Trabalho de Conclusão de Curso

Sistema preditivo de reprovação de alunos do Ensino Básico

Bruna Damaris Ramos dos Santos

orientado por
Prof. Dr. Marcelo Costa Oliveira

Universidade Federal de Alagoas
Instituto de Computação
Maceió, Alagoas
06 de Dezembro de 2023

UNIVERSIDADE FEDERAL DE ALAGOAS
Instituto de Computação

SISTEMA PREDITIVO DE REPROVAÇÃO DE ALUNOS DO ENSINO BÁSICO

Trabalho de Conclusão de Curso apresentado ao Instituto de Computação da Universidade Federal de Alagoas como requisito parcial para a obtenção do grau de Bacharel em Engenharia de Computação.

Bruna Damaris Ramos dos Santos

Orientador: Prof. Dr. Marcelo Costa Oliveira

Banca Examinadora:

Marcelo Costa Oliveira	Prof. Dr., IC-UFAL
Andressa Martins Oliveira	MsC, IC-UFAL
Rafael Sampaio de Melo Fragoso	Esp., Sistema FIEA

Maceió, Alagoas
06 de Dezembro de 2023

Catalogação na Fonte
Universidade Federal de Alagoas
Biblioteca Central
Divisão de Tratamento Técnico

Bibliotecário: Marcelino de Carvalho Freitas Neto – CRB-4 - 1767

S237s Santos, Bruna Damaris Ramos dos.
Sistema preditivo de reprovação de alunos do ensino básico / Bruna Damaris Ramos dos Santos. – 2023.
48 f. : il.

Orientador: Marcelo Costa Oliveira.
Monografia (Trabalho de conclusão de curso em Engenharia de Computação) - Universidade Federal de Alagoas, Instituto de Computação. Maceió, 2023.

Bibliografia: f. 46-48.

1. Aprendizagem de máquina. 2. Mineração de dados - Educação. 3. Predição de reprovação I. Título.

CDU: 004.81:159.953.5

Dedicatória

Dedico este trabalho a minha família e a todos que estiveram ao meu lado nessa jornada.

Agradecimentos

Gostaria de expressar meus sinceros agradecimentos aos funcionários, professores e técnicos administrativos do Instituto de Computação da UFAL. Em particular, ao professor Marcelo Costa que me orientou durante o processo de desenvolvimento deste trabalho. Agradeço também aos professores do Laboratório EASY e aos funcionários do Sistema FIEA que me proporcionaram meios para realizar o projeto deste trabalho.

Gostaria de agradecer aos meus amigos, Luana, Marcus, Hiago, Tiago Cabral, João Pedro, John Davi, Lucas Massa, Hugo, Mateus, Neto, e outros que me acompanharam na graduação e estiveram ao meu lado me motivando. Também, aqueles que me deram carona para me levar para a UFAL ou para casa, foram de grande ajuda. Agradeço também minha namorada, Laura Costa, que me acompanhou nos períodos finais da graduação, sempre me apoiando, incentivando e me auxiliando nos obstáculos encontrados.

Agradeço a minha família pelo apoio, especialmente a minha mãe Francisca por incentivar e apoiar a minha educação por todos esses anos, e também ao meu irmão Matheus pelo companheirismo.

Maceió, 6 de dezembro de 2023.

Resumo

Este trabalho apresenta os resultados do desenvolvimento de um sistema de predição de reprovação de alunos do Ensino Básico utilizando as notas de avaliações das escolas do Serviço Social da Indústria de Alagoas. O desempenho educacional é um fator essencial nas áreas da educação, assim, é importante o desenvolvimento de um sistema que pode indicar se o aluno está em risco de reprovação. O trabalho visou desenvolver modelos preditivos para predizer a probabilidade de reprovação em uma disciplina em cada marco avaliativo do ano, apresentando resultados com rótulos indicando se os alunos estão previstos como aprovados ou reprovados com a probabilidade associada. A identificação de alunos em risco pode orientar intervenções pedagógicas para melhorar o desempenho e evitar reprovações. Para desenvolvimento do sistema, foram aplicados algoritmos classificadores, os quais foram o K-vizinhos mais próximos, a árvore de decisão, o *Random Forest* e o *eXtreme Gradient Boosting*. Esses foram avaliados em questão de seu desempenho e conforme o funcionamento em relação à solução explorada. Para a avaliação, foram adotadas as métricas da acurácia e o F1-Score, além de ser aplicado o teste de *Mann-Whitney* nos resultados das métricas a fim de apontar se são estatisticamente significativos. Diante disso, o K-vizinhos mais próximos não apresentou diferenças significativas com a árvore de decisão, mas foi distinto do *Random Forest* e *eXtreme Gradient Boosting*. Estatisticamente, o *Random Forest* e *eXtreme Gradient Boosting* superaram os demais. Além disso, a média aritmética foi empregada para sintetizar os resultados dos modelos criados e disso, o *Random Forest* e o *eXtreme Gradient Boosting* apresentaram os melhores resultados de desempenho. Assim, a versão final do sistema preditivo utiliza o algoritmo classificatório *eXtreme Gradient Boosting* para gerar modelos e realizar predições a partir dos dados existentes, onde a escolha do *eXtreme Gradient Boosting* baseou-se em seu desempenho consistente, menor dispersão em relação às médias obtidas e eficiência computacional. Os resultados obtidos a partir da abordagem escolhida foram satisfatórios, com o desempenho dos modelos gerados pelo *eXtreme Gradient Boosting* apresentando cerca de 86% de acurácia e F1-Score.

***Palavras-chave*: Aprendizagem de máquina, Classificação supervisionada, Mineração de dados educacionais, Predição de reprovação**

Abstract

This paper presents the results of developing a student failure prediction system in Basic Education using the evaluation grades from schools affiliated with the Social Service of Industry in Alagoas. Educational performance is a crucial factor in the field of education, making it important to develop a system that can indicate if a student is at risk of failing. The aim of the study was to develop predictive models to estimate the probability of failure in a subject at each evaluation point throughout the year, presenting results with labels indicating whether students are predicted to pass or fail, along with the associated probability. Identifying students at risk can guide pedagogical interventions to improve performance and prevent failures. For the system development, classification algorithms, including K-Nearest Neighbors, Decision Tree, Random Forest, and eXtreme Gradient Boosting, were applied. They were evaluated in terms of performance and functionality concerning the explored solution. Evaluation metrics included accuracy and F1-Score, and the Mann-Whitney test was applied to metric results to determine statistical significance. In this regard, K-Nearest Neighbors showed no significant differences from the Decision Tree but differed from the Random Forest and eXtreme Gradient Boosting. Statistically, the Random Forest and eXtreme Gradient Boosting outperformed the others. Additionally, the arithmetic mean was used to synthesize model results, with the Random Forest and eXtreme Gradient Boosting showing the best performance. Consequently, the final version of the predictive system uses the eXtreme Gradient Boosting classification algorithm to generate models and make predictions based on existing data. The choice of eXtreme Gradient Boosting was based on its consistent performance, lower dispersion in relation to obtained averages, and computational efficiency. The results obtained from the chosen approach were satisfactory, with the performance of models generated by eXtreme Gradient Boosting showing approximately 86% accuracy and F1-Score.

Keywords*: *Machine Learning*, *Supervised Classification*, *Educational Data Mining*, *Failure Prediction

Lista de Figuras

Lista de Símbolos

x_i vetor de coordenadas de um ponto

y_i vetor de coordenadas de outro ponto

$obj(\theta)$ função objetivo

$L(\theta)$ função de perda de treinamento

$\Omega(\theta)$ termo de regularização

Lista de Abreviações

SESI Serviço Social da Indústria

KDD *Knowledge Discovery in Database*

KNN *K-Nearest Neighbors*

XGBoost *eXtreme Gradient Boosting*

FIEA Federação da Indústria do Estado de Alagoas

SGE Sistema de Gestão Educacional

DW *Data Warehouse*

UCs Unidades Curriculares

Sumário

Capítulo 1

Introdução

O desempenho acadêmico de estudantes é um fator de alta importância que requer constante monitoramento, visto que, é uma maneira de medir como os estudantes estão se saindo conforme o plano de estudos proposto pelas escolas. As instituições de ensino podem utilizar informações coletadas a partir de dados de reprovações visando reconhecer obstáculos que podem estar afetando a performance dos estudante e melhorar as ações feitas em sala de aula para trazer melhores oportunidades de ensino para os estudantes(Faria de Souza and Cazella (2022)). As reprovações causam impacto significativo na vida de estudantes, de maneira que, as dificuldades acadêmicas antecipam comportamentos problemáticos a curto prazo, aumentam o risco de abandono escolar e podem comprometer as trajetórias educacionais e profissionais até a idade adulta(Crosnoe (2002)). Assim, identificar e agir para recuperar alunos com potencial de reprovação são práticas cruciais no ambiente educacional, por visarem não apenas a promoção do sucesso acadêmico individual, mas também contribuem para o fortalecimento do sistema educacional na totalidade(Al-Tameemi et al. (2023)).

Nesse contexto, a detecção precoce da situação de risco do estudante quanto antes é relevante para serem aplicadas ações que auxiliem o estudante a melhorar o desempenho, e com isso é possível identificar problemas ou dificuldades de aprendizado no início(Chanlekha and Niramitranon (2018)). A atividade de detecção precoce diz respeito a analisar o desempenho dos estudantes e tentar prever se o estudante irá reprovar. Essa é uma tarefa exaustiva visto que existem grandes volumes de dados a serem analisados a fim de compreender e identificar quais alunos estão em situação de risco de reprovação, sendo esse risco variável entre disciplinas devido às diferentes complexidades que apresentam. Logo, a necessidade da criação de maneiras de intervenções no que diz respeito a alunos que apresentam déficit quanto ao desempenho no aprendizado é a motivação da criação de sistemas preditivos computacionais que podem auxiliar o professor com uma visão do risco de reprovação dos alunos a partir da análise dos resultados preditivos e do que se passa em sala de aula. Assim, surge a necessidade de implementar um método para identificar os riscos de reprovação de estudantes. Isso possibilita a adoção de medidas voltadas para

a melhoria do processo de aprendizado em sala de aula ou a implementação de intervenções e soluções pedagógicas antes que a reprovação ocorra, visando evitá-la(Albreiki et al. (2021)).

A identificação precoce pode ser realizada por meio de sistemas desenvolvidos com o uso de inteligência artificial, gerando indicadores que podem contribuir para a detecção de reprovações, juntamente com valores que refletem as probabilidades associadas a essas reprovações. O uso de inteligência artificial na educação ganha grande destaque devido à demanda crescente de meios digitais de armazenamento de dados. Como consequência, essa transição resultou na geração significativa de informações que populam bases de dados educacionais, com o uso de bancos de dados para o registro de informações dos estudantes das escolas. Isso, por sua vez, abre caminho para a aplicação de técnicas de mineração de dados em diversas aplicações educacionais. Portanto, em conjunto com a importância da análise do desempenho de estudantes, a mineração de dados educacionais juntamente com técnicas de aprendizagem de máquina podem ser utilizadas para prever o desempenho de estudantes e para traçar estrategias que podem trazer melhorias para os métodos de ensino das escolas(Almasri et al. (2019)). Em Souza et al. (2022) foi utilizada uma abordagem de predição de desempenho de alunos com uma probabilidade de reprovação superior a 50% em disciplinas específicas ministradas em um ambiente de ensino online, utilizando dados de informações dos estudantes e dados de atividades online feitas na plataforma, os resultados desse trabalho foram positivos, com especificidade de cerca de 93% para um dos classificadores selecionados, porém, esse trabalho tem as delimitações de ser aplicado somente com dados de turmas online, sendo utilizadas informações somente das duas primeiras atividades avaliativas e tem-se a dependência da avaliação contínua, com a exclusão de disciplinas que não adotam essas práticas, limitando a aplicabilidade. Em Amal Asselman and Aammou (2021) visa avaliar se os métodos de conjuntos podem aprimorar a previsão de desempenho dos estudantes. Utilizando três bases de dados distintas contendo informações sobre alunos, atividades realizadas e habilidades associadas, foram testados os algoritmos Random Forest, AdaBoost e XGBoost. Os resultados indicaram uma melhoria nas previsões, destacando o XGBoost como o classificador mais eficaz, porém, esse trabalho apresentou a limitação dada a sua aplicabilidade as condições especificas analisadas, dependência aos métodos de conjuntos e a dificuldade de generalização quanto as variáveis utilizadas para a criação dos modelos. Em Siddique et al. (2021) é destacada a escassez de pesquisas utilizando bases de dados do ensino básico e enfatiza a importância da predição do desempenho na educação secundária para o sucesso acadêmico futuro. O trabalho visa identificar fatores críticos que impactam o desempenho dos alunos, propondo a construção de modelos de classificação para prever o desempenho acadêmico. A abordagem envolve uma fusão de classificadores, e os resultados indicam que o modelo proposto tem o potencial de identificar precocemente o desempenho acadêmico, proporcionando oportunidades para aprimorar o processo de aprendizagem, porém possui

limitações quanto a consideração de informações especificas individuais dos estudantes que podem causar vieses e pode apresentar delimitações ao ser generalizado conforme os métodos utilizados. O presente trabalho utiliza dados de notas dos estudantes ao longo dos anos em escolas, presencialmente, nos quais existem registro de notas desses alunos de maneira digital, dessa forma, os dados a serem utilizados não consideram de informações individuais dos estudantes a fim de evitar vieses. Além disso, foram testadas diferentes técnicas de aprendizagem de máquina para se chegar a versão final do sistema, possibilitando assim a criação do sistema preditivo capaz de prever a probabilidade de reprovação do aluno a partir do desempenho acadêmico em avaliações.

1.1 Justificativa do projeto

Nas escolas do Serviço Social da Indústria de Alagoas se tem registro de aproximadamente três milhões de dados de notas de avaliações, onde 16,78% dos estudantes do Ensino Fundamental estão assinalados como reprovação nos anos de 2017 a 2022 e cerca de 5% para o novo Ensino Médio no ano de 2022. Os valores apresentados são de aspecto geral das disciplinas das escolas, porém algumas dessas disciplinas se destacam com maiores índices de reprovação como a disciplina de matemática com cerca de 25% de reprovações e a disciplina de língua portuguesa com cerca de 20% de reprovações.

Logo, se tem uma demanda do desenvolvimento de um sistema preditivo capaz de indicar a quantidade de estudante em risco de reprovação matriculados no ano corrente e o indicativo de quais estudantes estão nessa situação a fim de serem reconhecidos os fatores que podem ajuda-lós, e assim sendo possível realizar analises em relação às informações nos cenários que apresentam registros de reprovações preditos. Assim, a abordagem proposta no presente trabalho para o desenvolvimento do sistema preditivo visa a performance dos estudantes quanto as notas de avaliações realizadas e registradas nos sistemas das escolas, utilizando aprendizagem de máquina para aprender padrões históricos baseados nas notas dos anos anteriores ao ano corrente, fazendo assim a criação de modelos preditivos para cada disciplina presente na base de dados.

1.2 Objetivo

O objetivo principal desse trabalho foi o desenvolvimento de um sistema preditivo utilizando técnicas de aprendizagem de máquina, capaz de prever a probabilidade de um estudante ser reprovado em uma disciplina seguindo a classificação de 'Aprovado' ou 'Reprovado'. A partir dos resultados apresentados pelo sistema preditivo, foi possível ter indicativos dos estudantes que estão em risco de reprovação.

1.3 Organização do trabalho

O presente trabalho foi organizado em capítulos detalhando o processo realizado para a produção de um sistema preditivo, apresentando os conceitos e métodos de aprendizagem de máquina utilizados, sendo explicadas as fases empregadas, iniciando-se com a obtenção dos dados, a manipulação dos dados, a utilização dos dados em algoritmos classificatórios supervisionados e por fim a obtenção de resultados de desempenho dos modelos preditivos desenvolvidos assim como resultados de predições.

O Capítulo 2 apresenta os conceitos dos métodos utilizados para o desenvolvimento do sistema. São conceitos e métodos de inteligência artificial, com foco na manipulação de dados, extração de conhecimento, mineração de dados e aprendizagem de máquina supervisionada, assim como algoritmos classificadores.

O Capítulo 3 descreve o processo feito para aplicar as fases descritas no capítulo 2 para os dados da base de dados das escolas do Serviço Social da Indústria de Alagoas, assim como a arquitetura do sistema desenvolvido baseado nos dados obtidos e nos resultados esperados.

O Capítulo 4 apresenta os resultados de avaliações de desempenho para a escolha do algoritmo classificador, assim como justifica a resolução do classificador escolhido para a versão final do sistema. Mostra também a forma em que os resultados de saída do sistema são apresentados e o que se pode ser realizado a partir das informações obtidas.

Capítulo 2

Fundamentação Teórica

A análise manual e interpretativa se torna exaustiva e de grande custo ao ter uma grande e crescente quantidade de dados, em alguns casos a análise manual é inviável para o usuário e técnicas podem ser aplicadas a fim de ajudar no processo de reconhecimento de padrões e a extração de dados úteis em grandes bases de dados. Através das técnicas de extração de conhecimento em bases de dados, notavelmente destacadas pelo KDD (*Knowledge Discovery in Database*, em inglês), é viável empregar o método de mineração de dados, essencial para descobrir conhecimento em grandes conjuntos de dados. Dentro desse cenário, o processo de KDD não apenas facilita a extração de conhecimento, mas também permite a interpretação dos dados para além do que seria possível apenas com métodos estatísticos. Nesse contexto, as técnicas de KDD proporcionam uma abordagem abrangente, possibilitando a descoberta de percepções valiosas e a interpretação de padrões que poderiam passar despercebidos em análises puramente estatísticas. O KDD, assim, representa uma ferramenta crucial para explorar e compreender extensos conjuntos de dados, enriquecendo a extração de conhecimento e ampliando as possibilidades de interpretação dos dados(Fayyad et al. (1996)).

2.1 Extração de conhecimento(KDD)

O KDD é uma técnica voltada a resolver o problema do entendimento de dados que excedem uma quantidade plausível de ser analisada manualmente. Através da aplicação de métodos computacionais, o KDD é empregado, utilizando seus resultados para beneficiar o usuário com o propósito de adquirir conhecimento de alto nível a partir de dados inicialmente em níveis mais detalhados.(Fayyad et al. (1996)). Este processo serve como suporte a confirmação de conhecimento relevante na composição da base de dados para algoritmos de classificação(Deepthi et al. (2019)).

Essencialmente, o processo do KDD consiste em cinco fases, a seleção de dados, pré-processamento de dados, transformação de dados, mineração de dados, análise e interpretação dos resultados.

O funcionamento do KDD pode envolver iterações e loops entre quaisquer etapas do processo. Cada um desses processos pode ser descrito da seguinte forma:

- Seleção de dados: consiste em selecionar um conjunto de dados relevantes para a resolução do problema em questão. Isso ocorre primeiramente com o entendimento do problema a ser resolvido e então da análise inicial dos dados existentes, que pode ser feita com base em alguma interpretação manual, baseada em conhecimento prévio e em conjunto com métodos estatísticos.

- Pré-processamento: consiste na limpeza de dados feita a partir da remoção de ruídos ou partes dos dados que possam ser vistos como prejudiciais nos resultados da análise de padrões e, além disso, é feito o tratamento de dados faltantes e de erros nos dados quando ocorrem. Ainda, realiza o enriquecimento de dados no qual pode ser feita a adição, se necessária, de mais informações, também é uma maneira de lidar com dados faltantes quando não se pode excluí-los, procurando a melhor forma de preenchê-los conforme os dados existentes e do problema a ser resolvido pelo processo.

- Transformação de dados: essa etapa do processo consiste em transformar os dados existentes de modo a normalizá-los ou convertê-los em valores que podem ser melhor interpretados e analisados, assim como o mapeamento de dados para um tipo específico que seja ideal. Ainda, técnicas de redução de dimensionalidade podem ser aplicadas nessa fase a fim de diminuir o número de variáveis presentes.

- Mineração de dados: consiste em aplicar algoritmos de classificação, regressão e clusterização nos dados obtidos para criação de um modelo. Nessa fase, os dados são utilizados como entrada dos algoritmos, sendo feitos testes de avaliação de modelos para se saber o algoritmo adequado para cada caso. Esses testes são feitos pela máquina e seus resultados são analisados a partir de métodos estatísticos e análises comparativas de resultados das métricas.

- Análise e interpretação dos resultados: é a análise dos resultados obtidos baseado no conhecimento prévio e no que foi extraído ao longo do processo do KDD. É nessa fase que se definem as ações que podem ser feitas a partir dos resultados de saída, e dessa maneira pode ser definido como será utilizado o conhecimento útil extraído no problema no qual os dados iniciais fazem parte. Esse passo pode ser feito a partir da visualização dos dados de maneira gráfica, a partir de tabelas e estatísticas e a partir de interpretações.

2.1.1 Mineração de dados

O processo de mineração de dados consiste na descoberta de conhecimento sobre informações em grandes bases de dados por meio do reconhecimento de regras de associação e padrões (Chakarverti et al. (2019)). A mineração de dados concentra-se na fase em que os dados são submetidos a algoritmos de aprendizado de máquina e análise de dados. Esses algoritmos geram modelos e padrões que serão descobertos e extraídos para produzir resultados, os quais podem ser analisados pelo usuário, seja por meio de dados numéricos, classificações ou agrupamentos(Fayyad et al. (1996)). A definição de padrões nesse caso acontece em questão da busca, exploração e manipulação dos dados, sendo que, isso ocorre limitadamente dependendo do algoritmo utilizado(Chakarverti et al. (2019)). Os modelos criados a partir dos algoritmos nessa fase podem ser analisados por meio de métodos estatísticos e testes comparativos conforme o resultado de métricas obtidas baseado no resultado ao fim da execução dos algoritmos.

É necessário se ter conhecimento dos dados de entrada e também do funcionamento dos algoritmos que criam os modelos para se ter uma solução que se ajusta corretamente ao problema a ser tratado. Os modelos são criados a partir de parâmetros que podem ser definidos pela análise inicial dos dados ou por meio de testes baseados em buscas exaustivas e iterativas. Dessa forma, o conjunto de conhecimento dos dados, do algoritmo e dos resultados de métricas obtidos indicam que o problema está sendo resolvido com as ferramentas adequadas disponíveis, evitando assim a criação de vieses que podem dificultar a análise e interpretação de dados. A partir disso, os padrões presentes nos dados são encontrados e podem ser representados a depender do algoritmo escolhido e utilizado, podendo ser de forma gráfica com árvores e agrupamentos, de forma numérica, ou de regras, ou similaridades entre os dados(Chakarverti et al. (2019)).

Em diversos casos, se tem um foco na mineração preditiva, por meio da modelagem preditiva, que envolve especificamente algoritmos que produzem modelos preditivos a partir da aprendizagem de máquina supervisionada.

A modelagem preditiva é uma técnica estatística usada para prever o comportamento futuro e funciona por meio da análise de dados históricos e atuais, criando um modelo preditivo que usa mineração de dados e probabilidade para prever resultados(Nasteski (2017)).

2.2 Aprendizagem de Máquina

A aprendizagem de máquina é a área que lida com os algoritmos computacionais que podem ser utilizados na fase de mineração de dados. Consiste na criação de algoritmos que permitem à máquina aprender para representar o aprendizado semelhante ao de um humano(Nasteski (2017)).

A aprendizagem de máquina pode ser dividida em aprendizagem supervisionada, aprendizagem não-supervisionada e aprendizagem por reforço. A aprendizagem supervisionada utiliza dados com rótulos definidos e apresenta um supervisor que dita os resultados desejados para as entradas de modo que, o algoritmo deve aprender e encontrar uma função baseada nos padrões aprendidos a fim de chegar em uma conclusão conforme os resultados dados pelo supervisor. Logo, se tem variáveis chamadas de atributos e se tem variáveis que são chamadas de rótulos, o qual são as saídas definidas previamente para cada um dos dados. Assim, o algoritmo gerará uma função que mapeia os atributos a suas saídas, a partir dos padrões aprendidos para a criação de um modelo preditivo(Norvig and Russell (2014)).

Algoritmos de classificação tem os dados rotulados por classes. Essas classes são predefinidas, onde, nos dados selecionados, cada um será associado a uma ou mais classes de modo que essa definição é o rótulo para cada dado. O resultado desse algoritmo após o aprendizado dos padrões é a criação de um modelo preditivo onde a principal tarefa é construir um modelo capaz de prever o rótulo de um dado baseado pelo seu conjunto de características. Em seguida, o algoritmo de aprendizagem recebe um conjunto de recursos, como entradas com as saídas corretas, e aprende comparando sua saída real com as saídas corrigidas para encontrar erros(Nasteski (2017)).

A figura 2.1 apresenta um exemplo do fluxo de uma aplicação de aprendizagem de máquina. O processo de criação de um modelo preditivo é definido pelo treinamento do modelo com os dados de entrada rotulados e, então, a predição de novos dados inseridos. O treinamento do modelo consiste na parte do aprendizado, é nessa fase que os dados de entrada rotulados são processados pelo algoritmo, sendo realizada a aprendizagem dos padrões, onde esses padrões são salvos no modelo criado de uma maneira que irá depender do algoritmo escolhido a ser utilizado, de forma que são definidos os melhores parâmetros do algoritmo a fim de fazer o ajuste adequado para os dados de entrada. Para serem feitos testes comparativos em relação ao desempenho de modelos, são reservados dados de testes que não participam do treinamento, mas estão previamente rotulados. Em seguida, a predição sobre os novos dados registrados é feita e, então, é obtido um resultado de classificação para cada dos dados não-rotulados(Nasteski (2017)).

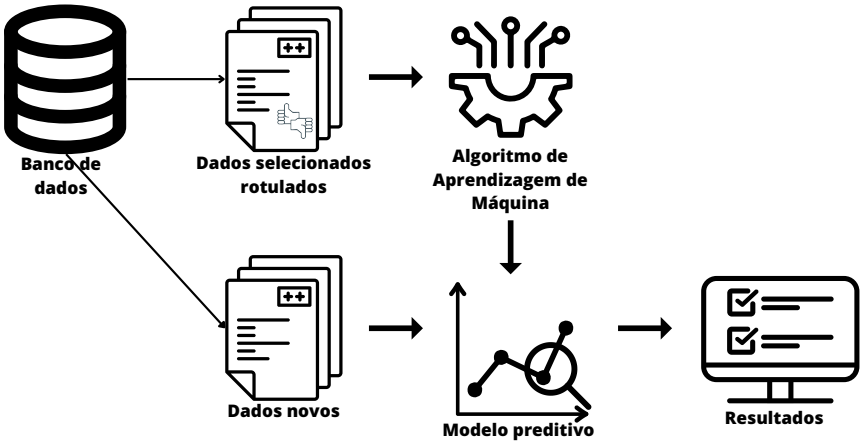

Figura 2.1: Processo de aprendizagem de máquina.

Os algoritmos de classificação usados neste trabalho são apresentados a seguir.

2.2.1 *K-Nearest Neighbors* (KNN)

O KNN é um modelo de aprendizagem supervisionada não paramétrico que visa encontrar os K vizinhos mais próximos de um valor x e então definir um resultado y ao valor x baseado nos K vizinhos predominantes mais próximos encontrados(Norvig and Russell (2014)). Os vizinhos mais próximos são definidos a partir de uma métrica de distância que pode ser a distância euclidiana entre os dados quando vistos em um plano, de forma que, os dados são divididos em grupos predefinidos e então para cada grupo um dado estará mais próximo dos dados do mesmo grupo, formando assim *clusters* com resultados predefinidos. Assim, para classificar ou obter um resultado sobre um novo dado, é vista a distância euclidiana entre o novo dado e seus K vizinhos mais próximos, onde o resultado do novo dado será a classificação predominante entre os seus vizinhos. O parâmetro K é definido pelo usuário baseado em testes nos quais o objetivo é encontrar o melhor K que os dados se ajustam. Os novos dados são classificados segundo a atribuição de uma classe contendo o número máximo de vizinhos próximos(Taunk et al. (2019)).

O método padrão para calcular a distância entre vizinhos no algoritmo KNN é a distância euclidiana padrão. Essa escolha é a mais prevalente, simplificando o algoritmo em comparação com abordagens mais complexas, como aquelas que envolvem métodos de integração ou diferenciação(Taunk et al. (2019)). Ainda, a distância euclidiana é comumente utilizada para definir a distância entre os pontos para dados contínuos. É uma medida que quantifica a distância entre dois pontos em um espaço euclidiano. É a distância "em linha reta" entre dois pontos em um plano cartesiano. Essa distância é baseada no teorema de Pitágoras e é aplicada em espaços bidimensionais ou tridimensionais, embora

possa ser generalizada para espaços de dimensões superiores.

$$\sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \tag{2.1}$$

Essencialmente, a fórmula calcula a diferença entre as coordenadas dos dois pontos, nesse caso x e y, eleva essas diferenças ao quadrado, soma os resultados e tira a raiz quadrada do total.

No caso do KNN para a predição de uma classificação temos as definições no qual o KNN realiza o agrupamento dos dados em conjuntos de acordo com sua classificação e então classifica os novos dados recebidos conforme o quão semelhante o novo dado é com os dados que foram previamente utilizados no treinamento do modelo e é um método bastante utilizado por sua simplicidade e baixo tempo computacional(Taunk et al. (2019)).

Por exemplo, na figura 2.2 um novo dado é inserido na base de dados. Os pontos da imagem representados como círculos indicam um grupo de dados que apresentam o mesmo rótulo, e da mesma forma os pontos representados como quadrados apresentam o mesmo rótulo entre si e diferente do rótulo do grupo de círculos. O novo dado não-rotulado, representado por um triângulo, é rotulado segundo os seus vizinhos mais próximos conforme o K definido previamente, assim, para a classificação definindo um K igual 3, o novo dado é predito como associado ao grupo de quadrados, porém para o K igual a 9, o novo dado é predito como associado ao grupo de círculos, o que mostra que a escolha do parâmetro K adequado é de extrema importância para os resultados gerados pelo algoritmo.

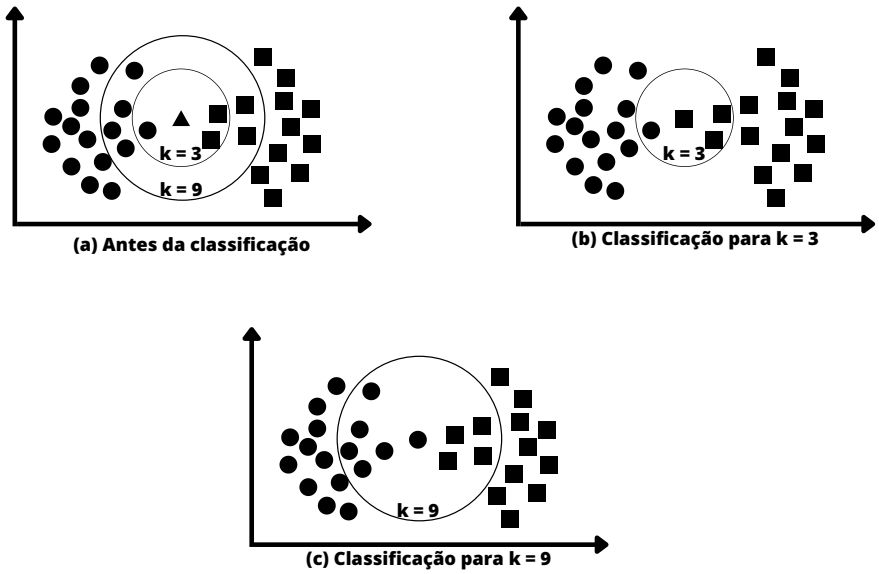

Figura 2.2: Exemplo do funcionamento do KNN.

2.2.2 Árvores de decisão e *Random Forests*

A árvore de decisão é um algoritmo de mineração de dados que classifica os dados conforme as folhas de uma árvore e os galhos que levam as folhas. O modelo criado a partir desse algoritmo é semelhante a uma árvore em que, na sua origem, está presente um nó raiz que representa a decisão inicial do modelo. O nó raiz é definido segundo a importância da informação no momento em que o modelo foi criado, onde, o atributo mais importante para a tomada de decisão é selecionado como o nó raiz conforme o treinamento do modelo. Os nós internos de decisão são os demais nós da árvore percorridos para se chegar a uma decisão no modelo. Os galhos da árvore estão ligados aos nós internos e indicam a resposta da pergunta feita ao nó de decisão, onde cada galho define um caminho para se chegar a outro nó a partir da resposta da pergunta do nó atual. Por fim, para se chegar a uma decisão, as perguntas são feitas e respondidas até se chegar a um nó folha, que representa a decisão final do modelo. Dessa maneira, os nós da árvore são definidos a partir dos dados de entrada considerados no treinamento do modelo, onde os atributos de entrada são utilizados para definir o nó raiz e os demais nós de decisão e os rótulos de saída do modelo são utilizados para determinar a classe de cada nó folha(Nasteski (2017); Song and Lu (2015)).

A figura 2.3 ilustra uma árvore de decisão simples, com a profundidade de três camadas, sendo a primeira a que possui o nó raiz, a segunda possui nós de decisão e a terceira

possui os nós folha.

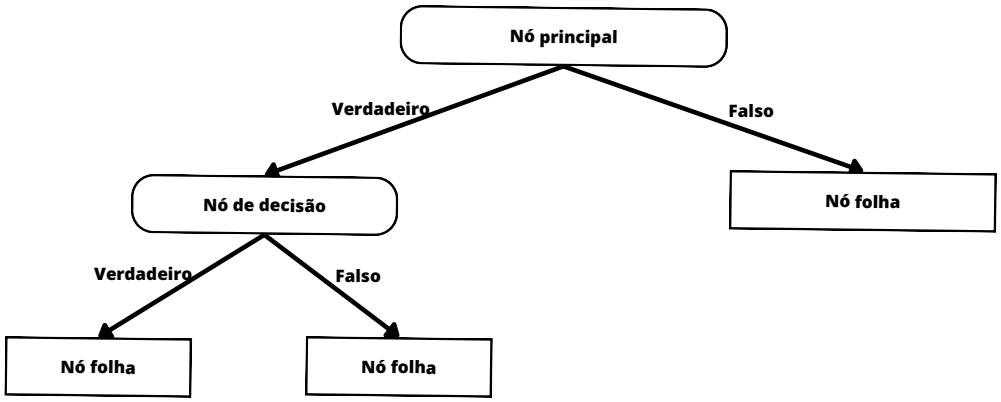

Figura 2.3: Diagrama de uma árvore de decisão simples.

Para a escolha de cada nó da árvore de decisão, os atributos escolhidos são relacionados ao grau de 'pureza' dos nós folha resultantes na criação do modelo. O grau de pureza é calculado de acordo com algumas características como o ganho de informação, a entropia, índice de Gini, erro de classificação, taxa de ganho e critérios de dualidade(Song and Lu (2015)).

O *Random Forest* é um estimador que faz o ajuste de vários classificadores de árvore de decisão utilizando a técnica de conjuntos, combinação de diferentes modelos, para se ter um conjunto de árvores de decisão criadas de maneiras aleatórias. A partir dos dados de entrada selecionados, são feitas várias sub amostras de tamanho controlado para criar vários modelos de árvores de decisão, e ainda, cada modelo criado apresenta variáveis selecionadas de maneira aleatória. É utilizada a média para controlar o ajuste dos modelos e melhorar a precisão preditiva. Para problemas de classificação, o resultado do algoritmo será o predominante entre os resultados das árvores(Louppe (2015); Breiman (2001)).

No caso do *Random Forest*, alguns dos parâmetros-base que podem ser destacados são os seguintes:

- criterion="gini";
- max_depth=None;

2.2.3 *eXtreme Gradient Boosting*(XGBoost)

O XGBoost, semelhante ao *Random Forest*, também utiliza a técnica de conjuntos, combinando resultados de modelos de árvore de decisão distintos criados para gerar uma predição conforme a soma ponderada de todas as previsões dos modelos gerados.

A maneira na qual as árvores de decisão são criadas o diferencia do algoritmo de *Random Forest* onde o XGBoost, utiliza *Gradient Tree Boosting* para gerar novas árvores.*Gradient Tree Boosting* consiste em um método de treinamento de modelo de maneira aditiva na qual a otimização da função de perda é baseada no algoritmo de *Gradient Descent*. Nesse caso, são criadas árvores simples iterativamente e se tenta minimizar os erros de predição ao longo dos modelos criados. Ainda, no XGBoost as árvores são criadas sequencialmente, onde cada árvore é criada em busca da redução de erros da anterior, enquanto no *Random Forest* são criadas paralelamente(Nirmala et al. (2022)).

Em tarefas de classificação, o treinamento do modelo é feito de maneira em que são feitos ajustes para serem encontrados os melhores parâmetros baseados nos dados de entrada em relação às saídas desejadas, para saber se o modelo criado está se ajustando bem aos dados com os parâmetros definidos, é utilizada uma função objetivo, que basicamente consiste em dois parâmetros, a função de perda e de um termo de regularização. A função de perda mede o quão preditivo o modelo criado é em relação aos dados de treinamento(Chen and Guestrin (2016)). Então, ao longo do treinamento, as árvores de decisão são criadas em vista da otimização da função objetivo.

Para o XGBoost, a função objetivo pode ser vista da seguinte forma:

$$obj(\theta) = L(\theta) + \Omega(\theta) \tag{2.2}$$

onde L é a função de perda de treinamento e Ω é o termo de regularização.

A perda de treinamento mede o quão preditivo nosso modelo é em relação aos dados de treinamento.

O termo de regularização controla a complexidade do modelo, ajudando a evitar *overfitting*, que consiste em um sobre-ajuste que faz com que dados novos sejam classificados de maneira ineficaz(Chen and Guestrin (2016)). Os principais termos de regularização utilizados são a Regularização L1 (Regularização Lasso), Regularização L2 (Regularização Ridge) e Regularização L1 e L2 combinada (regularização líquida elástica).

A regularização L1 ou regularização de operador de seleção e contração mínima absoluta, adiciona a soma dos valores absolutos dos coeficientes de características como um termo de penalidade à função objetivo. O algoritmo aborda a colinearidade reduzindo os coeficientes dos preditores correlacionados para zero, incentivando a dispersão no conjunto de recursos(Chen et al. (2019)). O parâmetro que controla essa regularização é o alpha.

A regularização L2 adiciona a soma dos valores quadrados dos coeficientes de características como um termo de penalidade à função objetivo. O termo de regularização é

proporcional à magnitude quadrada dos parâmetros do modelo(Hastie (2020)). O parâmetro que controla essa regularização é o lambda.

A regularização L1 e L2 combinada (regularização líquida elástica) é uma combinação linear de termos de regularização L1 e L2, superando as limitações da L1. Simultaneamente faz a seleção automática de variáveis e encolhimento contínuo, e pode selecionar grupos de variáveis correlacionadas(Zou and Hastie (2005)). O parâmetro alfa controla o equilíbrio entre a regularização L1 e L2 e os parâmetros que controlam essa regularização são o alpha e o lambda.

No caso do XGBoost, alguns dos parâmetros-base que podem ser destacados são os seguintes:

- min_child_weight=1;
- gamma(min_split_loss)=0;
- subsample=1;
- eta(learning_rate)=0,3;
- colsample_bytree=1;
- max_depth=6;
- lambda(reg_lambda)=1;
- alpha(reg_alpha)=0;

2.3 Métricas de avaliação de modelos de classificação

A fim de realizar testes comparativos para avaliar os modelos criados por algoritmos de classificação são empregadas diversas métricas de avaliação para analisar o desempenho de modelos de classificação. A escolha dessas métricas depende da natureza do problema em questão e dos objetivos específicos da análise. Entre as métricas de avaliação de classificação amplamente utilizadas estão a acurácia, a revocação, a precisão e o F1-Score. Essas métricas foram definidas conforme a matriz de confusão, usada para descrever modelos de classificação da seguinte maneira: as predições positivas ou corretas são os verdadeiros positivos(VP) e os verdadeiros negativos(VN), enquanto as predições negativas ou incorretas são os falsos positivos(FP) e os falsos negativos(FN)(Vujovic (2021)).

A acurácia é uma das métricas mais comumente usadas para avaliar o desempenho, tendo sido utilizada como valor que indica a medição do quão próxima às predições estão dos verdadeiros valores do teste. A acurácia é calculada como a soma das predições

corretas dividida pelo número total de dados(Vujovic (2021)). A equação correspondente dessa métrica é a 2.3:

$$ACC = \frac{VP + VN}{VP + VN + FP + FN} \tag{2.3}$$

A revocação (sensibilidade ou revocação) é calculada como o número de verdadeiros positivos dividido pela soma de verdadeiros positivos e falsos negativos(Vujovic (2021)). Assim, indica qual a porcentagem de dados classificados como positivos comparados com a quantidade real de positivos que existentes. A equação correspondente dessa métrica é a 2.4:

$$REV = \frac{VP}{VP + FN} \tag{2.4}$$

A precisão é calculada como o número de verdadeiros positivos, dividido pelo número total de predições positivas(Vujovic (2021)). Assim, é uma medida que indica e todos os dados classificados como positivos, quantos são realmente positivos. A equação correspondente dessa métrica é a 2.5:

$$PREC = \frac{VP}{VP + FP} \tag{2.5}$$

O F1-Score é utilizado como um indicativo de precisão e revogação, já que esse valor é a média harmônica dessas métricas(Vujovic (2021)). A equação correspondente dessa métrica é a 2.6:

$$F1 = \frac{2 * PREC * REV}{PREC + REV} \tag{2.6}$$

Além disso, após obtidos resultados das métricas, pode ser aplicado o teste de *Mann-Whitney*, em que ao lidar com duas amostras independentes, o teste verifica se uma variável tem uma tendência a apresentar valores superiores à outra (Hart (2001)). Esse teste pode ser utilizado como indicativo dos valores serem estatisticamente significativo, em outras palavras, a diferença entre os valores selecionados aleatoriamente das populações de uma amostra é significativamente grande do ponto de vista estatístico.

2.4 Trabalhos relacionados

Em Amal Asselman and Aammou (2021), se tem o objetivo de identificar se os métodos de conjuntos podem melhorar a predição de desempenho dos estudantes. Nesse caso, são utilizadas três bases diferentes que apresentam dados de estudante, de atividades realizadas e das habilidades associadas as atividades realizadas e essas bases é testado os algoritmos *Random Forest*, AdaBoost e XGBoost. Esse trabalho mostrou que a abordagem utilizada foi efetiva, na qual houve o melhoramento do desempenho das predições e o classificador que se saiu melhor foi o XGBoost e ainda foi visto que o *Random Forest* alcançou melhores resultados quando os dados em menores quantidades, e no caso de grandes conjuntos de dados, é recomendado o uso de um algoritmo de *boosting* como o AdaBoost e o XGBoost e, além disso, o classificador XGBoost obteve o maior desempenho

de previsão devido à sua escalabilidade assim como fornece resultados de última geração numa ampla gama de problemas.

Em Siddique et al. (2021) aponta que estudos que utilizam bases de dados voltadas ao ensino básico são escassas e propõem que a predição de desempenho na educação secundária é de importância para o desempenho do estudante em níveis educacionais mais altos. Dito isso, esse trabalho teve o objetivo de determinar fatores críticos que podem afetar o desempenho dos alunos e realizar a construção de modelos de classificação para a previsão do desempenho acadêmico. Os classificadores usados foram uma fusão de classificadores únicos e os que apresentaram melhor desempenho foram um *MultiBoost* com *Multilayer Perceptron* acurácia de 98,7% e com precisão, revogação e F-Score de 98,6%, e implicou que o modelo proposto pode servir para ser identificado o desempenho acadêmico em fases iniciais para melhorar a aprendizagem.

Em Singh and Pal (2020) propõe uma abordagem de comparação de algoritmos de classificação baseados em *bagging*(como o *Random Forest* e em *boosting*(como o XGBoost) utilizando uma base de dados para a predição de desempenho dos estudantes em universidades. A base de dados utilizada consistiu em informações individuais dos estudantes e o desempenho acadêmico. Os resultados obtidos apontaram que as técnicas de *boosting* apresentaram acurácia de 91,76% em comparação com 89,56% das técnicas de *bagging*.

Em Souza et al. (2022) foi proposta uma abordagem que visou uma estratégia para predição para o desempenho de alunos com uma probabilidade de reprovação superior a 50% em disciplinas específicas ministradas em ambientes de ensino online. Os dados utilizados foram em questão do ensino técnico integrado para turmas do primeiro ano, segundo ano e terceiro ano.As predições foram feitas utilizando algoritmos de classificação Naive Bayes, KNN, *Support Vector Machine*, *Random Forest*, *Gradient Boosting* e XGBoost. Os dados utilizados para a entrada dos modelos criados foram os dados cadastrais individuais dos estudantes, assim como as notas das duas primeiras avaliações feitas no ambiente de ensino online em conjunto com tempo de envio das avaliações e o tempo de último acesso do estudante. Os resultados obtidos apontaram cerca de 93% de especificidade para *Support Vector Machine* e 80% para *Gradient Boosting*.

Capítulo 3

Metodologia

3.1 Arquitetura do sistema

Para desenvolver a ferramenta, foram utilizados dados históricos das avaliações dos alunos do Ensino Básico e, então, aplicamos técnicas de modelagem preditiva para antecipar eventos ou desfechos futuros de aprovação, ou reprovação. A figura 3.1 apresenta uma visão geral do processo de execução do sistema, que consiste na aquisição e processamento dos dados, treinamento do modelo de classificação, e o uso do modelo desenvolvido para prever novos dados.

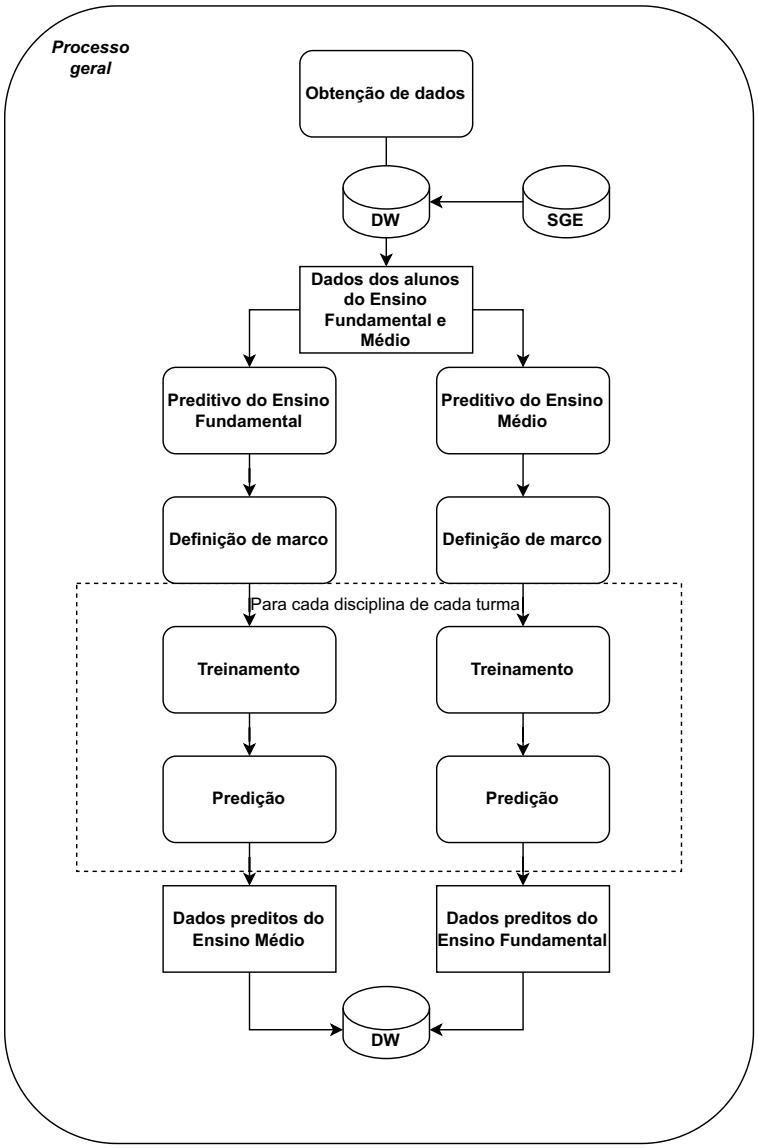

Figura 3.1: Fluxogramas de arquitetura do sistema.

O processo geral do sistema se inicia com a aquisição de dados de estudantes a partir da base de dados disponíveis. Os dados educacionais são adquiridos a partir do DW, que realiza a coleta das notas dos alunos na base do Software de Gestão Educacional (SGE) e as armazena em uma tabela específica no DW. Esse processo permite ao sistema acessar as notas individuais de cada aluno, para cada disciplina e a sua respectiva turma, de maneira consolidada em uma única tabela do DW. Em seguida, o sistema foi designado para fazer uma separação quanto ao tratamento de dados do Ensino Fundamental e Médio, isso é justificado pelo fato de que, primeiramente, os dois modos podem estar em diferentes marcos avaliativos no ano no momento em que o sistema preditivo é executado e também os dados históricos do Ensino Médio foram registrados no banco de dados de uma ma-

neira diferente da maneira na qual o Ensino Fundamental foi registrado e em vista disso, foram criados duas seções de códigos para tratar desses dados. Dessa forma, os marcos avaliativos são definidos segundo as notas que existem no sistema no momento da execução, e em seguida, definidos esses marcos, sabe-se quais notas de avaliações podem ser utilizadas para assim começar o treinamento do modelo preditivo. O treinamento ocorre para cada disciplina da série, de modo que, cada uma das disciplinas apresenta dados que são utilizados para a criação de um modelo preditivo. Em toda a execução do sistema, por volta de dez modelos preditivos são criados para cada série, em sete séries existentes no sistema. Portanto, para cada modelo criado são feitas as predições dos novos dados recebidos de estudantes matriculados. Ao término do processo, a ferramenta prediz a classificação de 'Aprovado' ou 'Reprovado' para cada estudante baseado no acumulado de notas por disciplina. Além da classificação, a ferramenta fornece a probabilidade do risco de reprovação do aluno. Os resultados obtidos são registrados em uma tabela em conjunto com as informações dos estudantes matriculados no ano corrente, sendo salvos no banco de dados.

Este trabalho utilizou a linguagem de programação *Python* 3.8 em conjunto de suas bibliotecas disponíveis de manipulação de dados e de aprendizagem de máquina. Essas bibliotecas foram:

- logging: para registro de eventos no código.
- pandas: para funções e manipulação de *Dataframes* de dados.
- numpy: para funções e manipulação de vetores de dados.
- datetime: para operações de tempo.
- dotenv: para ler pares chave-valor de um arquivo .env e poder defini-los como variáveis de ambiente.
- sqlalchemy: para criação da *engine* para conectar ao *Data Warehouse*(DW).
- urllib.parse: para processamento da senha das variáveis de ambiente para conectar ao DW.
- sklearn: que implementa uma variedade de algoritmos de aprendizado de máquina, pré-processamento, validação cruzada e visualização usando uma interface unificada.
- LabelEncoder: para codificação de rótulos do treinamento do classificador.
- RandomForestClassifier: utilizado como classificador para o algoritmo *RandomForest*.
- KNeighborsClassifier: utilizado como classificador para o algoritmo KNN.

- DecisionTreeClassifier: utilizado como classificador para o algoritmo árvore de decisão.
- XGBClassifier: utilizado como classificador para o algoritmo XGBoost.

Ainda, para a obtenção dos dados foi usado o *Microsoft SQL Server* e o *DBeaver*. O projeto está presente no *GitHub* de modo privado, e é executado a partir da automatização feita pelo *Airflow* em forma de *Directed Aciclic Graph*.

O projeto foi realizado em colaboração com a Federação da Indústria do Estado de Alagoas(FIEA). O código-fonte do projeto se encontra de modo privado devido a questões de privacidade, onde foi assinado um termo de confidencialidade. Ainda, os dados obtidos da base de dados se trataram de dados relativos a pessoas físicas, sendo tratados com restrições de acesso visando o manuseamento seguro. Dada a natureza sensível desses dados, medidas foram tomadas para evitar divulgações indevidas.

Este trabalho foi conduzido em etapas, seguindo o que foi realizado durante o desenvolvimento do sistema. A figura 3.2 mostra o fluxo do projeto de acordo com essas etapas. Inicialmente, os dados foram selecionados conforme a relevância para o problema, a qualidade e a quantidade de dados existentes na base de dados(3.2). Com base nos dados selecionados, esses foram extraídos a partir do DW os dados das séries 6º, 7, 8º e 9º ano do Ensino Fundamental e 1º, 2º e 3º ano do novo Ensino Médio base(3.2.1). Obtidos os dados, foi realizado o pré-processamento para preparação dos dados de entrada do modelo de classificação criados(3.3). Em seguida, foi feita a mineração de dados, com a utilização dos dados obtidos e processados como entrada de algoritmos de classificação(3.4).

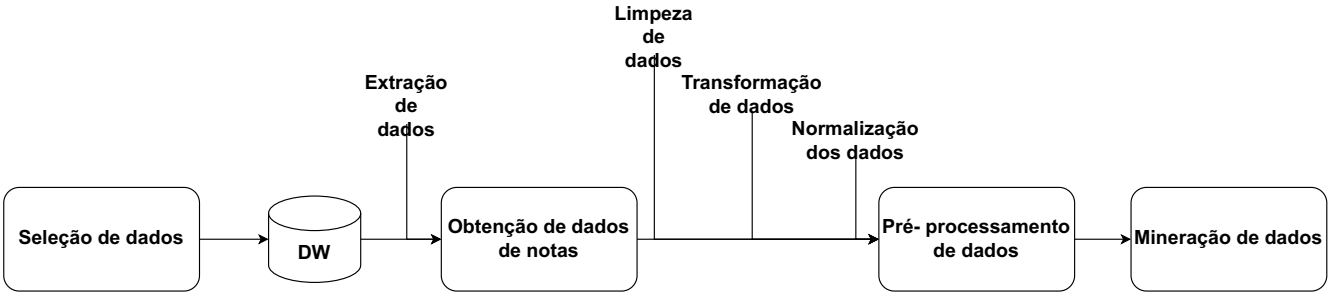

Figura 3.2: Fluxo do projeto.

3.2 Seleção de dados

A seleção dos dados baseou-se na natureza do problema que o sistema se propõe a resolver. O problema em questão relaciona-se ao desempenho dos estudantes do Ensino Básico em relação às reprovações em cada disciplina. Dessa forma, os dados selecionados para refletir esse desempenho são as notas das avaliações trimestrais registradas no sistema. Essas notas são os critérios utilizados pelos professores para determinar se um estudante foi aprovado ou reprovado, tornando-as essenciais para o propósito do sistema. As avaliações são pontuadas numa escala de 0 a 10, sendo que notas iguais ou superiores a 6,0 indicam um resultado positivo na avaliação, enquanto notas inferiores a 6,0 indicam um resultado negativo, o que pode resultar em reprovação.

A base de dados utilizada contém mais de três milhões de registros de notas de alunos em diversas disciplinas no período de 2017 a 2023. Os dados relevantes para o desenvolvimento foram as notas dos alunos, usadas como os atributos no algoritmo de aprendizagem de máquina, a classificação dos alunos como Aprovado ou Reprovado, utilizada como rótulo para o algoritmo de aprendizagem de máquina, as etapas avaliativas associadas às notas registrada, utilizadas para definir modelos segundo as notas a serem lançadas no sistema ao longo do ano e as disciplinas cursadas para serem criados modelos diferentes para cada disciplina do Ensino Fundamental e Médio.

3.2.1 Extração de dados

Para o desenvolvimento do sistema, foram obtidos os dados de notas de avaliações de alunos do Ensino Fundamental e do novo Ensino Médio das escolas Serviço Social da Indústria de Alagoas(SESI) presentes no banco de dados disponibilizado pela FIEA. Esses dados são disponibilizados a partir do DW, utilizado para armazenar os dados que foram coletados e limpos para serem feitas análises e aplicadas em sistemas. Os dados foram inseridos no DW a partir dos dados presentes no Sistema de Gestão Educacional (SGE) das escolas SESI das séries do Ensino Fundamental, 6º, 7º, 8º e 9º anos, e Ensino Médio, abrangendo o 1º, 2º e 3º anos do novo Ensino Médio, no qual o Ensino Médio se divide em duas categorias, Ensino Médio Base, equivalente a disciplinas-base do Ensino Médio, e Ensino Médio por Unidades Curriculares(UCs), equivalente a disciplinas profissionalizantes.

3.2.2 Marcos avaliativos

Uma vez que os dados de notas de avaliações são inseridos ao longo do ano, foram definidos modelos consistindo em listas de quais avaliações foram utilizadas para as predições no momento. Essas definições foram feitas em marcos avaliativos e cada marco avaliativo é associado a um modelo que possui como entrada as notas específicas de cada trimestre

e avaliação. As avaliações são realizadas ao longo de três trimestres, nos quais cada trimestre consistem em três avaliações, uma avaliação final do trimestre e uma reavaliação do trimestre.

As fases de predições se dividem em cinco marcos avaliativos criados de forma que o resultado do algoritmo indique o marco atual, assim, pode-se determinar qual modelo, com notas específicas definidas, foi utilizado na predição apresentada no resultado do algoritmo, em conjunto com os resultados de classificação da predição como 'Aprovado' ou 'Reprovado' e a probabilidade da predição correspondente. A figura 3.3 mostra como são definidos os marcos avaliativos ao longo do ano.

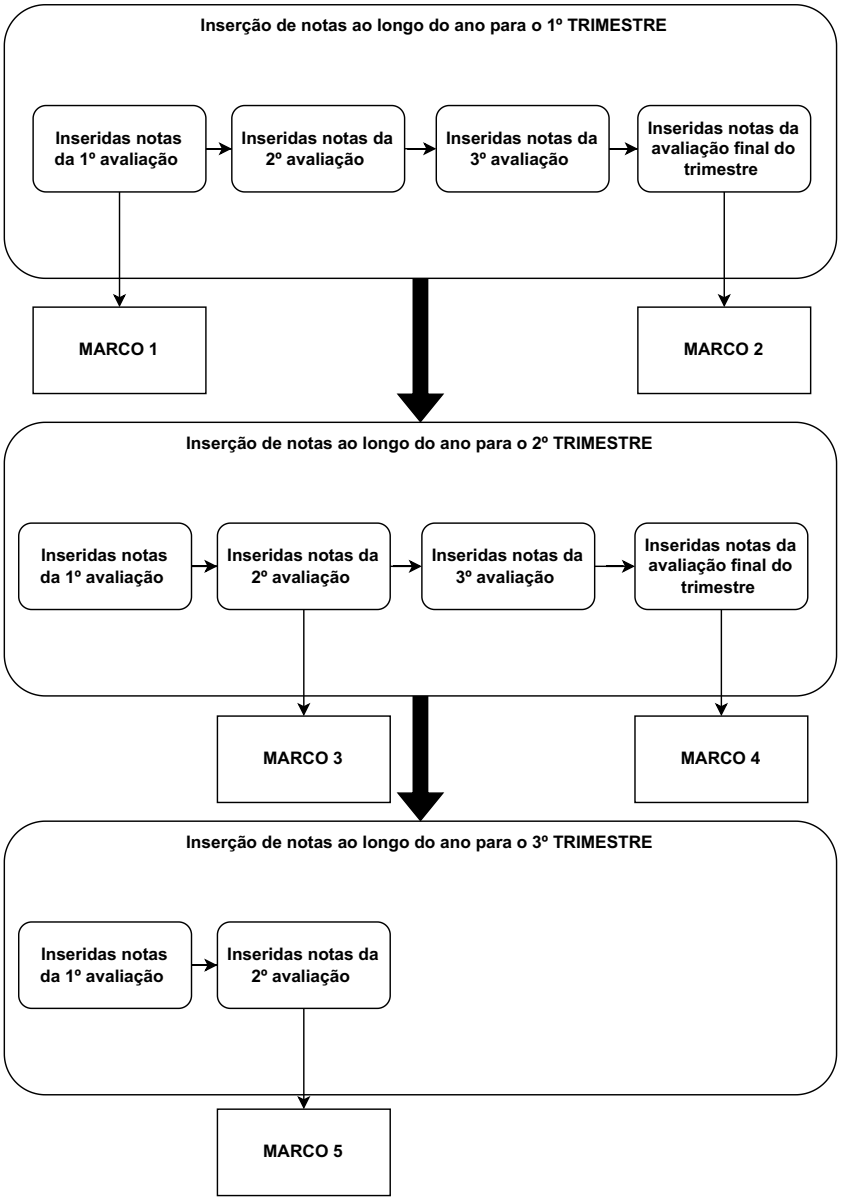

Figura 3.3: Marcos Avaliativos.

Ao longo dos períodos de adição de notas no sistema, são considerados novos modelos

dependendo da avaliação adicionada, assim os marcos avaliativos correspondem a:

- MARCO 1 → Predições do Modelo 1.
- MARCO 2 → Predições do Modelo 2.
- MARCO 3 → Predições do Modelo 3.
- MARCO 4 → Predições do Modelo 4.
- MARCO 5 → Predições do Modelo 5.

Dessa forma, são utilizadas todas as avaliações, exceto reavaliações, referentes a todo o primeiro e segundo trimestre. No caso do terceiro trimestre, são utilizadas as notas referentes somente as duas primeiras avaliações devido a que as demais avaliações desse trimestre estão muito próximas do fim do ano letivo e com isso a nota mais próxima ao fim do ano que foi definida para se ter um indicativo precoce de reprovação foi a da segunda avaliação do terceiro trimestre.

É necessário destacar que, existe uma peculiaridade quanto ao Marco 1, na fase de obtenção de dados para esse marco, para cada nota obtida em relação às notas sinalizadas como pertencente a esse modelo, é feita uma obtenção a parte de dados do ano anterior, ou seja, para alunos que não são novatos e estão no 7º, 8º, 9º ano do Ensino Fundamental e 2º e 3º ano do novo Ensino Médio, são obtidos os dados de notas da disciplina a ser avaliada do ano anterior para esse estudante, visto que, assim se pode ter dados histórico diretamente relacionado ao estudante. Para os demais estudantes, é repetida a nota do marco atual para preenchimento dos dados. Essas condições foram estabelecidas inicialmente por que o primeiro marco engloba somente uma nota do aluno, a primeira avaliação, então, se viu a necessidade de adicionar mais dados ao treinamento relacionado ao estudante. Ainda, em testes iniciais, o Marco 1 apresentou desempenho inferior aos demais marcos. Dessa forma, para alunos novatos, ou seja, que não apresentam informações de anos anteriores no sistema, a nota do primeiro marco corrente é repetida e então é refletido somente o desempenho atual aluno. São considerados novatos alunos sem registro, alunos ingressantes do 6º ano do Ensino Fundamental e alunos ingressantes do 1º ano do Ensino Médio, em vista que, mesmo que o aluno tenha cursado o 9º ano em uma escola SESI, o desempenho relacionado ao Ensino Fundamental e Médio deve ser diferenciado em vista que as disciplinas são diferentes. Para os demais alunos, são considerados veteranos, tendo sido utilizadas as notas presentes no sistema. No caso haver alguma nota faltante, essa é preenchida com a média das notas existentes.

Vale ressaltar que, após serem obtidos os dados presentes no sistema, para as transformações, normalizações e padronizações consideradas, os dados do Ensino Fundamental utilizados para a criação dos modelos preditivos foram de 2017 a 2022 para serem feitas as predições quanto aos alunos matriculados em 2023, porém para o novo Ensino Médio,

só foram possíveis obter dados referentes ao ano de 2022, devido à falta de registros em anos anteriores que atendessem as definições feitas.

Ainda, não foram utilizados dados pessoais dos estudantes a fim de evitar vieses, de modo que, a única informação contida para identificação de um estudante foi um identificador numérico único, e as demais informações, além das notas, foram as ligadas a identificação de séries, turmas e disciplinas.

3.3 Pré-processamento

Os dados foram divididos em duas categorias, séries do Ensino Fundamental e do novo Ensino Médio. Em ambos os casos, as séries foram divididas em disciplinas, onde para cada disciplina foi criado um modelo preditivo. Assim, são utilizados os dados históricos dos anos anteriores ao corrente para cada disciplina, onde o rótulo que define se o estudante foi aprovado ou reprovado na disciplina é determinado nessa fase baseado na nota final do estudante na disciplina.

A figura 3.4 apresenta o fluxograma de definição da nota final do aluno. Essas considerações foram definidas a fim de refletir o desempenho do estudante quanto às notas de avaliações da melhor forma possível, desconsiderando as aprovações de estudantes feitas por notas de conselho de classe. O conselho de classe foi desconsiderado devido ao modo que essa nota é registrada, já que é feita pelo professor quando o estudante não obtém a nota necessária para ser aprovado na disciplina, porém o professor decide aprovar o estudante baseado em outros fatores analisados em sala de aula. Logo, é feita uma análise em que se for visto que o estudante tem uma nota de conselho de classe existente para a disciplina, significa que o mesmo foi aprovado baseado em outros fatores, e assim, essa nota não deve ser utilizada pelo algoritmo para definir aprovação. Portanto, se a nota do conselho de classe não estiver preenchida, a nota final do estudante será baseada em sua última avaliação, que pode ter sido a recuperação final da disciplina ou a média parcial anual da disciplina.

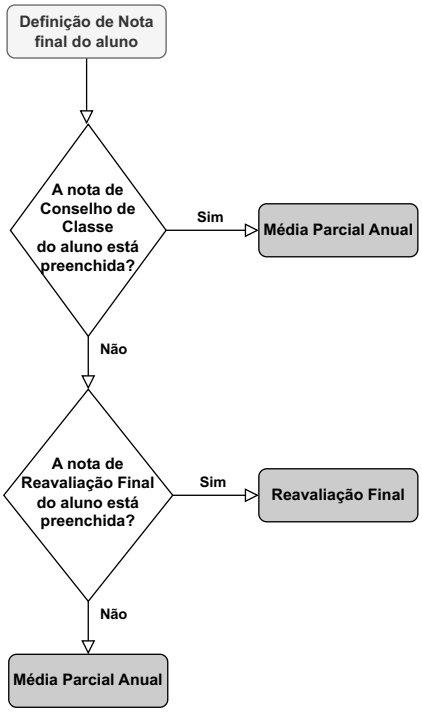

Figura 3.4: Fluxo de definição da Nota final.

Dessa forma, as seguintes considerações foram feitas para determinar a nota final do aluno e definir a aprovação ou reprovação:

- Se a nota do Conselho de Classe estiver preenchida:
  - Nota final = Média parcial anual da disciplina
- Se a nota do Conselho de Classe não estiver preenchida:
  - Se a nota da reavaliação final estiver preenchida:
    - Nota final = Reavaliação final
  - Se a nota da reavaliação final não estiver preenchida:
    - Nota final = Média parcial anual da disciplina

Além disso, esses dados passam por uma limpeza quanto a eliminação de dados faltantes e de dados repetidos. Para cada disciplina, são retirados os dados faltantes sinalizados como nulos ou *NaN(Not a Number)*, além disso, dados repetidos como registros duplicados de notas de um estudante na mesma avaliação para a mesma turma foram retirados.

Para maximizar a inclusão de dados na base de treinamento, foram implementadas transformações nos dados, além da necessidade de normalizá-los. Nesse contexto, tiveram que ser feitas associações de etapas de anos anteriores para um formato em comum,

visto que ao longo dos anos as formas de registro de notas foram sendo modificadas no sistema do SGE. Quanto à normalização dos dados, os valores das notas obtidas foram normalizados para que todos os valores ficassem com um valor de zero a dez e para serem consideradas notas de avaliações.

3.4 Mineração de dados

Na fase de mineração de dados, os dados recebidos após o pré-processamento passaram pelo processo de *oversampling* a fim de balancear os dados quanto à quantidade de aprovações e reprovações. Nas diferentes disciplinas, a taxa de reprovação variou entre 9% e 26% dos estudantes, sendo os demais considerados aprovados, então, a necessidade desse processo surgiu porque a quantidade de alunos aprovados foi observada como superior à quantidade de alunos reprovados. Para evitar qualquer viés na aprendizagem dos padrões pelos classificadores, optou-se por balancear esses dados. Isso envolveu um ajuste no qual alunos reprovados foram adicionados à disciplina, para igualar a quantidade de alunos aprovados.

Para a criação do modelo preditivo foram feitos testes comparativos segundo as métricas de acurácia, revocação, precisão e f1-score. Os algoritmos testados foram o KNN, árvore de decisão, *Random Forest* e XGBoost. Esses foram escolhidos nos testes iniciais feitos a partir do uso da biblioteca *Lazy Prediction* do *Python*, que disponibiliza os resultados de testes com parâmetros-base de diversos algoritmos.

Para os demais algoritmos, estão sendo utilizados os parâmetros-base definidos pela biblioteca do *Python* da qual os algoritmos são implementados já que esses foram vistos como adequados para a resolução do problema e, o *Grid Search*, por ser uma busca exaustiva e iterativa, tem um longo tempo de execução para o *Random Forest* e o XGBoost, os quais são algoritmos de árvore e apesar da sua rápida execução, possuem diversos parâmetros a serem testados, então, não foi visto como vantajoso usar o *Grid Search* nesses casos.

As tabelas 3.1 e 3.2 mostram os parâmetros testados utilizando o *Grid Search* para o *Random Forest* e XGBoost respectivamente e os valores considerados para cada parâmetro a fim de definir os valores ótimos de cada. Com o *Grid Search* foi possível fazer o teste dos modelos criados para os valores definidos, no qual os parâmetros ótimos oscilaram entre os valores testados, variando entre as disciplinas.

Tabela 3.1: Parâmetros testados no *Grid Search* para o *Random Forest*

Parâmetro	**Valores**
criterion	'gini', 'entropy' ou 'log_loss'
max_depth	De 1 a 30

Tabela 3.2: Parâmetros testados no *Grid Search* para o XGBoost

Parâmetro	**Valores**
min_child_weight	1, 5 ou 10
gamma	0,5; 1; 1,5; 2 ou 5
subsample	0,6; 0,8 ou 1
colsample_bytree	0,6; 0,8 ou 1
max_depth	De 1 a 31

Quanto aos dados de entrada dos modelos, foram divididos em base de treino e de teste, no qual houve uma divisão de dados com 80% na base de treino e 20% na base de teste, e então, foi feito *oversampling* somente na base de treino para balanceamento dos dados das classes de aprovação e reprovação. Ainda, as amostras de treino e teste foram divididas de modo estratificado, no qual cada conjunto contém aproximadamente a mesma porcentagem de amostras de cada classe alvo da base de dados de entrada.

A fim de ser feita a avaliação dos modelos criados a partir dos algoritmos classificatórios, foram obtidos e avaliados os resultados de métricas de acurácia e F1-Score, assim como os desvios padrão das médias dessas métricas calculadas para se chegar a conclusões dos resultados.

Os resultados obtidos são salvos no DW e introduzidos em um painel onde a equipe pedagógica do SESI pode visualizar essas informações de risco de reprovações dos alunos para então serem aplicadas intervenções definidas pela equipe em conjunto com os professores das escolas SESI.

Capítulo 4

Resultados e Discussão

O foco da avaliação do desempenho de cada classificador é saber qual deles consegue melhor identificar e classificar um estudante como pertencente ao rótulo de 'Reprovação', e isso é indicado primeiramente com a acurácia e em seguida com os verdadeiros positivos, para esse problema, falsos positivos não são tão graves quanto falsos negativos se tendo em mente que a classe positiva é a 'Reprovação' visto que é a procurada pelo problema.

Primeiramente, tem-se os resultados em relação aos marcos avaliativos que utilizam diferentes notas dos alunos que são adicionadas ao longo do ano. Até o momento de conclusão deste trabalho, só foi possível realizar a obtenção de dados relacionados as notas que vão até o quarto marco avaliativo do ano, visto que o quinto marco avaliativo condiz a notas próximas as registradas ao fim do ano.

Foram criados modelos preditivos para cada disciplina de cada turma para cada marco, como se tem sete séries, compostas por aproximadamente dez disciplinas, sendo obtidos resultados para quarto marcos avaliativos, foram criados cerca de duzentos modelos para cada classificador. Para avaliar os valores obtidos para cada métrica, realizou-se o teste de Mann-Whitney visando comparar conjuntos de resultados de desempenho dois a dois. No caso do KNN e da árvore de decisão, o teste foi inconclusivo ($p = 0{,}1963 > 0{,}05$), indicando que pode não haver diferença entre as distribuições das duas amostras. No entanto, para o KNN e o *Random Forest* ($p = 0{,}0001224 < 0{,}05$) e para o KNN e o XGBoost ($p = 0{,}0007437 < 0{,}05$), o teste indicou significância estatística. Adicionalmente, para a árvore de decisão e o *Random Forest* ($p = 0{,}01134 < 0{,}05$) e a árvore de decisão e o XGBoost ($p = 0{,}04241 < 0{,}05$), o teste indicou significância estatística. Por fim, para o *Random Forest* e o XGBoost ($p = 0{,}588 > 0{,}05$), o teste foi inconclusivo. Como se os resultados são extensos, como método de síntese desses resultados, empregou-se a média aritmética, dependendo do aspecto específico que se desejava avaliar.

Portanto, para o cálculo da média para os marcos avaliativos, foram obtidos os valores de métricas de cada disciplina por série, e então a partir desses valores, foi definida a média de cada marco de cada série do Ensino Fundamental e Médio Base. Logo, obtivemos os valores de médias aritméticas dos marcos de cada série. Em seguida, foi feito um novo

cálculo de média conforme os valores das métricas de cada série, sendo obtido assim, os valores de média de avaliação de cada marco englobando todas as séries.

As tabelas 4.1, 4.2, 4.3, 4.4 apresentam resultados das médias de acurácia e médias de F1-Score obtidas a partir dos resultados por turma gerados pelos testes realizados com os classificadores KNN, Árvore de Decisão, *Random Forest* e XGBoost receptivamente. Os valores obtidos indicam que ao longo dos marcos avaliativos, as métricas estão melhorando, o que condiz com a inserção de novas variáveis. Os testes de Mann-Whitney revelam diferenças estatisticamente significativas entre alguns classificadores, com uma tendência para valores superiores no Random Forest e XGBoost em comparação com a árvore de decisão e KNN.

Além disso, com base nas médias dos marcos obtidas, é evidente que o Random Forest e o XGBoost apresentam desempenhos semelhantes, uma vez que não mostraram diferenças estatisticamente significativas entre eles. Esses dois classificadores destacaram-se dos demais, exibindo também um menor desvio padrão, indicando uma menor dispersão em relação à média aritmética e, portanto, uma consistência mais robusta nos resultados.

Tabela 4.1: Avaliação de classificadores por marco — KNN

Classificador	Marco	Média das acurácias	Desvio padrão das acurácias	Média dos F1-Score	Desvio padrão dos F1-Score
KNN	Marco 1	79,65%	0,07931	79,65%	0,07931
KNN	Marco 2	82,00%	0,06723	82,00%	0,06723
KNN	Marco 3	86,29%	0,07285	86,29%	0,07285
KNN	Marco 4	88,84%	0,05806	88,84%	0,05806

Tabela 4.2: Avaliação de classificadores por marco — Árvore de decisão

Classificador	Marco	Média das acurácias	Desvio padrão das acurácias	Média dos F1-Score	Desvio padrão dos F1-Score
Árvore de decisão	Marco 1	78,60%	0,10975	78,60%	0,10975
Árvore de decisão	Marco 2	83,61%	0,06280	83,61%	0,06280
Árvore de decisão	Marco 3	88,21%	0,07323	88,21%	0,07323
Árvore de decisão	Marco 4	90,24%	0,04925	90,24%	0,04925

Tabela 4.3: Avaliação de classificadores por marco — *Random Forest*

Classificador	Marco	Média das acurácias	Desvio padrão das acurácias	Média dos F1-Score	Desvio padrão dos F1-Score
Random Forest	Marco 1	79,81%	0,09833	79,81%	0,09833
Random Forest	Marco 2	84,65%	0,05190	84,65%	0,05190
Random Forest	Marco 3	89,46%	0,05043	89,46%	0,05043
Random Forest	Marco 4	92,28%	0,03178	92,28%	0,03178

Tabela 4.4: Avaliação de classificadores por marco — XGBoost(Sem *Grid Search*)

Classificador	Marco	Média das acurácias	Desvio padrão das acurácias	Média dos F1-Score	Desvio padrão dos F1-Score
XGBoost(base)	Marco 1	79,48%	0,10326	79,48%	0,10326
XGBoost(base)	Marco 2	84,52%	0,05283	84,52%	0,05283
XGBoost(base)	Marco 3	89,61%	0,05575	89,61%	0,05575
XGBoost(base)	Marco 4	91,95%	0,04166	91,95%	0,04166

Observação: 'base' indica o uso de paramêtros-base.

Vale destacar que, em testes iniciais, o Marco 1 em todos os modelos apresentou desempenho inferior aos demais marcos, foram adicionados dados de notas equivalentes ao quinto marco do ano anterior para serem feitas as predições.

Foram aplicados testes de *Mann-Whitney* para as amostras correspondentes aos resultados de métricas antes e depois da adição de notas ao primeiro marco avaliativo. Para o KNN ($p = 0{,}8748 > 0{,}05$), a árvore de decisão ($p = 0{,}4082 > 0{,}05$), o *Random Forest* ($p = 0{,}1251 > 0{,}05$), o XGBoost(GS) ($p = 0{,}2358 > 0{,}05$) e o XGBoost(base) ($p = 0{,}1865 > 0{,}05$), os teste foram inconclusivos para todos os algoritmos, indicando que pode não haver diferença entre as distribuições das amostras. Porém, foi visto um melhoramento nas métricas para a maioria dos classificadores em relação às médias aritméticas. Assim, além disso, foi preferível manter a adição de notas, devido a se ter uma maior quantidade de atributos para o treinamento dos modelos e se ter dados históricos específicos de cada estudante.

A tabela 4.5 apresenta o desempenho geral dos classificadores no marco 1 quanto a antes da adição de notas passadas.

Tabela 4.5: Avaliação de classificadores para o Marco 1 antes da adição de notas passadas

Classificador	Média das acurácias por disciplina	Desvio padrão das acurácias	Média dos F1-Score por disciplina	Desvio padrão dos F1-Score
KNN	83,84%	0,05130	83,84%	0,05130
Árvore de decisão	77,16%	0,11563	77,16%	0,11563
Random Forest	77,78%	0,10982	77,78%	0,10982
XGBoost(GS)	77,80%	0,10403	77,80%	0,10403
XGBoost(base)	77,39%	0,11298	77,39%	0,11298

Observação: 'GS' indica o uso de *Grid Search* para definir parâmetros ótimos. E, 'base' indica o uso de paramêtros-base.

Nesse caso o KNN, apresentou melhores valores de acurácia e desvio padrão, sendo o único classificador a ter um desvio padrão inferior a 10% da média aritmética.

Em seguida, a tabela 4.6 apresenta a síntese de valores do Marco 1 após a adição de notas.

Tabela 4.6: Avaliação de classificadores para o Marco 1 após a adição de notas passadas

Classificador	Média das acurácias por disciplina	Desvio padrão das acurácias	Média dos F1-Score por disciplina	Desvio padrão dos F1-Score
KNN	79,65%	0,07931	79,65%	0,07931
Árvore de decisão	78,60%	0,10975	78,60%	0,10975
Random Forest	79,81%	0,09833	79,81%	0,09833
XGBoost(GS)	79,65%	0,099495	79,65%	0,099495
XGBoost(base)	79,48%	0,10326	0,794866	0,10326

Observação: 'GS' indica o uso de *Grid Search* para definir parâmetros ótimos. E, 'base' indica o uso de paramêtros-base.

O KNN foi o único classificador que apresentou desvio padrão abaixo de 10% da média aritmética para esse marco, porém houve uma diminuição no valor da acurácia.

Após esse processo, a partir das médias por marco de todas as séries, esses valores foram utilizados para calcular a média geral de cada classificador. Então, as médias gerais foram obtidas a partir do cálculo da média dos marcos por série.

Tabela 4.7: Avaliação de classificadores

Classificador	Média geral das acurácias	Desvio padrão das acurácias	Média geral dos F1-Score	Desvio padrão dos F1-Score
KNN	85,15%	0,0762	85,15%	0,0762
Árvore de decisão	85,83%	0,0689	85,83%	0,0689
Random Forest	87,60%	0,0814	87,60%	0,0814
XGBoost(GS)	86,97%	0,0709	86,97%	0,0709
XGBoost(base)	86,39%	0,06249	86,39%	0,06249

Observação: 'GS' indica o uso de *Grid Search* para definir parâmetros ótimos. E, 'base' indica o uso de paramêtros-base.

Os valores de F1-Score foram muito próximos dos valores de acurácia, isso ocorre devido ao balanceamento da base de treino.

Os resultados obtidos apontam que os melhores modelos foram gerados pelos classificadores *Random Forest* e XGBoost, com resultados próximos. Por marco, como apresentado na tabela 4.4, o XGBoost(base) foi o algoritmo com melhor desempenho. O próximo passo após a obtenção dos resultados de desempenho foi a escolha definitiva do algoritmo classificatório que acabou sendo o XGBoost. Isso é justificado pelos resultados de desempenho e também pela eficiência do seu uso e superioridade ao *Random Forest* em casos de *overfitting*, que pode acontecer com esses dados visto que a base pode não apresentar uma quantidade de notas para algumas disciplinas. Além disso, o XGBoost apresentou menor desvio padrão, de aproximadamente 6,24% em relação à média aritmética em comparação com 8,14% do *Random Forest*, indicando que o XGBoost teve uma menor dispersão quanto as médias aritméticas de desempenho obtidas. Ainda, foi definido que a velocidade de acesso dos resultados de saída é de importância para os usuários do sistema, tendo sido analisado que o XGBoost utilizando *Grid Search* é eficiente, porém, o uso de *Grid Search* por ser uma busca exaustiva causa o algoritmo a ter uma alta duração de tempo, em contrapartida, com o XGBoost base que mais rápido, mas ainda assim, o desempenho dos dois métodos tem valores aproximados então foi definido como algoritmo da versão final o XGBoost com parâmetros-base.

Assim, o XGBoost com parâmetros-base apresentou bom bons resultados de métricas de avaliação em conjunto com um desvio padrão menor do que os demais algoritmos, indicando que se tem menos dispersão em relação à média aritmética dos desempenhos por disciplina, apresentando menor variabilidade e maior consistência nos resultados, além disso, o XGBoost foi escolhido como o algoritmo classificatório para a versão final do sistema devido a sua simplicidade, escalabilidade e desempenho quanto a evitar o *overfitting* e criação de padrões com viés que devem ser evitados.

Nesse contexto, o XGBoost tem se mostrado com um impacto altamente reconhecido no qual, de acordo com Chen and Guestrin (2016), é destacado que esse algoritmo foi bastante utilizado em soluções no blog do Kaggle e em 2015, o blog publicou que 17 das 29 soluções vencedoras de suas competições usaram XGBoost em seus modelos. Ainda, o XGBoost em comparação com o *Random Forest*, apresenta uma melhora iterativa da função de perda, fazendo assim ajustes conforme a criação das árvores invés de manter

paramêtros fixos ao longo do treinamento. O XGBoost realiza poda das árvores mediante definições de ganho as árvores parem de serem construídas para evitar *overfitting*. O XGBoost da importância a redução de custo do modelo.

Assim, o XGBoost com parâmetros-base foi escolhido como algoritmo classificador na versão final do sistema, visto que apresentou resultados sólidos nas métricas de avaliação, acompanhados de um desvio padrão menor em comparação com os demais algoritmos, além de ser escalável, ter eficácia na prevenção de overfitting e na criação de padrões com viés, aspectos cruciais para serem evitados.

Os resultados obtidos ao final do sistema apresentam informações relacionadas a identificação dos estudante com um número identificador único, assim como a turma a que pertence, a série, a disciplina na qual foi feita a predição, a predição obtida, os valores de probabilidade associados a predição, o marco avaliativo associado àquela predição e o ano a qual a predição está ligada. A tabela 4.8 exemplifica a saída do sistema.

Tabela 4.8: Exemplos de resposta de saída do sistema preditivo

Estudante	**Série-Turma**	**Disciplina**	**Predição**	**Probabilidade**	**Marco**	**Ano**
001	6º ANO-D	Matemática	APROVADO	0,885	MARCO 1	2023
002	6º ANO-B	Matemática	REPROVADO	0,630	MARCO 2	2023
003	2º ANO-C	Matemática e suas tecnologias	APROVADO	0,886	MARCO 2	2023

Nesse exemplo, diferentes estudantes são apresentados na tabela de resultados gerais gerados pelo sistema, as informações apresentadas permitem se ter noções de quantos alunos estão indicados pelo sistema como reprovados e a probabilidade de reprovação de todos os estudantes matriculados no momento da obtenção dos resultados, dessa forma, podem ser feitos filtros quanto a quantidade de estudantes com risco de reprovação por série, turma, marco e ano.

Nesse contexto, outro exemplo do funcionamento da ferramenta poderia ocorrer da seguinte maneira: inicialmente, é realizada a predição de reprovação para um estudante no Marco 1, utilizando as respectivas notas desse marco. Posteriormente, ao incorporar as notas necessárias para a previsão no Marco 2, os resultados são gerados para ambos os marcos, ou seja, para o Marco 1 e para o Marco 2. Isso inclui tanto a classificação de reprovação ou aprovação quanto a probabilidade de reprovação correspondente em cada um dos marcos avaliativos. Esse processo oferece uma perspectiva abrangente das probabilidades de reprovação do aluno ao longo do ano, analisando os indicadores dos diferentes marcos avaliativos. A tabela 4.9 mostra um exemplo com informações fictícias de um estudante para melhor entendimento dos resultados dos marcos.

Tabela 4.9: Exemplos de resposta de saída do sistema preditivo para um estudante

Estudante	Série-Turma	Disciplina	Predição	Probabilidade	Marco	Ano
001	9º ANO-C	Língua Portuguesa	APROVADO	0,8	MARCO 1	2023
001	9º ANO-C	Língua Portuguesa	REPROVADO	0,6	MARCO 2	2023
001	9º ANO-C	Língua Portuguesa	APROVADO	0,6	MARCO 3	2023
001	9º ANO-C	Língua Portuguesa	REPROVADO	1,0	MARCO 4	2023
001	9º ANO-C	Língua Portuguesa	REPROVADO	0,8	MARCO 5	2023

A partir desses resultados, se pode ter uma projeção do risco de reprovação do estudante ao longo do ano. A figura 4.1 mostra graficamente baseado nos resultados da tabela 4.9.

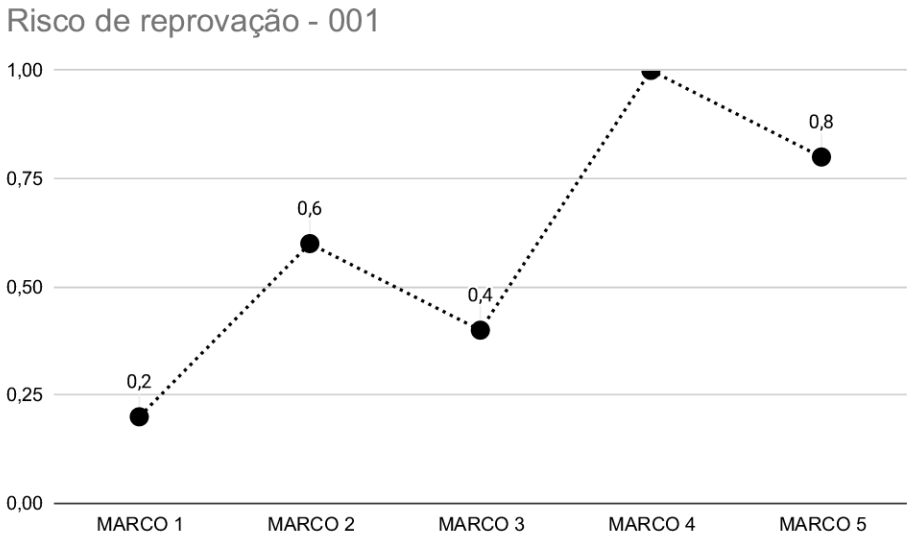

Figura 4.1: Exemplo de projeção para o risco de reprovação de um estudante ao longo do ano.

Dessa forma, para cada estudante, pode-se ter uma síntese do seu risco de reprovação ao longo do ano.

Capítulo 5

Conclusão

O presente trabalho propôs uma abordagem quanto ao desenvolvimento de um sistema preditivo de reprovação de estudantes do Ensino Básico em base das notas de avaliações registradas em disciplinas para as séries do Ensino Fundamental e do novo Ensino Médio. Para isso, foram obtidas as notas dos estudantes dos anos de 2017 a 2023 das escolas SESI Alagoas por meio do processo do KDD, que consistiu em cinco fases, seleção de dados, pré-processamento dos dados selecionados, transformações de dados, mineração de dados e análise e interpretação dos resultados. Em destaque, se tem a fase de mineração de dados, na qual foi implementada após as notas de avaliações dos estudante nas disciplinas passarem por transformações, padronizações e normalizações para seguir um padrão desejado e utilizado atualmente nas escolas SESI, para então serem criados modelos preditivos a partir dos dados balanceados para cada disciplina conforme os padrões históricos presentes nas notas registradas e aprendidos a partir de aprendizagem de máquina com a utilização de algoritmos classificadores de aprendizagem supervisionada. Dito isso, os algoritmos escolhidos apresentaram resultados de métricas de avaliação de desempenho, onde esses resultados foram sintetizados a partir do cálculo de médias aritméticas dos valores de desempenho por disciplina sendo obtido para o KNN 85,5% de média geral de acurácia e F1-Score, para a árvore de decisão 85,5% de média geral de acurácia e F1-Score, para o Random Forest 87,6% de média geral de acurácia e F1-Score, para o XGBoost com parâmetros ótimos 86,9% de média geral de acurácia e F1-Score e para o XGBoost com parâmetros-base 86,3% de média geral de acurácia e F1-Score, em vista disso, o XGBoost com parâmetros-base apresentou bons resultados de métricas de avaliação em conjunto com um desvio padrão menor do que os demais algoritmos, indicando que se tem menos dispersão em relação à média aritmética dos desempenhos por disciplina, apresentando menor variabilidade e maior consistência nos resultados, além disso, o XGBoost foi escolhido como o algoritmo classificatório para a versão final do sistema devido a sua simplicidade, escalabilidade e desempenho quanto a evitar o *overfitting* e criação de padrões com viés que devem ser evitados. Sendo assim, foi construído o sistema preditivo utilizando o algoritmo classificatório XGBoost com aprendizagem supervisionada que cria

modelos preditivos para cada disciplina presente no sistema, sendo obtidos resultados de predições com classificação de 'Aprovado' ou 'Reprovado' assim como a probabilidade da predição para cada aluno matriculado nas disciplinas ao sistema ser executado.

Referências Bibliográficas

Al-Tameemi, R. A. N., Johnson, C., Gitay, R., Abdel-Salam, A.-S. G., Hazaa, K. A., Ben-Said, A., and Romanowski, M. H. (2023). Determinants of poor academic performance among undergraduate students—a systematic literature review. *International Journal of Educational Research Open*, 4:100232.

Albreiki, B., Zaki, N., and Alashwal, H. (2021). A systematic literature review of student' performance prediction using machine learning techniques. *Education Sciences*, 11(9).

Almasri, A., Celebi, E., and Alkhawaldeh, R. S. (2019). Emt: Ensemble meta-based tree model for predicting student performance. *Scientific Programming*, 2019:3610248.

Amal Asselman, M. K. and Aammou, S. (2021). Enhancing the prediction of student performance based on the machine learning xgboost algorithm. *Interactive Learning Environments*, 31(6):3360–3379.

Breiman, L. (2001). Random forests. *Machine Learning*, 45(1):5–32.

Chakarverti, M., Sharma, N., and Divivedi, R. (2019). Prediction analysis techniques of data mining: A review. *SSRN Electronic Journal*.

Chanlekha, H. and Niramitranon, J. (2018). Student performance prediction model for early-identification of at-risk students in traditional classroom settings. pages 239–245.

Chen, J., de Hoogh, K., Gulliver, J., Hoffmann, B., Hertel, O., Ketzel, M., Bauwelinck, M., van Donkelaar, A., Hvidtfeldt, U. A., Katsouyanni, K., Janssen, N. A., Martin, R. V., Samoli, E., Schwartz, P. E., Stafoggia, M., Bellander, T., Strak, M., Wolf, K., Vienneau, D., Vermeulen, R., Brunekreef, B., and Hoek, G. (2019). A comparison of linear regression, regularization, and machine learning algorithms to develop europe-wide spatial models of fine particles and nitrogen dioxide. *Environment International*, 130:104934.

Chen, T. and Guestrin, C. (2016). XGBoost. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM.

Crosnoe, R. (2002). High school curriculum track and adolescent association with delinquent friends. *Journal of Adolescent Research - J ADOLESCENT RES*, 17:143–167.

Deepthi, P. N., Anitha, R., and Swathi, K. (2019). Retracted: A review on bioinformatics using data mining techniques. *Journal of Physics: Conference Series*, 1228(1):012023.

Faria de Souza, V. and Cazella, S. C. (2022). Mineração de dados educacionais com algoritmos de regressão: um estudo sobre a predição do desempenho. *Revista Educar Mais*, 6:183–198.

Fayyad, U., Piatetsky-Shapiro, G., and Smyth, P. (1996). From data mining to knowledge discovery in databases. *AI Magazine*, 17(3):37.

Hart, A. (2001). Mann-Whitney test is not just a test of medians: differences in spread can be important. *BMJ*, 323(7309):391–393.

Hastie, T. (2020). Ridge regularization: An essential concept in data science. *Technometrics*, 62(4):426–433.

Louppe, G. (2015). Understanding random forests: From theory to practice.

Nasteski, V. (2017). An overview of the supervised machine learning methods. *HORIZONS.B*, 4:51–62.

Nirmala, I., Wijayanto, H., and Notodiputro, K. (2022). Prediction of undergraduate student's study completion status using missforest imputation in random forest and xgboost models. *ComTech: Computer, Mathematics and Engineering Applications*, 13:53–62.

Norvig, P. and Russell, S. (2014). *Inteligência artificial: Tradução da 3a Edição*. Elsevier Brasil.

Siddique, A., Jan, A., Majeed, F., Qahmash, A., Quadri, N. N., and Wahab, M. (2021). Predicting academic performance using an efficient model based on fusion of classifiers. *Applied Sciences*, 11:11845.

Singh, R. and Pal, S. (2020). Machine learning algorithms and ensemble technique to improve prediction of students performance. *International Journal of Advanced Trends in Computer Science and Engineering*, 9:3970–3976.

Song, Y.-Y. and Lu, Y. (2015). Decision tree methods: applications for classification and prediction. *Shanghai archives of psychiatry*, 27:130–5.

Souza, J. P. L. d., Dutra, J. F., and Fernandes, D. Y. D. S. (2022). Predição precoce de problemas de desempenho de estudantes em modalidade de educação on-line: um estudo de caso no ensino médio integrado. *Revista Principia - Divulgação Científica e Tecnológica do IFPB*, 59(3):764.

Taunk, K., De, S., Verma, S., and Swetapadma, A. (2019). A brief review of nearest neighbor algorithm for learning and classification. *2019 International Conference on Intelligent Computing and Control Systems (ICCS)*, pages 1255–1260.

Vujovic, Z. (2021). Classification model evaluation metrics. *International Journal of Advanced Computer Science and Applications*, Volume 12:599–606.

Zou, H. and Hastie, T. (2005). Regularization and Variable Selection Via the Elastic Net. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 67(2):301–320.