight=1*
- *max_delta_step=0*
- *subsample=1*
- *colsample_bytree=1*
- *colsample_bylevel=1*
- *reg_alpha=0*
- *reg_lambda=1*
- *scale_pos_weight=1*
- *base_score=0.5*
- *random_state=42*
- *seed=None*
- *missing=None*

e) *BernoulliNB*

- *alpha=1.0*

Tabela 18 – Resultado do método ensemble Voting de classificação para a base de dados Attrition

Técnicas	Acurácia Balanceada	Precisão	Recuperação	F1	Tempo
XGBRFClassifier,LinearSVC,QuadraticDiscriminantAnalysis,LGBMClassifier	0.98673	0.9876	0.98673	0.98671	1.32898
LGBMClassifier,LinearDiscriminantAnalysis,XGBClassifier	0.98	0.9815	0.98	0.97996	0.02097
LogisticRegression,MLPClassifier,XGBClassifier,LinearSVC,MLPClassifier	0.96373	0.96686	0.96373	0.96362	20.31532
NuSVC,XGBClassifier,LinearDiscriminantAnalysis,ExtraTreeClassifier	0.96331	0.96481	0.96331	0.96326	43.15471
KNeighborsClassifier,LinearDiscriminantAnalysis,BernoulliNB,DecisionTreeClassifier	0.95612	0.95949	0.95612	0.95601	0.12197
LinearSVC,XGBRFClassifier	0.94942	0.95193	0.94942	0.94934	2.96463
DecisionTreeClassifier,LinearDiscriminantAnalysis,LinearDiscriminantAnalysis,LogisticRegression	0.9385	0.94117	0.9385	0.9384	0.02259
GaussianNB,MLPClassifier,SVC,KNeighborsClassifier	0.93788	0.94677	0.93788	0.93718	11.96697
LogisticRegression,LogisticRegression,ExtraTreeClassifier	0.93116	0.93466	0.93116	0.93098	0.05153
BernoulliNB,MLPClassifier	0.9299	0.93513	0.9299	0.92965	9.84513
DecisionTreeClassifier,LinearSVC,LogisticRegression,LinearSVC	0.92437	0.92818	0.92437	0.92417	2.65542
NuSVC,RidgeClassifier,DecisionTreeClassifier,LinearDiscriminantAnalysis,QuadraticDiscriminantAnalysis	0.92133	0.93104	0.92133	0.92038	28.49478
LinearSVC,LinearSVC	0.91155	0.91686	0.91155	0.91128	1.84821

- *force_alpha='warn'*
- *binarize=0.0*
- *class_prior=None*

2. Técnica de saída:

a) *DecisionTreeClassifier*

- *criterion='gini'*
- *splitter='best'*
- *max_depth=5*
- *min_samples_split=2*
- *min_samples_leaf=1*
- *min_weight_fraction_leaf=0.0*
- *max_features=None*
- *random_state=42*
- *max_leaf_nodes=None*
- *min_impurity_decrease=0.0*
- *class_weight=None*
- *ccp_alpha=0.0*

A Tabela 18 apresenta os resultados da execução da técnica *ensemble Voting*, onde é possível verificar que a combinação de técnicas *XGB RF Classifier, Linear SVC,Quadratic Discriminant Analysis, LGBM Classifier* obteve um resultado de 98.67% na métrica Acurácia Balanceada, 98.76% na métrica Precisão, 98.67% na métrica Recuperação e 98.67% na métrica F1. Este resultado é um pouco menor se comparado com o resultado da aplicação da técnica *Stacking*, mas ainda é um bom resultado.

Já para o *Voting Classifier* a variação de resultados foi de 91.15% a 98.67% e os cinco melhores conjuntos de técnicas tiveram seus resultados variando de 95.61% a 98.67%, confirmado por todas as métricas.

Os parâmetros utilizados pelo conjunto de técnicas utilizando o *Voting Classifier* com melhor resultado utilizando para esta base de dados estão apresentados na lista abaixo:

1. Conjunto de técnicas:
    a) *XGBRFClassifier*
        - *n_estimators=100*
        - *learning_rate=1*
        - *colsample_bynode=0.8*
        - *subsample=0.9*
        - *booster='gbtree'*
    b) *LinearSVC*
        - *penalty='l2'*
        - *loss='squared_hinge'*
        - *dual='warn'*
        - *tol=0.0001*
        - *C=1.0*
        - *multi_class='ovr'*
        - *fit_intercept=True*
        - *intercept_scaling=1*
        - *class_weight=None*
        - *random_state=42*
        - *max_iter=1000*
    c) *QuadraticDiscriminantAnalysis*
        - *priors=None*
        - *reg_param=0.0*
        - *store_covariance=False*
        - *tol=0.0001*
    d) *LGBMClassifier*
        - *boosting_type='gbdt'*
        - *num_leaves=31*
        - *max_depth=-5*
        - *learning_rate=0.09*
        - *n_estimators=100*

- *subsample_for_bin=200000*
- *objective=None*
- *class_weight=None*
- *min_split_gain=0.0*
- *min_child_weight=0.001*
- *min_child_samples=20*
- *subsample=1.0*
- *subsample_freq=0*
- *colsample_bytree=1.0*
- *reg_alpha=0.0*
- *reg_lambda=0.0*
- *random_state=42*
- *n_jobs=None*
- *importance_type='split'*

O conjunto de técnicas do *Stacking Classifier* teve resultado um pouco melhores em comparação com os conjuntos do *Voting Classifier*, no entanto o tempo de execução do primeiro é muito superior e, vale destacar que o tempo gasto para executar o conjunto de técnicas não compensa a diferença de resultados. Então neste caso, o *Voting Classifier* apresentou um conjunto com resultados melhores.

4.4 Resultados Tree-based Pipeline Optimization Tool (TPOT)

Os resultados provenientes da aplicação da biblioteca *Tree-based Pipeline Optimization Tool (TPOT)* foram compilados e são agora apresentados de maneira detalhada na Tabela 19. O que se destaca é a notável melhoria em todos os cenários analisados em comparação com as técnicas previamente discutidas. Esse aprimoramento é particularmente evidente ao examinarmos o conjunto de dados "Turnover", onde os resultados obtidos superam significativamente aqueles alcançados pelas demais abordagens.

A aplicação do TPOT, uma ferramenta de otimização de pipelines baseados em árvores, parece trazer um ganho substancial em termos de desempenho preditivo. Esses resultados encorajadores indicam que a abordagem automatizada do TPOT na busca e configuração de pipelines eficazes pode ser uma estratégia promissora para aprimorar os modelos de predição em diversos contextos.

A próxima etapa será uma análise mais detalhada desses resultados em comparação com as demais abordagens já executadas. Essa comparação proporcionará uma

Tabela 19 – Resultado do uso do Tree-based Pipeline Optimization Tool (TPOT)

Resultados Tree-based Pipeline Optimization Tool (TPOT)					
Base de dados	**Acurácia Balanceada**	**Precisão**	**Recuperação**	**F1**	**Tempo**
HR Comma Sep	0.997112	0.997113	0.997112	0.997112	1260
IBM Employee Attrition	1.0	1.0	1.0	1.0	360
Turnover	0.997373	0.996503	0.998249	0.997375	120
Attrition	0.99945	0.999149	0.999751	0.99945	5760

compreensão mais completa da eficácia e potencial do TPOT na tarefa de predição para os conjuntos de dados em questão.

Para a base de dados HR Comma Sep, o *pipeline* com os melhores parâmetros está apresentado na lista a seguir:

1. *Pipeline*:
    a) *StackingEstimator*:
        i. *RandomForestClassifier*
            - *n_estimators=100*
            - *min_samples_split=4*
            - *min_samples_leaf=3*
            - *max_features=0.5*
            - *criterion="entropy"*
            - *bootstrap=True*
    b) *StackingEstimator*:
        i. *XGBClassifier*
            - *learning_rate=0.01*
            - *max_depth=5*
            - *min_child_weight=16*
            - *n_estimators=100*
            - *n_jobs=1*
            - *subsample=0.8500000000000001*
    c) *XGBClassifier*
        - *learning_rate=0.01*
        - *max_depth=4*
        - *min_child_weight=6*
        - *n_estimators=100*
        - *n_jobs=1*
        - *subsample=0.9000000000000001*

Para a base de dados *IBM Employee Attrition*, o *pipeline* com os melhores parâmetros está apresentado na lista a seguir:

1. *Pipeline*:
    a) *StackingEstimator*:
        i. *XGBClassifier*
            - *learning_rate=0.01*
            - *max_depth=7*
            - *min_child_weight=1*
            - *n_estimators=100*
            - *n_jobs=1*
            - *subsample=0.55*
    b) *StackingEstimator*:
        i. *SGDClassifier*
            - *alpha=0.0*
            - *eta0=0.1*
            - *fit_intercept=True*
            - *learning_rate="invscaling"*
            - *loss="squared_hinge"*
            - *penalty="elasticnet"*
            - *power_t=50.0*
    c) *GradientBoostingClassifier*
        - *learning_rate=0.01*
        - *max_depth=9*
        - *max_features=0.25*
        - *min_samples_leaf=16*
        - *min_samples_split=12*
        - *n_estimators=100*
        - *subsample=0.3*

Para a base de dados *Turnover*, o *pipeline* com os melhores parâmetros está apresentado na lista a seguir:

1. *Pipeline*:
    a) *StackingEstimator*:

i. *XGBClassifier*

- *learning_rate=0.01*
- *max_depth=4*
- *min_child_weight=1*
- *n_estimators=100*
- *n_jobs=1*
- *subsample=0.9000000000000001*

b) *StackingEstimator*:

i. *DecisionTreeClassifier*

- *criterion="gini"*
- *max_depth=2*
- *min_samples_leaf=4*
- *min_samples_split=12*

c) *GradientBoostingClassifier*

- *learning_rate=0.01*
- *max_depth=9*
- *max_features=0.4*
- *min_samples_leaf=2*
- *min_samples_split=8*
- *n_estimators=100*
- *subsample=0.7500000000000001*

Já para a base de dados *Attrition*, o *pipeline* com os melhores parâmetros está apresentado na lista a seguir:

1. *Pipeline*:

a) *StackingEstimator*:

i. *XGBClassifier*

- *learning_rate=0.001*
- *max_depth=5*
- *min_child_weight=10*
- *n_estimators=100*
- *n_jobs=1*
- *subsample=0.8*

b) *StackingEstimator*:

i. *GaussianNB*

- *priors=None*
- *var_smoothing=1e09*

c) *GradientBoostingClassifier*

- *learning_rate=0.01*
- *max_depth=1*
- *max_features=0.8500000000000001*
- *min_samples_leaf=1*
- *min_samples_split=4*
- *n_estimators=100*
- *subsample=0.9000000000000001*

O TPOT foi executado com os seguintes parâmetros:

- *cv=4*
- *early_stop=12*
- *generations=5*
- *offspring_size=12*
- *population_size=24*
- *random_state=42*
- *template='Classifier-Classifier-Classifier'*

O parâmetro *template* define a quantidade de classificadores que serão utilizados no *pipeline* e, no caso da pesquisa, foram utilizados três classificadores em cada *pipeline*.

A Tabela 20 apresenta de maneira abrangente os resultados mais destacados de cada técnica, além de incluir as performances alcançadas pelo TPOT. Ao analisar os dados, torna-se evidente que as técnicas de classificação utilizadas de forma isolada destacam-se como as mais eficazes, seguindo-se aos resultados gerados pelo TPOT. Entretanto, é crucial observar que, apesar de alcançarem performances notáveis, as técnicas de classificação demandaram um tempo de execução inferior em comparação com tempo utilizado pela biblioteca TPOT.

Essa observação levanta uma ponderação significativa sobre a escolha de abordagens. Enquanto a biblioteca TPOT demonstra um desempenho superior em termos de

acurácia, sua eficiência temporal pode ser um fator determinante na decisão de implementação prática, especialmente no ambiente desejado pela pesquisa.

Os resultados ressaltam a importância de considerar não apenas a acurácia dos modelos, mas também fatores como o tempo de execução, ao decidir a estratégia mais adequada para a aplicação prática na realidade da universidade. A evidência do destaque na tabela, indicada pelas linhas destacadas, simplifica a identificação dos modelos mais promissores, proporcionando uma base sólida para a tomada de decisões informadas na implementação futura.

Tabela 20 – Comparativo dos melhores resultados de cada técnica para cada base de dados

Comparativo de Resultados - melhor de cada aplicação					
Base de dados	**Acurácia Balanceada**	**Precisão**	**Recuperação**	**F1**	**Tempo**
HR Comma Sep Detector de Anomalia	0.57885	0.69443	0.69445	0.69442	5249
HR Comma Sep Classificador	**0.98119**	**0.98148**	**0.98119**	**0.98118**	**0.38411**
HR Comma Sep Stacking	0.97839	0.97846	0.97839	0.97839	0.82992
HR Comma Sep Voting	0.95914	0.95922	0.95914	0.95913	0.06516
HR Comma Sep TPOT	**0.997112**	**0.997113**	**0.997112**	**0.997112**	**1260**
IBM Employee Attrition Detector de Anomalia	0.52803	0.74551	0.75714	0.75112	0.183
IBM Employee Attrition Classificador	**0.90033**	**0.90795**	**0.90027**	**0.89928**	**0.01068**
IBM Employee Attrition Stacking	0.78963	0.7964	0.78957	0.78638	0.07002
IBM Employee Attrition Voting	0.79654	0.8036	0.79647	0.7943	0.04127
IBM Employee Attrition TPOT	**1.0**	**1.0**	**1.0**	**1.0**	**360**
Turnover Detector de Anomalia	0.5397	0.53988	0.53942	0.53916	0.06
Turnover Classificador	**0.59459**	**0.59516**	**0.59455**	**0.59396**	**0.00995**
Turnover Stacking	0.57967	0.57999	0.57966	0.57928	0.0801
Turnover Voting	0.58401	0.58415	0.58402	0.5838	0.01528
Turnover TPOT	**0.997373**	**0.996503**	**0.998249**	**0.997375**	**120**
Attrition Detector de Anomalia	0.92156	0.9909	0.9909	0.9909	3.048
Attrition Classificador	0.99869	0.9987	0.99869	0.99869	**0.00061**
Attrition Stacking	**0.99912**	**0.99912**	**0.99912**	**0.99912**	62.72038
Attrition Voting	0.98673	0.9876	0.98673	0.98671	1.32898
Attrition TPOT	0.99945	0.999149	0.999751	0.99945	5760

5 Conclusão

Esta pesquisa buscou identificar as aplicações da Inteligência Artificial (I.A.) no ambiente de gestão de recursos humanos, quais as áreas onde era possível aplicar essa nova tecnologia e dentro disso quais as técnicas utilizadas e seus resultados.

Diante dos diversos tópicos de aplicação da I.A. na gestão de RH, houve a concentração no tema de *turnover/attrition* com o objetivo de verificar quais abordagens de aprendizagem de máquina poderiam ser aplicadas nessa área específica. Foram realizados testes com as técnicas de cada abordagem e de acordo com os resultados obtidos, foi possível verificar se a aplicação seria viável ou não ao ambiente de gestão de recursos humanos de uma universidade pública.

Foram selecionados quatro bases de dados públicas, de diferentes dimensões para testar as aplicações das técnicas de aprendizado de máquina para identificação de casos de *turnover/attrition*. Testes foram iniciados com a aplicação de técnicas de aprendizado de máquina para detecção de anomalias/novidades. Neste caso, as técnicas utilizadas foram: *Elliptic Envelope, One Class SVM, pipeline - Nystroem e SGD One Class SVM, Isolation Forest* e o *Local Outlier Factor*.

Dando continuidade nos testes da pesquisa, as bases passaram por fatoração dos campos texto e foram aplicadas as técnicas de classificação: *Ridge Classifier*, *Logistic Regression*, *Dummy Classifier*, *Linear Discriminant Analysis*, *Quadratic Discriminant Analysis*, *AdaBoost Classifier*, *Bagging Classifier*, *Extra Trees Classifier*, *Gradient Boosting Classifier*, *RandomForestClassifier*, *Hist Gradient Boosting Classifier*, *Bernoulli NB*, *Categorical NB*, *Complement NB*, *Multinomial NB*, *Gaussian NB*, *K Neighbors Classifier*, *Radius Neighbors Classifier*, *Nearest Centroid*, *MLPClassifier*, *Linear SVC*, *Nu SVC*, *SVC*, *Decision Tree Classifier*, *Extra Tree Classifier*, *XGB Classifier*, *XGB RF Classifier*, *DaskLGBMClassifier* e *LGBM Classifier*. Dando continuidade na execução de classificadores, foram utilizados os métodos de conjuntos de classificadores *Stacking* e *Voting*, e por fim uma biblioteca pública que utiliza algoritmo genético para otimizar pipelines de classificadores do Sklearn, o *Tree-based Pipeline Optimization Tool (TPOT)*.

No geral as métricas indicaram uma variação na qualidade de acertos das abordagens utilizadas. A abordagem proposta obteve a melhor classificação na base de dados *Attrition* e terceira melhor no HR Comma Sep com diferença inferior de 3% de acerto. A abordagem com um otimização com algoritmo genético, o TPOT, apresenta melhores resultados, mas um tempo computacional superior a 1000% acima das demais técnicas, tornando-se inviável para grandes empresas. A abordagem de detecção de anomalias apresentou resultados inferiores em comparação aos demais quando utilizada em bases de da-

dos menores, o que demonstra que a aplicação desta abordagem é inviável para ambientes corporativos com uma menor quantidade de dados de funcionários.

É possível afirmar que a pesquisa atingiu, em parte, os objetivos delineados no 1, pois identificou áreas promissoras para a aplicação de Inteligência Artificial (I.A.) na gestão de recursos humanos. Ao concentrar-se na detecção de *turnover/attrition*, a pesquisa mostrou ser capaz identificar este tipo de situação, fundamentando-se nos dados dos colaboradores mediante a aplicação de técnicas de aprendizado de máquina.

Embora haja sucesso em delinear as possíveis áreas de aplicação da I.A. e focar em uma área específica, é relevante observar que a plena realização dos objetivos propostos na 1 permanece parcial. No entanto, o enfoque na detecção de *turnover* representa um avanço significativo, revelando a capacidade da pesquisa em aplicar eficazmente a aprendizagem de máquina para identificar indicadores da ocorrência dessa situação entre os funcionários.

A pesquisa se mostrou viável ao gerar uma ferramenta de detecção de *turnover/attrition* desenvolvida em *python*, que faz uso de diferentes abordagens de aprendizado de máquina. A ferramenta citada está disponível no repositório da pesquisa Dias e Moraes(60).

Devido ao tempo de execução e uso de recursos, as técnicas de aprendizado profundo não foram utilizadas na pesquisa e devem ser utilizadas no identificação do problema de *turnover/attrition* em trabalhos futuros. De forma semelhante, embora a pesquisa tenha gerado bons resultados para a identificação dos casos de saída do trabalho, não foi possível adaptá-la para a identificação dos casos de insatisfação nesse momento, ficando essa adaptação para trabalhos futuros com a disponibilidade de melhores recursos computacionais.

5.1 Trabalhos Futuros

Como trabalhos futuros é recomendada a execução das técnicas de aprendizado profundo com as mesmas bases de dados para verificar o resultado que será obtido. Além disso é sugerida a execução otimizada do TPOT para tentar reduzir o tempo de execução.

Sobre as técnicas de detecção de anomalias/novidades, recomenda-se que seja testado em outras bases de dados reais tão grandes quanto a base "Attrition" para determinar a performance destas na identificação dos casos de *turnover/attrition* e confirmar se essa abordagem continua sendo inviável para o problema proposto.

Além disso, em uma análise prévia na estrutura dos dados dos servidores presentes no Sistema Integrado de Gestão (SIG) da universidade juntamente com os dados que a Pró-Reitoria de Gestão de Pessoas possui, existem a possibilidade de construção de uma base de dados semelhante à base HR Comma Sep e, considerando os resultados

obtidos com a base citada, existe uma expectativa de que o desempenho das abordagens de aprendizado de máquina também sejam promissores junto com os dados gerados na universidade.

Sugere-se, como continuidade da pesquisa, a elaboração de uma ferramenta de suporte ao RH utilizando esses dados extraídos do SIG para monitorar sinais que indicam a possibilidade de *turnover/attrition*, pois isso permitiria que o servidor fosse consultado com o objetivo de evitar seu desligamento ou simplesmente melhorar sua satisfação no trabalho.

Referências

1 SCIKIT-LEARN. *Scikit-learn*. 2023. Last accessed 25 September 2023. Disponível em: <https://scikit-learn.org/>. 9, 60

2 FERREIRA, M. R. de L. et al. Gestão de pessoas no setor público: um estudo dos níveis de conflito a partir da visão interacionista people management in the public sector: a study of levels of conflict from the perspective interactionist. p. 510–528, 2010. 13, 22

3 AFFONSO, L. M. F.; MARTINS, R. H. Fatores organizacionais que geram insatisfação no servidor público e comprometem a qualidade dos serviços prestados. 2010. 13, 23

4 CARVALHO, J. N. F. de; SILVA, A. de S. Motivação no setor público como ferramenta estratégica de gestão: desafios e reflexões. *Revista Gestão Políticas Públicas*, Universidade de Sao Paulo, Agencia USP de Gestao da Informacao Academica (AGUIA), v. 9, p. 306–321, 12 2019. 13, 14, 23

5 SILVA, D. D. S.; DINIZ, I. S. F. N.; PELLIZZONI, L. N. SatisfaÇÃo no trabalho em uma empresa pÚblica: Uma anÁlise explicativa com funcionÁrios do transporte pÚblico 1. *Revista Livre de Sustentabilidade e Empreendedorismo*, v. 5, p. 173–197, 2020. ISSN 2448-2889. 13, 21

6 CARVALHO, K. D. S.; FALCE, J. L. L.; GUIMARãES, L. D. V. M. A motivação de serviço público e a satisfação no trabalho: Pesquisa em uma universidade federal public service motivation and job satisfaction: Research at the federal university. *Amazônia, Organizações e Sustentabilidade*, Galoa Events Proceedings, v. 10, p. 77, 12 2021. 13, 23

7 ANDRADE, D. C. T. D. Engajamento no trabalho no serviço público: Um modelo multicultural. *Journal of Contemporary Administration*, v. 24, p. 49–76, 2020. Disponível em: <http://rac.anpad.org.br>. 13, 22

8 DESORDI, D.; BONA, C. D. A inteligência artificial e a eficiência na administração pública. *Revista de Direito*, Revista de Direito, v. 12, p. 01–22, 9 2020. ISSN 1806-8790. 14, 20

9 CEDRAZ, A.; ARRAES, A.; TCU. Levantamento inteligencia artificial - tcu. 2022. 15, 16

10 CHOWDHURY, S. et al. Unlocking the value of artificial intelligence in human resource management through ai capability framework. *Human Resource Management Review*, Elsevier Ltd, v. 33, 3 2023. ISSN 10534822. 15, 21

11 BIRZNIECE, I. et al. Predictive modeling of hr dynamics using machine learning. In: . [S.l.]: Association for Computing Machinery, 2022. p. 17–23. ISBN 9781450395748. 15, 26

12 SADANA, P.; MUNNURU, D. Machine learning model to predict work force attrition. In: . [S.l.]: Institute of Electrical and Electronics Engineers Inc., 2021. ISBN 9781728188768. 16, 22, 25, 26, 31

13 SHANKAR, R. S. et al. Analyzing attrition and performance of an employee using machine learning techniques. In: . [S.l.]: Institute of Electrical and Electronics Engineers Inc., 2021. p. 1601–1608. ISBN 9781665435246. 16, 22, 25, 26

14 JIN JIAXING SHANG, Q. Z. C. L. W. X. Z.; QIANG, B. *RFRSF: Employee Turnover Prediction Based on Random Forests and Survival Analysis*. [S.l.]: Systems Engineering – WISE 2020, 2020. ISBN 9783030620042. 16, 25

15 JOHNSON, B. A.; COGGBURN, J. D.; LLORENS, J. J. Artificial intelligence and public human resource management: Questions for research and practice. *Public Personnel Management*, SAGE Publications Inc., v. 51, p. 538–562, 12 2022. ISSN 19457421. 16, 21

16 FELIPE, S. L. *TECNOLOGIA APLICADA À GESTÃO DOS CONFLITOS NO ÂMBITO DO PODER JUDICIÁRIO BRASILEIRO*. [S.l.: s.n.], 2020. 112 p. 16

17 KUMAR, M. R. et al. Human resource management using machine learning-based solutions. In: . [S.l.]: Institute of Electrical and Electronics Engineers (IEEE), 2022. p. 801–806. ISBN 9781665479714. 21

18 PAMPOUKTSI, P. et al. Applied machine learning techniques on selection and positioning of human resources in the public sector. *Open Journal of Business and Management*, v. 09, p. 536–556, 2021. ISSN 2329-3284. 21, 26

19 MATHEW, V.; CHACKO, D. A. M.; UDHAYAKUMAR, D. A. *Prediction of suitable human resource for replacement in skilled job positions using Supervised Machine Learning*. [S.l.]: IEEE, 2018. ISBN 9781538665756. 22, 26

20 KAGGLE. *Kaggle*. 2023. Last accessed 04 October 2023. Disponível em: <https://www.kaggle.com/>. 22, 25, 32, 33, 36

21 VIVEK, M.; YAWALKAR, V. A study of artificial intelligence and its role in human resource management. *IJRAR19UP004 International Journal of Research and Analytical Reviews*, v. 6, p. 20–24, 2019. ISSN 2349-5138. Disponível em: <www.ijrar.org>. 22

22 AL-DARRAJI, S. et al. Employee attrition prediction using deep neural networks. *Computers*, MDPI, v. 10, 11 2021. ISSN 2073431X. 22, 26

23 QUTUB, A. et al. Prediction of employee attrition using machine learning and ensemble methods. *International Journal of Machine Learning and Computing*, v. 11, p. 110–114, 3 2021. ISSN 20103700. Disponível em: <http://www.ijmlc.org/index.php?m=content&c=index&a=show&catid=113&id=1203>. 22, 26

24 SRIVASTAVA, P. R.; EACHEMPATI, P. Intelligent employee retention system for attrition rate analysis and churn prediction: An ensemble machine learning and multi-criteria decision-making approach. *Journal of Global Information Management*, IGI Global, v. 29, 11 2021. ISSN 15337995. 22, 26

25 KANG, I. G.; CROFT, B.; BICHELMEYER, B. A. Predictors of turnover intention in u.s. federal government workforce: Machine learning evidence that perceived comprehensive hr practices predict turnover intention. *Public Personnel Management*, SAGE Publications Inc., v. 50, p. 538–558, 12 2021. ISSN 19457421. 22, 26

26 YUAN, J. Research on employee turnover prediction based on machine learning algorithms. In: . [S.l.]: Institute of Electrical and Electronics Engineers Inc., 2021. p. 114–120. ISBN 9780738131702. 22, 26

27 JUDRUPS, J. et al. Machine learning based solution for predicting voluntary employee turnover in organization. In: . [S.l.]: Latvia University of Life Sciences and Technologies, 2021. v. 20, p. 1359–1366. ISSN 16915976. 22, 26

28 CHAI, T.; QIAN, Y.; WANG, W. Early warning analysis of enterprise employee turnover based on machine learning. *Modern Management*, Hans Publishers, v. 12, p. 1617–1629, 2022. ISSN 2160-7311. 22, 26

29 PUNNOOSE, R.; XAVIER, C. X. Prediction of employee turnover in organizations using machine learning algorithms a case for extreme gradient boosting. *IJARAI) International Journal of Advanced Research in Artificial Intelligence*, v. 5, 2016. Disponível em: <www.ijarai.thesai.org>. 22, 24

30 KANG, I. G. et al. A machine-learning classification tree model of perceived organizational performance in u.s. federal government health agencies. *Sustainability (Switzerland)*, MDPI, v. 13, 9 2021. ISSN 20711050. 22, 26

31 MEHR, H. Artificial intelligence for citizen services and government artificial intelligence for citizen services and government artificial intelligence for citizen services and government. 2017. 23

32 ANASTASOPOULOS, L. J.; WHITFORD, A. B. Machine learning for public administration research, with application to organizational reputation. *Journal of Public Administration Research and Theory*, v. 29, p. 491–510, 6 2019. ISSN 1053-1858. Disponível em: <https://academic.oup.com/jpart/article/29/3/491/5161227>. 24

33 YOUNG, M. M.; BULLOCK, J. B.; LECY, J. D. Artificial discretion as a tool of governance: A framework for understanding the impact of artificial intelligence on public administration. *Perspectives on Public Management and Governance*, v. 2, p. 301–313, 10 2019. ISSN 2398-4910. Disponível em: <https://academic.oup.com/ppmg/advance-article/doi/10.1093/ppmgov/gvz014/5602198>. 24

34 VALLE, M. A.; RUZ, G. A. Turnover prediction in a call center: Behavioral evidence of loss aversion using random forest and naïve bayes algorithms. *Applied Artificial Intelligence*, Bellwether Publishing, Ltd., v. 29, p. 923–942, 10 2015. ISSN 10876545. 24

35 NAMRATA, B. et al. *Employee Attrition Prediction Using Classification Models*. [S.l.]: IEEE, 2019. ISBN 9781538680759. 25, 26, 31

36 JHAVER, M.; GUPTA, Y.; MISHRA, A. K. Employee turnover prediction system. In: . [S.l.]: IEEE, 2019. ISBN 9781728136516. 25, 26

37 A., M. A. et al. Performance predicting in hiring process and performance appraisals using machine learning. In: . [S.l.: s.n.], 2019. ISBN 9781728100456. 26

38 LONG, Y. et al. Prediction of employee promotion based on personal basic features and post features. In: . [S.l.]: Association for Computing Machinery, 2018. p. 5–10. ISBN 9781450364188. 26

39 KAEWWISET, T.; TEMDEE, P. Promotion classification using decisiontree and principal component analysis. In: . [S.l.]: Institute of Electrical and Electronics Engineers Inc., 2022. p. 489–492. ISBN 9781665495103. 26

40 SAHINBAS, K. Employee promotion prediction by using machine learning algorithms for imbalanced dataset. In: . [S.l.]: Institute of Electrical and Electronics Engineers (IEEE), 2022. p. 1–5. ISBN 9781665474832. 26, 31

41 QADIR, M.; NOREEN, I.; SHAH, A. A. Bi-lstm deep learning approach for employee churn prediction. 2021. Disponível em: <http://www.jictra.com.pk/index.php/jictra,pISSN:2523-5729,eISSN:2523-5739>. 26

42 BASKAN, M. et al. A machine learning framework to address customer churn problem using uplift modelling and prescriptive analysis msc research project data analytics. 2022. 26

43 ROUSSEEUW, P. J.; DRIESSEN, K. V. A fast algorithm for the minimum covariance determinant estimator. 1999. 26

44 LIU, F. T.; TING, K. M.; ZHOU, Z.-H. Isolation forest. 2008. 26

45 BREUNIG, M. M. et al. Lof: Identifying density-based local outliers. 2000. 27

46 HOERL, A. E.; KENNARD, R. W. *Ridge Regression: Biased Estimation for Nonorthogonal Problems*. 1970. 27

47 FREUND, Y.; SCHAPIRE, R. E. A decision-theoretic generalization of on-line learningand an application to boosting. v. 55, p. 1199139, 1997. 27

48 BREIMAN, L. Bagging predictors. v. 24, p. 123–140, 1996. 28

49 H., F. J. Gradient boosting machine. 1999. 28

50 HINTON, G. E. Connectionist learning procedures - mlp. 1989. 29

51 C., P. J. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. 1999. 29

52 CHANG, C.-C.; LIN, C.-J. Libsvm: A library for support vector machines. 2001. Disponível em: <www.csie.ntu.edu.tw/>. 29

53 BREIMAN et al. *Classification And Regression Trees*. [S.l.: s.n.], 1984. 29

54 BREIMAN, L. Random forests. v. 45, p. 5–32, 2001. 30

55 CHEN, T.; GUESTRIN, C. Xgboost: A scalable tree boosting system. 3 2016. Disponível em: <http://arxiv.org/abs/1603.02754http://dx.doi.org/10.1145/2939672.2939785>. 30

56 KE, G. et al. Lightgbm: A highly efficient gradient boosting decision tree. 2017. Disponível em: <https://github.com/Microsoft/LightGBM.> 30

57 CHAWLA, N. V. et al. Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, v. 16, p. 321–357, 2002. 31

58 SCIKIT-LEARN. *StackingClassifier*. 2023. Last accessed 04 October 2023. Disponível em: <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.StackingClassifier.html#sklearn.ensemble.StackingClassifier>. 31, 58

59 OLSON, R. S.; EDU, O.; MOORE, J. H. Tpot: A tree-based pipeline optimization toolfor automating machine learning. v. 64, p. 66–74, 2016. Disponível em: <https://github.com/rhiever/tpot.> 31, 59

60 DIAS, A. L. A.; MORAES, C. H. V. *Projeto Turnover*. [S.l.]: GitHub, 2023. <https://github.com/andre-alves-dias/projeto-mestrado>. 32, 63, 101

61 KAKISAMA. *HR-Comma-Sep.csv*. 2017. Last accessed 11 September 2023. Disponível em: <https://www.kaggle.com/datasets/liujiaqi/hr-comma-sepcsv>. 33

62 COKELAER. *fitter*. 2023. Last accessed 19 december 2023. Disponível em: <https://github.com/cokelaer/fitter>. 34, 36, 46, 53

63 PAVANSUBHASH. *IBM HR Analytics Employee Attrition Performance*. 2017. Last accessed 11 September 2023. Disponível em: <https://www.kaggle.com/datasets/pavansubhasht/ibm-hr-analytics-attrition-dataset>. 36

64 WIJAYA, D. *Employee Turnover*. 2020. Last accessed 11 September 2023. Disponível em: <https://www.kaggle.com/datasets/davinwijaya/employee-turnover>. 46

65 REPOSITORY, P. H. A. *Employee Attrition*. 2017. Last accessed 11 September 2023. Disponível em: <https://www.kaggle.com/datasets/HRAnalyticRepository/employee-attrition-data>. 53

66 SCIKIT-LEARN. *VotingClassifier*. 2023. Last accessed 04 October 2023. Disponível em: <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.VotingClassifier.html#sklearn.ensemble.VotingClassifier>. 59