UNIVERSIDADE FEDERAL DE OURO PRETO
INSTITUTO DE CIÊNCIAS EXATAS E BIOLÓGICAS
DEPARTAMENTO DE COMPUTAÇÃO

GABRIEL CATIZANI FARIA OLIVEIRA
Orientador: Guilherme Tavares De Assis

PROPOSTA, DESENVOLVIMENTO E VALIDAÇÃO DE UMA ESTRATÉGIA PARA PREDIÇÃO DE DESEMPENHO DE DISCENTES NA PLATAFORMA TÔSABENDO

Ouro Preto, MG
2023

UNIVERSIDADE FEDERAL DE OURO PRETO
INSTITUTO DE CIÊNCIAS EXATAS E BIOLÓGICAS
DEPARTAMENTO DE COMPUTAÇÃO

GABRIEL CATIZANI FARIA OLIVEIRA

PROPOSTA, DESENVOLVIMENTO E VALIDAÇÃO DE UMA ESTRATÉGIA PARA PREDIÇÃO DE DESEMPENHO DE DISCENTES NA PLATAFORMA TÔSABENDO

Monografia apresentada ao Curso de Ciência da Computação da Universidade Federal de Ouro Preto como parte dos requisitos necessários para a obtenção do grau de Bacharel em Ciência da Computação.

Orientador: Guilherme Tavares De Assis

Ouro Preto, MG
2023

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE OURO PRETO
REITORIA
INSTITUTO DE CIENCIAS EXATAS E BIOLOGICAS
DEPARTAMENTO DE COMPUTACAO

FOLHA DE APROVAÇÃO

Gabriel Catizani Faria Oliveira

Proposta, desenvolvimento e validação de uma estratégia para predição de desempenho de discentes na plataforma TôSabendo

Monografia apresentada ao Curso de Ciência da Computação da Universidade Federal de Ouro Preto como requisito parcial para obtenção do título de Bacharel em Ciência da Computação

Aprovada em 15 de Fevereiro de 2024.

Membros da banca:

Guilherme Tavares De Assis (Orientador) - Doutor - Universidade Federal de Ouro Preto
Reinaldo Silva Fortes (Examinador) - Doutor - Universidade Federal de Ouro Preto
Rodrigo Geraldo Ribeiro (Examinador) - Doutor - Universidade Federal de Ouro Preto

Guilherme Tavares De Assis, Orientador do trabalho, aprovou a versão final e autorizou seu depósito na Biblioteca Digital de Trabalhos de Conclusão de Curso da UFOP em 15/02/2024.

Documento assinado eletronicamente por **Guilherme Tavares de Assis**, **PROFESSOR DE MAGISTERIO SUPERIOR**, em 15/02/2024, às 17:12, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

A autenticidade deste documento pode ser conferida no site http://sei.ufop.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0 , informando o código verificador **0667167** e o código CRC **8600D203**.

R. Diogo de Vasconcelos, 122, - Bairro Pilar Ouro Preto/MG, CEP 35402-163
Telefone: 3135591692 - www.ufop.br

Dedico este trabalho a todos os meus familiares, amigos e professores, que sempre me incentivaram nesse árdua jornada em me tornar um pesquisador e profissional na área de computação. Destacando especialmente meu orientador, Guilherme Tavares, por seu constante apoio e orientação ao longo desta desafiadora jornada. Dedico também à UFOP por proporcionar um ambiente propício ao meu crescimento tanto intelectual quanto pessoal.

Agradecimentos

Agradeço profundamente a Deus, a todos os meus familiares, amigos e professores, que sempre estiveram ao meu lado e me fizeram evoluir intelectualmente e pessoalmente.

Primeiramente a Deus, por ter me dado todas as forças para seguir em frente e por me ensinar a amar ao próximo,

Aos meus familiares, por cuidarem de mim toda minha vida, me demonstrando o primeiro amor. Meus pais por demonstrarem o caminho correto, explicando como a vida é feita de escolhas. Meus irmãos por me demonstrarem o qual árduo é o caminho do estudo e por serem meus primeiros amigos. Meus avôs, tios e primos pelos experientes ensinamentos.

Ao meu orientador Guilherme Tavares de Assis, por toda a paciência, dedicação e conselhos dados por inúmeras horas dedicadas para a conclusão deste trabalho.

Aos meus amigos do ensino fundamental e ensino médio, tanto do Santa Marcelina quanto do Bernoulli, por me mostraram o que é uma amizade verdadeira. Por me ensinarem a ser mais humano e que me deram forças e sentido para seguir com a vida.

Aos meus amigos de faculdade, por serem minha família de Ouro Preto e seguiram junto comigo nessa jornada complicada que foi para chegar a esse tão glorioso momento da formatura. Sem vocês nada disso seria possível.

Aos meus amigos do trabalho que me ensinaram toda experiência que levarei em minha vida profissional.

Por fim, e não menos importante, meu grande e querido amigo, filho e um dos maiores amores da minha vida, meu cachorro Fox! Infelizmente, não se encontra mais ao nosso lado, porém esteve comigo toda minha infância e durante quase todo meu período acadêmico. Quem me fazia levantar da cama com seu glorioso latido, pedindo para sair para comer e dar uma volta. Com certeza meu grande herói e um enviado de Deus. Muito obrigado Fox, por me demostrar o que é amor de verdade.

A imaginação é mais importante que o conhecimento. O conhecimento é limitado, enquanto a imaginação abraça o mundo inteiro. (Einstein, 1931)

Resumo

Modelos de predição que utilizam algoritmos de classificação, como árvores de decisão, KNNs, Naive Bayes e RNAs, são constantemente utilizados na educação para previsão de desempenho de discentes, seja no ensino médio ou no âmbito acadêmico, a fim de compreender como os mesmos podem sair-se em provas, exames ou atividades extracurriculares, a partir de uma série de fatores, como, por exemplo, notas passadas, sua idade, seu gênero, a duração de um semestre ou a disciplina que está cursando ou cursou. Uma das possíveis aplicações desses modelos de predição na educação é no ensino *online*. Assim, considerando o ensino *online*, a plataforma gamificada, TôSabendo, foi criada baseando-se em *Quizzes* (jogos de perguntas e respostas) no intuito de gerar experiências envolventes em Instituições de Ensino Superior buscando favorecer um ambiente de desafio para o jogador, motivando-o a aprender os conceitos apresentados em cada questão e dando a ele uma sensação de progressão naquela tarefa que está realizando. Entretanto, a mesma ainda não apresenta uma estratégia de predição utilizando modelos de predição a fim dos professores compreenderem, por meio do conhecimento previsto, como determinado discente pode se sair na plataforma, no intuito de aperfeiçoar o ensino e as atividades de conteúdos programáticos de disciplinas em sala de aula e também na própria TôSabendo. Assim, desejou-se propor, desenvolver e validar uma estratégia para predição de desempenho de discentes na plataforma TôSabendo. Com a estratégia de predição proposta e desenvolvida, para validá-la, foi realizada uma experimentação prática envolvendo diferentes modelos de predição e conjuntos de dados sintéticos, onde se avaliou inicialmente o modelo que se sairia melhor com 10 diferentes conjuntos de dados, um para discentes novatos e outro para veteranos. Em seguida, os modelos que obtiveram os melhores resultados nesse primeiro experimento, passaram por uma avaliação de diferentes hiperparâmetros. De uma forma geral, após as avaliações dos modelos, as árvores de decisão foram as que apresentaram os resultados mais satisfatórios tanto para novatos quanto para veteranos e, aperfeiçoando mais esse modelo com treinamentos com diferentes hiperparâmetros, foram obtidos resultados de acurácia e precisão próximos ou igual a 100% de acerto, um valor que deve ser analisado e avaliado futuramente por conta da necessidade da criação de dados sintéticos, o que sugere um possível *overfitting*.

Palavras-chave: Estratégia de predição. Modelos de predição. Plataforma TôSabendo. Desempenho de discentes.

Abstract

Prediction models that use classification algorithms, such as decision trees, KNNs, Naive Bayes, and ANNs, are consistently employed in education for predicting the performance of students, whether in high school or in academic settings. The goal is to understand how they may perform in exams, tests, or extracurricular activities based on various factors such as past grades, age, gender, semester duration, or the discipline they are currently enrolled in or have completed. One of the potential applications of these prediction models in education is in online learning. Considering online education, the gamified platform "TôSabendo" was created based on quizzes (question and answer games) with the aim of generating engaging experiences in Higher Education Institutions. The intention is to create a challenging environment for the player, motivating them to learn the concepts presented in each question and giving them a sense of progression in the task at hand. However, the platform currently lacks a prediction strategy using prediction models to help teachers understand, through predicted knowledge, how a particular student may perform on the platform. This understanding would be valuable for improving teaching methods and the content activities of classroom subjects, both in the traditional classroom setting and on the TôSabendo platform itself. Therefore, the goal was to propose, develop, and validate a strategy for predicting student performance on the TôSabendo platform. With the proposed and developed prediction strategy, a practical experimentation was conducted involving different prediction models and fictitious datasets. The evaluation initially assessed the model that would perform best with 10 different datasets, one for novice students and another for veterans. Subsequently, the models that achieved the best results in this first experiment underwent an evaluation of different hyperparameters. Overall, after evaluating the models, decision trees yielded the most satisfactory results for both novices and veterans. By further refining this model through training with different hyperparameters, accuracy and precision results close to or equal to 100% were obtained, a value that must be analyzed and evaluated in the future due to the need to create synthetic data, which suggests a possible overfitting.

Keywords: Prediction strategy. Prediction models. TôSabendo Plataform. Student Performance.

Lista de Ilustrações

Lista de Tabelas

Lista de Abreviaturas e Siglas

IES	Instituição de Ensino Superior
EDM	*Educational Data Mining*
RNA	Rede Neural Artificial
ANN	*Artificial Neural Network*
KNN	*K-Nearest Neighbors*
EER	Esquema Entidade-Relacionamento
ENADE	Exame Nacional de Desempenho de Estudantes
API	*Application Programming Interface*
IA	Inteligência Artificial
SIS	Sistema de Informação do Estudante (SIS)
OULAD	*Open University Learning Analytics Dataset*
ORM	*Object-Relational Mapping*
SGDB	Sistema de gerenciamento de banco de dados

Sumário

1 Introdução

A educação é um direito humano fundamental, que deve ser de acesso universal com qualidade para todos (Gomede *et al.*, 2018). Para melhorá-la, é importante encontrar outras formas eficazes e eficientes e a tecnologia na educação desempenha um papel crucial para isso, trazendo consigo uma série de benefícios e oportunidades para discentes, educadores e instituições de ensino. Desde o advento da tecnologia digital, testemunhou-se uma mudança significativa na forma como a informação é acessada, compartilhada e processada. De acordo com Giroto, Poker e Omote (2012), a tecnologia tem o poder de romper barreiras geográficas, democratizar o acesso ao conhecimento e proporcionar experiências de aprendizagem mais interativas e envolventes. Portanto, a tecnologia na educação promove a personalização do ensino. Segundo Sousa *et al.* (2011), com o auxílio de ferramentas digitais, é possível adaptar o conteúdo e a abordagem pedagógica de acordo com as necessidades individuais de cada discente. Isso permite que os discentes aprendam em seu próprio ritmo, explorando diferentes recursos e recebendo *feedback* imediato, otimizando o processo de aprendizagem e facilitando a compreensão dos conceitos (Valentini; Soares, 2005).

Um dos recursos didáticos personalizados que vem sendo utilizado frequentemente são elementos de jogos em instituições que buscam oferecer aos discentes maneiras de envolvê-los de forma mais eficaz no processo de aprendizagem dos conteúdos discutidos em sala de aula. Segundo Prensky (2001), a nova geração de discentes, conhecida como “nativos digitais”, possui uma afinidade natural com a tecnologia e experiências de jogos, o que torna a gamificação uma estratégia eficaz para capturar sua atenção e motivação. Os jogos promovem cenários educativos, “proporcionando ao discente a vivência de experiências de aprendizagem que talvez não fossem tão fáceis de serem alcançadas através do ensino tradicional” (GIARDINETTO; MARIANI, 2005). De acordo com (Seixas *et al.*, 2014), pesquisas a respeito da utilização dos games na educação têm evidenciado resultados positivos com relação à experiência de aprendizagem, bem como relatam melhorias significativas na motivação e, consequentemente, no engajamento.

Seguindo essa visão da gamificação no ensino, surgiu-se a ideia da criação de uma plataforma de ensino com participação dos discentes e professores do curso de ciência da computação da Universidade Federal de Ouro Preto (França *et al.*, 2021). Para este feito, foi desenvolvida e validada uma plataforma baseada em Quizzes, denominada TôSabendo, no intuito de gerar experiências envolventes em Instituições de Ensino Superior (IESs). Esse Quizzes buscam favorecer um ambiente de desafio para o jogador, motivando-o a aprender os conceitos apresentados em cada questão e dando a ele uma sensação de progressão naquela tarefa que está realizando (Ferreira, 2022).

Além da gamificação, para atualizar e melhorar a plataforma TôSabendo com a quantidade

de dados existentes dos discentes que a utilizariam diariamente, a combinação com modelos de predição é uma excelente alternativa. De acordo com Gerds, Cai e Schumacher (2008), Modelos de predição são algoritmos diretamente ligados ao Aprendizado de Máquina, que utilizam dados históricos para fazer previsões ou estimativas sobre eventos ou comportamentos futuros. Eles são construídos a partir de dados de treinamento, nos quais padrões e relações são identificados, e esses padrões são usados para fazer previsões ou classificações em novos conjuntos de dados (Carraro, 2019). Existem diversos modelos preditivos: Redes Neurais Artificais, KNNs, Árvores de decisão, Naive Bayes, dentre outros. Eles podem trazer benefícios significativos para o processo de aprendizagem como (Alyahyan; Düştegör, 2020):

- ***Feedback* personalizado**: com base na análise de dados, os modelos preditivos podem fornecer *feedback* personalizado e direcionado aos discentes. Isso vai além do *feedback* imediato fornecido pela gamificação, oferecendo informações específicas sobre as áreas em que o discente está se destacando ou precisa melhorar. De acordo com Pallathadka *et al.* (2023), o *feedback* personalizado ajuda os discentes a entenderem seus pontos fortes e fracos, fornecendo orientações claras sobre como podem melhorar seu desempenho;
- **Intervenções direcionadas**: podem identificar discentes que estão enfrentando dificuldades ou riscos de desempenho acadêmico inferior. Ao detectar padrões nos dados de desempenho, os modelos podem alertar os educadores sobre a necessidade de intervenções específicas para esses discentes. Segundo Karimi, Huang e Derr (2020), isso permite que os educadores ofereçam suporte adicional, tutoria ou materiais de aprendizagem complementares para auxiliar os discentes em áreas em que estão enfrentando desafios.

Assim, a implementação de um desses modelos de predição na plataforma TôSabendo pode auxiliar a entender os discentes, principalmente para prever o desempenho de futuros discentes cadastrados na mesma. O desempenho acadêmico está diretamente relacionado a *Educational Data Mining* (EDM), que recentemente tem se baseado em algoritmos de Aprendizado de Máquina em conjunto com mineração de dados, educação baseada em computação e análises estatísticas, como se pode visualizar na Figura 1.1. A EDM (Ji; Zhang; Zhang, 2020) refere-se a técnicas de mineração de dados usadas para analisar dados educacionais. Diversos dados, ou também chamados de fatores, estudados na EDM podem influenciar esse desempenho, como desempenho acadêmico prévio, dados demográficos desses discentes, fatores psicológicos, entre outros (Alyahyan; Düştegör, 2020). Portanto, esses dados fornecidos por essa extensa pesquisa, após pré-processá-los e tratá-los, são de grande importância para os modelos de predição e, consequentemente, para a plataforma.

Este capítulo encontra-se organizado como se segue. A Seção 1.1 apresenta a motivação para a realização desse trabalho. A Seção 1.2 descreve os objetivos geral e específicos. A Seção 1.3 aborda o método utilizado no desenvolvimento desse trabalho. Finalmente, a Seção 1.4 apresenta o delineamento do restante da monografia.

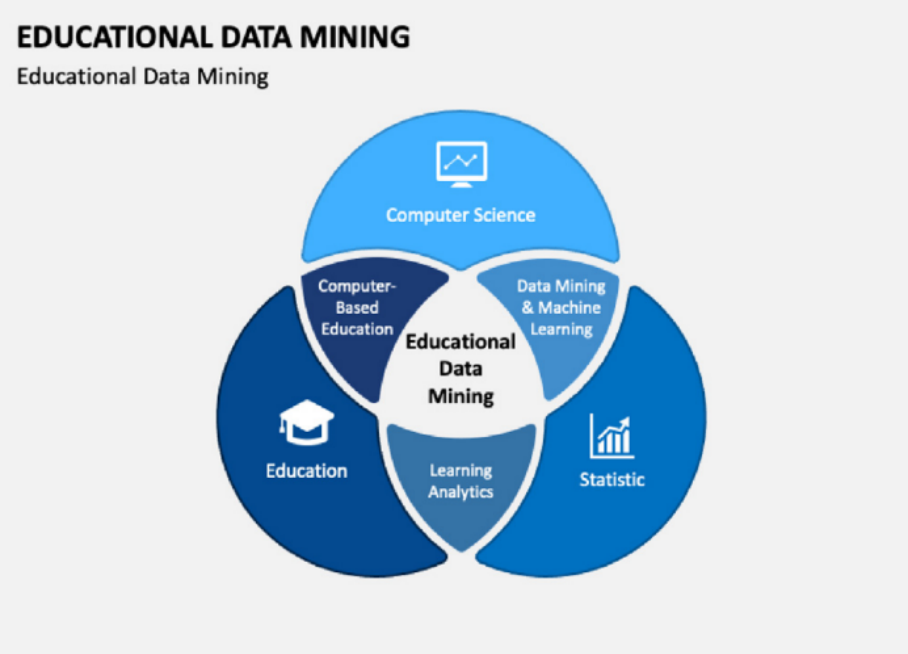

Figura 1.1 – Áreas de formação da EDM (Pallathadka *et al.*, 2023)

1.1 Justificativa

Ter conhecimento de como os discentes sair-se-ão em sua formação acadêmica é um dos requisitos mais significantes para as instituições de ensino. O desempenho dos discentes (Ji; Zhang; Zhang, 2020) pode ser antecipado com base em seu desempenho acadêmico anterior. De acordo com os resultados, as habilidades e interesses dos discentes também podem estar ligados ao seu desempenho. Esse tipo de análise permite aos professores concentrarem mais atenção nos discentes que mais precisam. É importante ressaltar que o sucesso de um professor é frequentemente medido pelo desempenho de seus discentes (Pallathadka *et al.*, 2023). A faculdade deve avaliar a competência do seu corpo docente por meio disso. Além disso, é importante entender em que nível se encontra um determinado curso. Esses tipos de análises ajudam uma instituição a melhorar a qualidade de seu ensino.

Apesar de existirem diversas ferramentas gamificadas para auxiliar os docentes a melhorar o desempenho de seus discentes como o

- **Kahoot**, orientada por meio de questionários e que apresenta benefícios notáveis na melhoria do aprendizado (Wang; Tahir, 2020),
- ou **Kolligo** (Wiener; Campos, 2019), que é um aplicativo para dispositivos móveis que promove o apoio aos professores na elaboração de práticas diferentes levando em consideração o perfil de cada turma,

nenhuma é focado em IESs e muito menos possuem análises para previsão do desempenho do discente utilizando modelos preditivos (Romero; Ventura, 2020). A grande importância

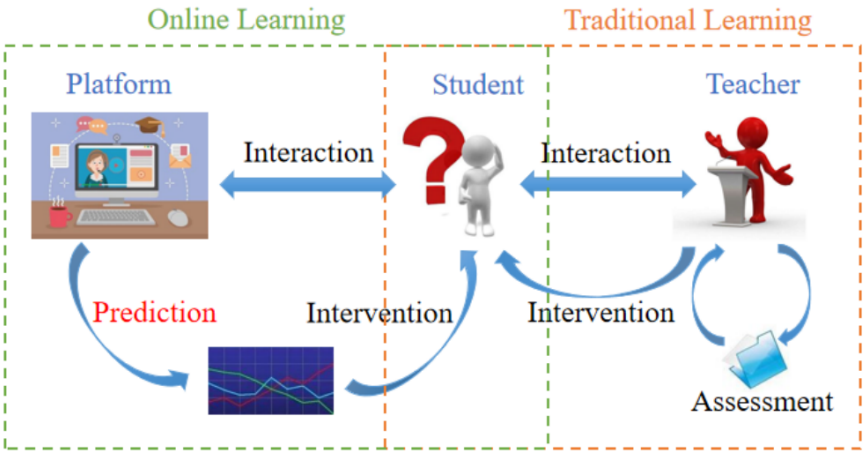

Figura 1.2 – Comparação visual do processo de aprendizagem/intervenção entre os sistemas de ensino *online* e tradicional (Karimi; Huang; Derr, 2020)

.

de combinar a plataforma TôSabendo com um modelo preditivo é explicado pela Figura 1.2, onde se observa que no ambiente de aprendizado tradicional, os instrutores podem interagir com os discentes, avaliar seu desempenho e agir para fornecer intervenção se sentirem que um discente provavelmente terá um desempenho ruim na aula. No entanto, no ensino *online* os discentes interagem principalmente com a plataforma, ou seja, há inerentemente menos interação entre discentes e instrutores. Mais especificamente, devido à alta proporção discente-professor, os professores não conseguem avaliar de forma abrangente o ganho de aprendizado de cada discente nessas plataformas (Karimi; Huang; Derr, 2020). Assim, um modelo preditivo poderia ser usado em tempo real ao longo do curso dentro da TôSabendo para identificar os discentes que têm previsão de desempenho ruim e fornecer a eles alguma intervenção com os recursos da própria plataforma. Isso de forma geral ajudaria os discentes em si, os próprios professores e, consequentemente, os próprios cursos e IESs.

1.2 Objetivos

Este trabalho possui, como objetivo geral, a proposta, o desenvolvimento e a validação de uma estratégia para predição de desempenho de discentes na plataforma TôSabendo, elegendo os melhores modelos de predição por meio de diversas análises dos dados pré-selecionados dos discentes, desde notas anteriores até o ambiente escolar em que vive. Uma vez a plataforma TôSabendo ficando novamente funcional, a estratégia de predição, proposta e validada neste trabalho, poderá ser incorporada à plataforma TôSabendo.

De um modo geral, os principais objetivos específicos, a serem alcançados neste trabalho, são:

- entendimento do processo de EDM em uma plataforma de ensino *online* com informações dos discentes;
- identificação prévia das dificuldades de discentes quanto a determinados assuntos ou conteúdos programáticos com futuras intervenções por parte dos professores;
- possibilidade de aplicação da plataforma com predição do desempenho dos discentes em distintas disciplinas e distintos cursos de IESs como forma de complementar o ensino.

1.3 Método de trabalho

Visando alcançar o objetivo geral deste trabalho, inicialmente, foi proposta uma estratégia de predição a partir da construção de arquiteturas de funcionamento, as quais foram adotadas para desenvolvimento e validação da estratégia. Em seguida, precisou-se definir os conjuntos de dados utilizados tanto para mineração dos dados como a construção de modelos preditivos. Para tanto, utilizou-se de dados sintéticos, criados a partir de estudos realizados (vide Seção 2.1), selecionando, assim, atributos relevantes para a experimentação prática da estratégia de predição proposta considerando os discentes novatos e veteranos. Em seguida, foi proposto uma mecanismo de mineração de dados para tratamento desses atributos relevantes utilizando algumas das etapas da EDM, como o pré-processamento de dados.

Com a modelagem desses conjuntos de dados, uma variedade de modelos de predição foi desenvolvida utilizando algoritmos de classificação. Para validação da estrategia de predição, inicialmente, esses modelos passaram por fases de treinamento e testes em conjuntos de dados diversos. Posteriormente, os modelos que apresentaram os melhores desempenhos, um destinado a novatos e outro a veteranos, foram aprimorados por meio de testes com diferentes conjuntos de hiperparâmetros, no intuito de se definir os mais adequados a serem utilizados na plataforma TôSabendo.

1.4 Organização do Trabalho

Os seguintes capítulos deste trabalho encontram-se na seguinte ordem: o Capítulo 2 apresenta a revisão de literatura necessária para a realização deste trabalho, envolvendo fundamentação teórica e trabalhos diretamente relacionados. O Capítulo 3 apresenta a metodologia utilizada de forma detalhada, exibindo cada uma das etapas realizadas para implementação da estratégia de predição proposta para a plataforma TôSabendo com as devidas explicações sobre cada uma das decisões tomadas ao longo do processo. O Capitulo 4 discute os experimentos práticos realizados e os resultados obtidos. Por fim, o Capítulo 5, apresenta as conclusões deste trabalho e as perspectivas de trabalho futuro

2 Revisão Bibliográfica

Este capítulo apresenta a revisão de literatura com informações e referências importantes pesquisadas para realização deste trabalho, garantindo a confiabilidade e a qualidade técnica e científica do mesmo. Para tal, foi feita uma divisão em duas seções gerais: a Seção 2.1 apresenta a fundamentação teórica, que descreve conceitos, tecnologias e ferramentas relevantes para o desenvolvimento deste trabalho, e a Seção 2.2 expõe trabalhos e referências diretamente relacionados ao objetivo geral deste trabalho, auxiliando no embasamento científico e para definição da estrategia de predição proposta nesse trabalho

2.1 Fundamentação Teórica

Nesta seção, é apresentado o suporte teórico necessário para o entendimento e desenvolvimento deste trabalho. Para este fim, dividiu-se em quatro subseções: a Subseção 2.1.1 descreve a plataforma TôSabendo, cenário da aplicação da estratégia proposta neste trabalho, a Subseção 2.1.2 apresenta conceitos e uma exposição sobre alguns modelos preditivos que poderão vir a ser testados e validados, a Subseção 2.1.3 define o Desempenho Acadêmico a partir de diversos fatores específicos e sua importância para as universidades, e a Subseção 2.1.4 descreve o processo da EDM.

2.1.1 Plataforma TôSabendo

FILIPOUSKI, Marchi e SIMÕES (2009) consideram aspectos importantes na aplicação da gamificação em ambientes de aprendizagens: o planejamento, a possibilidade de trabalhar com experimentações, ciclos rápidos de *feedback*, diferentes níveis de complexidades, subdivisão de tarefas complexas em várias menores, sistema efetivo de recompensas, possibilidade de vivências de papéis, diversão e prazer. Por meio disso, a plataforma TôSabendo foi desenvolvida pensando na criação de diferentes tipos de Quizzes (jogos de perguntas e respostas), engajando tanto discentes quanto professores no possível e divertido jogo de ensinar e aprender (França *et al.*, 2021), e com um diferencial: o foco é na aprendizagem para disciplinas de cursos de IESs

No funcionamento inicial da TôSabendo, cada um dos Quizzes é apresentado na forma de um tabuleiro que representa um mapa (vide Figura 2.1), tendo um caminho a seguir. Cada uma das casas contém um nível diferente de dificuldade de questão, levando o jogador a evoluir nos conceitos do conteúdo abordado naquele *Quizz*. Ao percorrer todo o percurso desse "tabuleiro do conhecimento", o jogador é informado que "está sabendo" aquele determinado conteúdo. Esta é a razão da escolha do nome "TôSabendo": uma expressão descontraída que representa o sentimento que se espera que o jogador tenha ao final do *Quizz* que jogar.

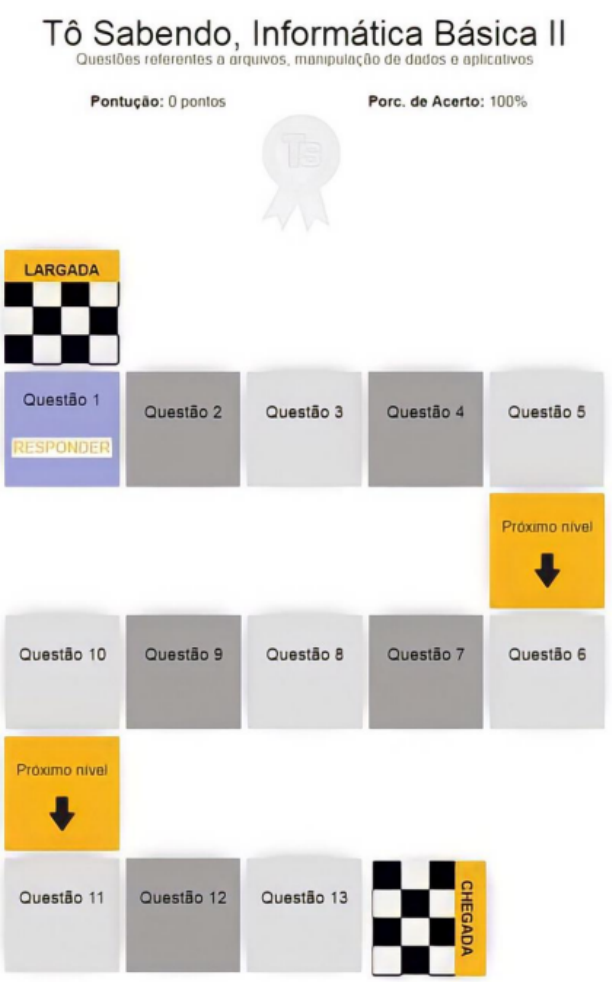

Figura 2.1 – Protótipo do tabuleiro (França *et al.*, 2021)

Com essa versão pronta, discentes do Curso de Ciência da Computação da Universidade Federal de Ouro Preto a utilizaram com o intuito de obterem bons resultados no Exame Nacional de Desempenho de Estudantes (ENADE). Para este fim, a metodologia empregada consistiu em comparar o desempenho dos discentes antes e após utilizarem a TôSabendo. Para isso, foram realizados dois simulados com questões retiradas de edições anteriores do ENADE, dos quais o segundo simulado foi aplicado após o acesso a TôSabendo. Com esse segundo simulado aplicado, foram obtidos resultados satisfatórios, o que mostrou que a plataforma proporcionou engajamento e aprendizagem efetiva dos discentes que a utilizaram, atendendo satisfatoriamente às exigências de qualidade de um método de ensino-aprendizagem. Segundo França *et al.* (2021), 82% dos discentes, que participaram desse experimento, consideram a plataforma de grande importância para a preparação para o ENADE e 73% dos discentes consideram a TôSabendo uma boa ferramenta educacional.

Portanto, atingiu-se o objetivo principal da plataforma de reunir e apresentar, de forma descontraída, conteúdos de disciplinas, auxiliando na aprendizagem de discentes quanto aos

conteúdos apresentados. Isso foi um indicativo para a continuidade do uso da plataforma, podendo contribuir na preparação de futuros ENADEs e também auxiliar professores no momento de lecionar suas disciplinas.

Entretanto, a plataforma tornou-se inutilizada por um tempo, visto que as tecnologias as quais ela foi criada tornaram-se obsoletas e, desta forma, algumas mudanças, implementadas por Ferreira (2022), foram definidas e encontram-se apresentadas na Subseção 2.1.1.1, envolvendo o *back-end* da plataforma, e na Subseção 2.1.1.2, envolvendo o banco de dados de suporte da plataforma.

2.1.1.1 *Back-end*

Segundo Ferreira (2022), para o desenvolvimento do *back-end*, usou-se como linguagem o *Javascript*, para gerar uma API REST. Basicamente, a interação entre APIs é a requisição e resposta entre um cliente um servidor. Uma API que segue os padrões REST para construir serviços *web* são chamadas de API REST. O REST é um padrão de arquitetura designado para aplicações em rede (Zhou *et al.*, 2014), utilizando o protocolo HTTP com os métodos *GET, POST, PUT, DELETE e PATCH* (vide Tabela 2.1) e apresentando as seguintes restrições de acordo com Fielding (2000):

- **Cliente-servidor:** utiliza-se uma arquitetura onde se separa o cliente e o servidor, tornando-os independentes;
- ***Stateless*:** a comunicação entre o cliente e servidor é *stateless*, ou seja, qualquer solicitação do cliente não deve ser salva. Além disso, nenhuma seção ou histórico deve ser criado de uma requisição do cliente;
- ***Cache*:** armazenar os dados em *cache* é muito importante, pois melhora o desempenho no lado do cliente, permitindo que ele reutilize a resposta retornada, e a escalabilidade avança no lado do servidor;
- **Interface Uniforme:** esse tipo de interface desacopla e simplifica a arquitetura, o que permite sua evolução de forma independente;
- **Sistema em camadas:** há um hierarquia sobre os componentes de software e eles não conseguem visualizar o que está acontecendo dentro de outro componente no qual não esteja conectado;
- **Código sob demanda:** o software envia código executável de um servidor para um cliente mediante solicitação do cliente. Essa restrição melhora a extensibilidade do sistema.

No caso de uma API REST, o *back-end* é separado do *front-end*, que seria toda interface que o usuário consegue manipular. O *back-end* fica responsável por manipular as informações do

Métodos	**Responsabilidade**
GET	Buscar dados
POST	Salvar os dados
PUT	Substitui determinado dado
DELETE	Apaga determinado dado
PATCH	Atualiza determinados dados

Tabela 2.1 – Métodos HTTP (Kimura, 2021)

banco de dados, receber requisições e enviar respostas. Para esse, já se implementou a autenticação do usuário, adequou-se as funções para cada tipo de usuário da plataforma e desenvolveu-se o CRUD das instâncias da plataforma (Ferreira, 2022), ou seja, funções de criação de dados para o banco, de leitura de dados, de atualização dos dados e de deleção dos dados.

2.1.1.2 Banco de dados

De acordo com Ferreira (2022), a Figura 2.2 apresenta o Esquema Entidade Relacionamento (EER) atual da plataforma TôSabendo, o qual foi gerado seguindo a metodologia para concepção do banco de dados. Como pode ser observado, existem três tipos específicos de usuários: o administrador, o colaborador e o jogador, por meio de uma especialização sobreposta, o que permite que qualquer um deles atue como outro, ou seja, um jogador pode atuar como colaborador e entre outras possibilidades. O administrador gerencia permissões dos demais usuários da plataforma. O colaborador cria e edita questões e Quizzes na plataforma, sendo passado essa função para um professor. E o jogador, que é o usuário-alvo, joga os Quizzes ao seu dispor, contendo informações do seu nível dentro da plataforma, sua pontuação geral e uma breve biografia. Todos os usuários específicos possuem o login, nome completo, sexo, ultimo acesso e a data de cadastro. Ademais, eles também possuem um curso, a instituição de ensino e qual a sua localidade.

Além desses três importantes usuários, há também as questões, as quais são formadas por enunciado, dicas, fonte, nível de dificuldade e quatro alternativas, com um detalhe de que o enunciado e as alternativas de uma questão podem ser do tipo textual, imagem ou vídeo. Elas estão organizadas por subcategorias, as quais estão organizadas por categorias. Esse esquema de categorias e subcategorias assemelha-se à forma de organização dos planos de ensino de disciplinas de IESs, facilitando sua adaptação para a TôSabendo. Enfim, cada questão possui um nível de dificuldade, podendo ser básico, médio ou avançado, que define a pontuação que o jogador receberá ao acertá-la e o tempo que terá para respondê-la.

Já em relação aos Quizzes, são formados, em suma, por questões e um título. É onde os

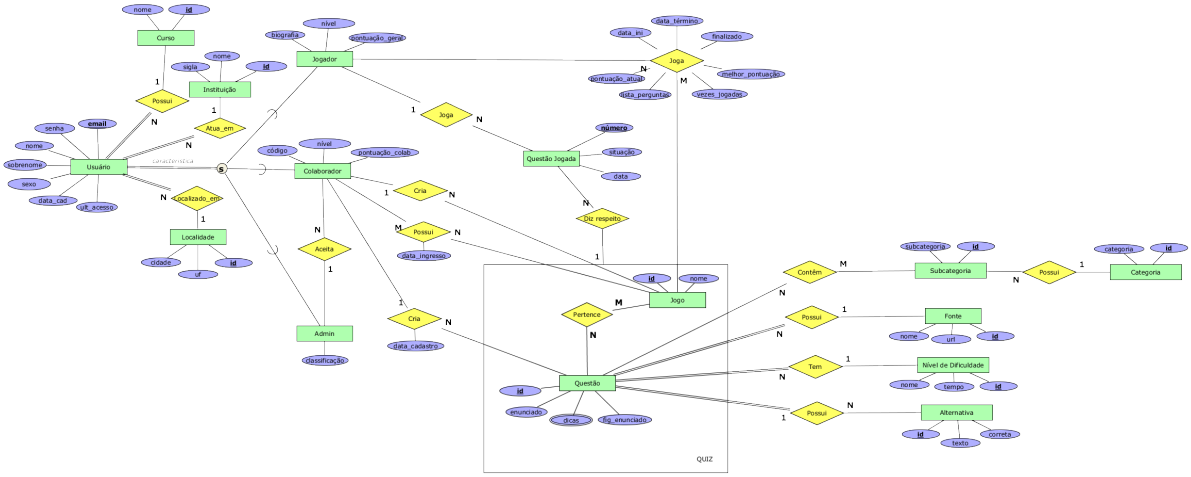

Figura 2.2 – Esquema Entidade-Relacionamento da atual versão da TôSabendo

colaboradores poderão criar novas questões para os jogadores. Para cada um desses Quizzes, um jogador possui uma pontuação e o nível em que se encontra.

Para melhor entendimento do banco dados confeccionado para a plataforma TôSabendo, o Anexo A apresenta o esquema relacional do mesmo gerado a partir do EER ilustrado na Figura 2.2.

2.1.2 Modelos de predição

Os modelos de predição são um ramo direto do Aprendizado de Máquina. O *Machine Learning*, ou a tradução para Aprendizado de Máquina, (Zhou, 2021) é um subcampo da inteligência artificial (IA) que vem sendo cada vez mais difundido em aplicações de todos os tipos a fim de aumentar a capacidade de lidar com a crescente quantidade de dados disponíveis e extrair percepções úteis e informações relevantes a partir desses dados. De forma resumida, como falado por Murphy (2012), Aprendizado de Máquina é um conjunto de algoritmos que podem detectar automaticamente padrões nos dados e, em seguida, usar os padrões descobertos para prever futuros dados, ou para realizar outros tipos de tomada de decisão sob incerteza.

De acordo com Samuel (1959), "Aprendizado de Máquina é o campo de estudo que dá aos computadores a capacidade de aprender sem ser explicitamente programado". Essa capacidade de aprendizado contínuo é uma das principais vantagens do Aprendizado de Máquina em relação aos algoritmos tradicionais de programação, que exigem atualizações manuais e intervenção humana para se adaptarem a novas situações ou padrões.

No âmbito acadêmico, o Aprendizado de Máquina pode ser aplicado para auxiliar os docentes a direcionar os discentes sobre os conceitos que precisam ser reforçados, os testando de diversas formas (Joyce; Harris, 2018); um dos ramos responsáveis por esse auxilio é o de modelos de predição.

Os modelos de predição são formados por algoritmos ou abordagens utilizadas para prever valores ou eventos futuros com base em dados históricos (Bramer; Bramer, 2016). Eles são amplamente aplicados em diversas áreas, como ciência de dados, finanças, medicina e educação, para tomar decisões informadas e antecipar resultados. Alguns exemplos de uso nessas áreas seriam:

- **Ciência de dados** (Xu *et al.*, 2022): identificar padrões que podem sugerir fraudes e, a partir disso, emitir alertas ou ativar mecanismos de proteção capazes de barrá-las em tempo hábil;
- **Finanças** (Ala'raj *et al.*, 2022): estimar o comportamento de pontuação de crédito dos clientes em bancos;
- **Medicina** (Javeed *et al.*, 2023): realizar a previsão de diagnósticos médicos para alguma doença específica, como a demência;
- **Educação** (Sekeroglu; Dimililer; Tuncal, 2019): predizer a performance de discentes; no caso, um modelo pode descobrir pontos fracos e sugerir maneiras de melhorá-los a partir da realização do acompanhamento do discente.

Particularmente na educação, como apresentado no artigo de Sekeroglu, Dimililer e Tuncal (2019), a partir do uso de três diferentes modelos de predição, observou-se que "o ambiente social e familiar do discente é de grande relevância para melhorar a qualidade da educação para futuras gerações". Assim, conclui-se que o uso de diferentes modelos de predição e com diferentes e mais extensas bases de dados gerariam melhores resultados e seriam vitais para uma intervenção precoce do discente, melhorando o desempenho do mesmo.

Segundo Bramer e Bramer (2016), esses modelos de predição podem ser divididos em duas vertentes: tarefa de classificação e a tarefa de regressão. A tarefa de classificação é usada quando se quer atribuir uma determinada classe ou categoria a uma instância de dados. O objetivo é gerar um modelo capaz de classificar corretamente as instâncias desconhecidas em classes pré-definidas; a saída desejada é discreta e representa uma classe ou categoria. Já na tarefa de regressão, ao contrário da classificação, a saída desejada é um valor numérico contínuo ou uma função de saída contínua. Os modelos de predição para regressão são projetados para aprender padrões e relacionamentos entre os atributos de entrada e a variável de destino contínua. Em resumo, a principal diferença entre tarefas de classificação e regressão está na natureza da saída desejada.

As Subseções 2.1.2.1 a 2.1.2.4 descrevem, respectivamente, o funcionamento dos modelos de predição com os algoritmos de classificação: Árvores de decisão, KNN, Naive Bayes e Redes Neurais Artificiais, visto que são os algoritmos de classificação que mais aparecem em modelos de predição na literatura de previsão de desempenho acadêmico de acordo com Alyahyan e Düştegör (2020).

2.1.2.1 Árvores de decisão

Uma árvore de decisão usa a estratégia dividir para conquistar de modo a resolver um problema de decisão (Faceli *et al.*, 2021). Um problema complexo é dividido em problemas mais simples, aos quais recursivamente é aplicada a mesma estratégia. As soluções dos subproblemas podem ser combinadas, na forma de uma árvore, para produzir uma solução do problema complexo. A força dessa proposta vem da capacidade de dividir o espaço de instâncias em subespaços e cada subespaço é ajustado usando diferentes modelos. Essa é a ideia básica por trás de algoritmos baseados em árvores de decisão, tais como: *Iterative Dichotomiser3* (ID3) (Quinlan, 1986), ASSISTANT (Cestnik; Kononenko; Bratko, 1987), *Classification And Regression Tree* (CART) (Breiman, 2017) e C4.5 (Quinlan *et al.*, 1996).

Uma árvore de decisão é uma estrutura de dados definida (Monard; Baranauskas, 2003) recursivamente como: um nó folha que corresponde a uma classe ou um nó de decisão que contém um teste condicional baseado nos valores do atributo. Na proposta padrão, os testes são uni-variados: as condições envolvem um único atributo e valores no domínio desse atributo. Para cada resultado do teste existe uma aresta para uma subárvore. Cada subárvore tem a mesma estrutura que a árvore.

Esses testes condicionais variam de acordo com o resultado final que deseja chegar, ou seja, qual classe a árvore está tentando induzir. Exemplos de testes condicionais para o prever o caso de um discente ser aprovado ou não em Estrutura de dados I, uma matéria ofertada em um curso de ciência da computação, seriam:

- Média de notas >= 75;
- Nota em Estrutura de dados I > 70;
- Nota em Introdução a Programação >= 80.

Outros exemplos de teste condicionais seriam para o caso de um diagnóstico de um paciente, por exemplo, como observa-se na Figura 2.3, onde cada elipse é um teste em um atributo para um dado conjunto de dados de pacientes e cada retângulo representa uma classe, ou seja, o diagnóstico. Para diagnosticar (classificar) um paciente, basta começar pela raíz, seguindo cada teste até que uma folha seja alcançada.

No intuito de mostrar um outro exemplo mais geral, a Figura 2.4 apresenta um esquema proposto por Faceli *et al.* (2021) de uma árvore de decisão e a divisão correspondente definidos pelos Atributos x1 e x2. Cada nó folha da árvore corresponde a uma região nesse espaço. As regiões definidas pelas folhas da árvore são mutuamente excludentes e a reunião dessas regiões cobre todo o espaço definido pelos atributos. A interseção das regiões abrangidas por quaisquer duas folhas é vazia. A união de todas as regiões (todas as folhas) é U. Uma árvore de decisão

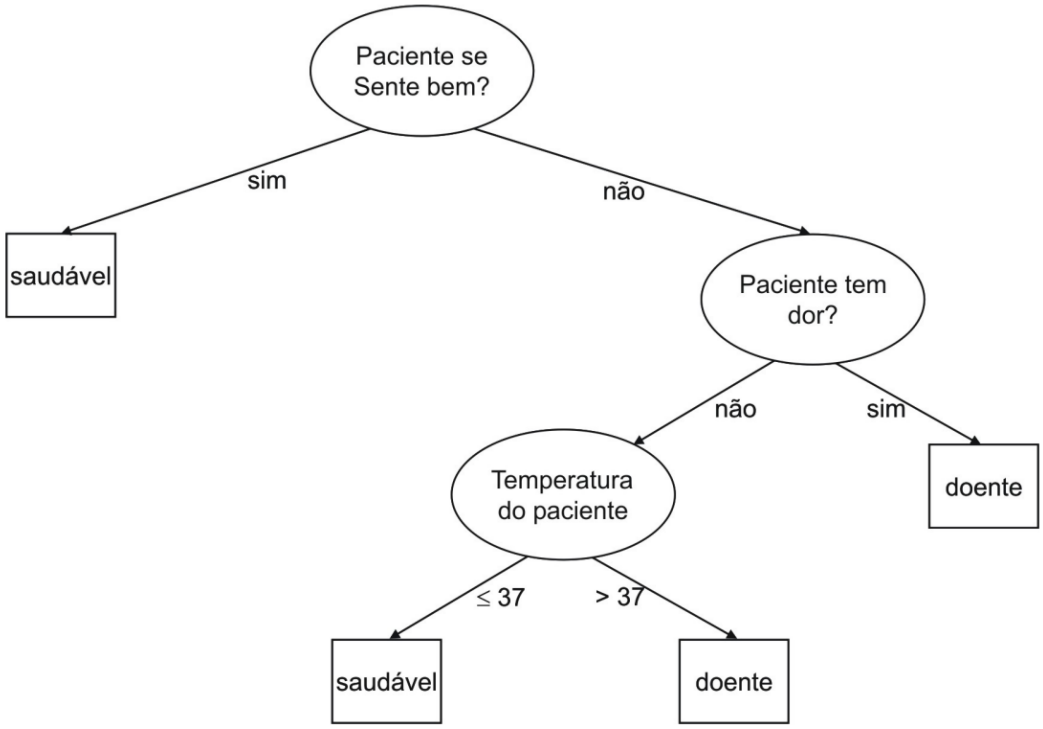

Figura 2.3 – Árvore de decisão simples para diagnóstico de um paciente (Monard; Baranauskas, 2003)

abrange todo o espaço de instâncias. Esse fato implica que uma árvore de decisão pode fazer predições para qualquer exemplo de entrada.

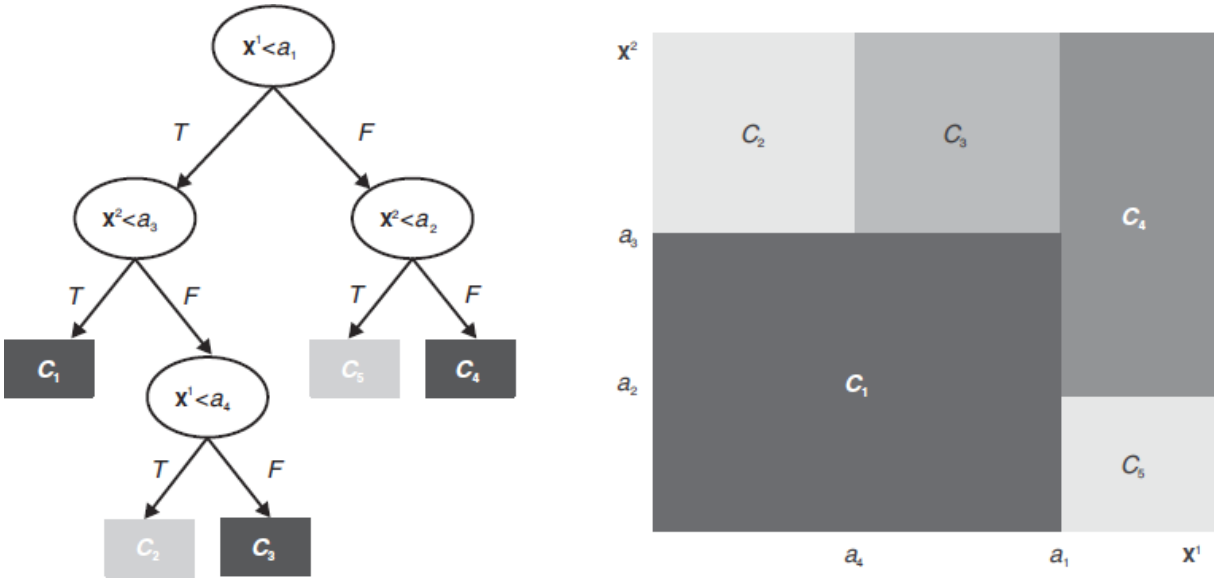

Figura 2.4 – Árvore de decisão e as regiões de decisão no espaço de busca (Faceli *et al.*, 2021)

.

De acordo com Gama (2002), as principais desvantagens das árvores de decisão são a sua instabilidade, em que pequenas perturbações do conjunto de treino podem provocar grandes

alterações no modelo aprendido, e valores ausentes (valor de um atributo é desconhecido) podem causar problemas em decidir que da árvore seguir. Apesar disso, elas apresentam uma alta interpretabilidade, em que uma decisão complexa (prever o valor da classe) é decomposta numa sucessão de decisões elementares, uma eficiência muito boa, com sua complexidade de tempo linear, e a seleção de atributos, que produz modelos que tendem a ser bastante robustos contra a adição de atributos irrelevantes e redundantes.

2.1.2.2 KNN

O algoritmo *k-Nearest Neighbour* (KNN) é considerado um dos algoritmos de classificação mais populares e simples utilizado em modelos de predição. A ideia básica por trás do KNN é que objetos semelhantes tendem a estar próximos uns dos outros (Faceli *et al.*, 2021). O algoritmo reconhece os vizinhos mais próximos da amostra (instância do conjunto de dados, como, por exemplo, um discente) no espaço de características, e a previsão para a nova amostra é então calculada como a média das respostas dos K vizinhos mais próximos (Kuhn; Johnson *et al.*, 2013). Segundo Raschka e Mirjalili (2017), esse algoritmo apresenta duas características cruciais que antecedem a previsão da resposta desejada para o novo conjunto de amostras: a determinação do número de vizinhos mais próximos K que formarão a vizinhança da nova amostra e a definição da métrica de distância que identificará as K amostras mais próximas na base de dados de treinamento em relação à nova amostra. Segundo Faceli *et al.* (2021), o estabelecimento do número de K vizinhos mais próximos é considerado um parâmetro que influencia a capacidade de generalização do modelo para dados futuros.

Quanto à escolha da medida de distância, ela é influenciada pela maneira como é definida a separação entre as amostras. A distância euclidiana, que é a distância direta entre duas amostras, é a métrica mais frequente nesse contexto (Kuhn; Johnson *et al.*, 2013). A mesma é apresentada pela Equação 2.1. Além disso, Raschka e Mirjalili (2017) ressalta que para aplicar essa medida, é crucial estabelecer um padrão consistente na base de dados durante a etapa de pré-processamento (vide Seção 2.1.4.4), de forma que cada característica/atributo presente contribua igualmente para o cálculo da distância.

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \tag{2.1}$$

onde:

- $d(\mathbf{x}, \mathbf{y})$ representa a distância euclidiana entre os vetores X e Y;
- n é a dimensão dos vetores (ou seja, o número de características);
- x_i e y_i são os elementos correspondentes nos vetores X e Y

Para a classificação, a resposta a ser predita deve ser representada pela classe mais comum observada na vizinhança de x^*. Ou seja, para cada classe da resposta de interesse *j*, calcula-se a probabilidade condicional da nova observação pertencer a *j-ésima* classe através da fração de pontos em $N_k(x)$ cujo valor da resposta é j, apresentada na Equação 2.2,

$$P(Y = j|X = x^*) = \frac{1}{K} \sum_{i \in N_k(x)} I(y_i = j) \tag{2.2}$$

onde a classe predita *j* para x^* será representada pela classe que evidenciar a maior probabilidade condicional (Hastie *et al.*, 2009; Santos, 2018).

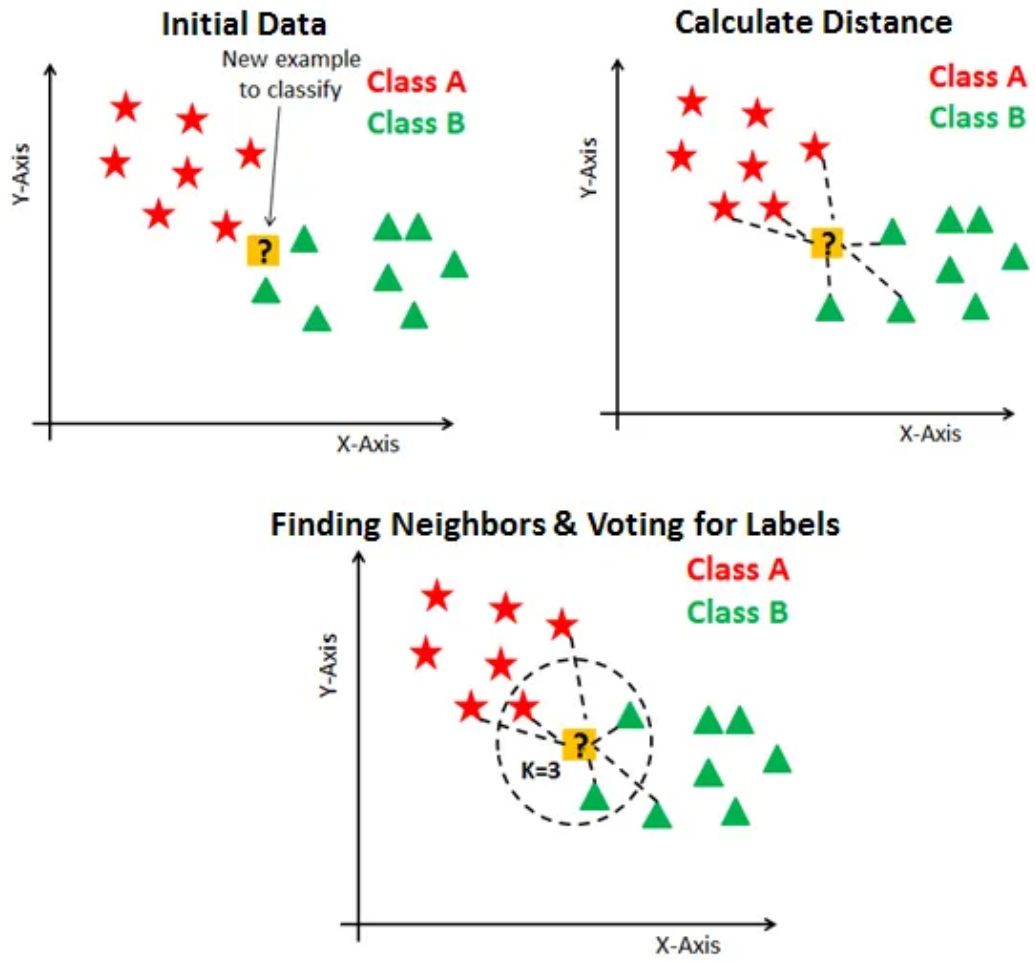

Figura 2.5 – Modelo gráfico 2D das ações básicas do modelo, sendo os eixos x e y características normalizadas presentes no conjunto de dados (Azank, 2019)

Uma demonstração geral de execução do algoritmo KNN é apresentada na Figura 2.5, iniciando pelo passo "*Initial Data*", que indica o algoritmo KNN treinado com suas respectivas classes A e B, com as características X e Y utilizadas para o treinamento do mesmo e a nova amostra a ser classificada. Em seguida, realiza-se o passo "*Calculate Distance*", onde se utiliza

uma métrica de distância como, por exemplo, a distância euclidiana, para calcular a distância entre a nova amostra a ser classificada e as amostras já classificadas. Por fim, no passo "*Finding Neighbors & Voting for Labels*" é calculada a probabilidade condicional, como apresentada na Equação 2.2 a partir dos K vizinhos mais próximos, que são definidos pela distância euclidiana calculada anteriormente. Com a probabilidade condicional calculada, identifica-se que a nova amostra pertence classe B, que apresenta mais amostras na proximidade dentre as K existentes.

De acordo com Cunningham e Delany (2021), o algoritmo KNN é relativamente simples de implementar e entender. No entanto, ele pode ser computacionalmente caro para conjuntos de dados grandes, pois precisa calcular a distância entre a nova instância e todas as outras instâncias de treinamento. Além disso, a escolha adequada de K e da medida de distância é fundamental para obter bons resultados com o KNN. Em alguns casos, pré-processamento de dados, normalização ou redução de dimensionalidade também podem ser necessários para melhorar o desempenho do algoritmo. Portanto, apesar de suas limitações, o KNN continua sendo um algoritmo popular devido à sua simplicidade e capacidade de lidar com uma variedade de problemas de aprendizado de máquina.

2.1.2.3 Naive Bayes

Os algoritmos de classificação bayesianos, baseados no teorema de Bayes, são classificadores estatísticos, os quais buscam prever a probabilidade de participação na classe, ou seja, a probabilidade de uma determinada amostra, pertencer a uma determinada classe, apresentando alta precisão e velocidade quando aplicados a grandes conjuntos de dados (Han; Pei; Tong, 2022).

Han, Pei e Tong (2022) e Lantz (2019) afirmam que há um classificador bayesiano simples conhecido como *naive bayes classifier*, ou classificador bayesiano "ingênuo", comparável em eficiência e performance com os algoritmos de classificação de árvores de decisão e redes neurais artificiais. O algoritmo de classificação naive bayes assume que o efeito de uma característica em uma determinada classe é independente dos valores das outras características. Essa suposição é chamada de *class conditional independence* (independência condicional de classe) e é imposta para simplificar as análises envolvidas e, por esse motivo, é considerado um classificador "ingênuo".

Inicialmente, o teorema de bayes é útil na medida em que fornece uma maneira de calcular a probabilidade condicional P(y/**X**) de P(y), P(**X**/y) e P(**X**). Assim, assumindo que **X** seja um vetor de p covariadas ou parâmetros e y a variável de classe, tem-se o teorema de bayes dado pela Equação 2.3.

$$P(y|X) = \frac{P(X|y) \cdot P(y)}{P(X)} \tag{2.3}$$

onde

- P(y/**X**) é probabilidade condicional de *y* dado **X**
- P(**X**/y) é a probabilidade condicional de **X** dado *y*
- P(*y*) probabilidade prior de *y*
- P(**X**) é a probabilidade marginal de **X**

Com esse teorema como base, de acordo com Han, Pei e Tong (2022), supondo que existam m classes, C1, C2, ..., Cm, o classificador naive bayes, ou classificador bayesiano simples, prevê que o rótulo de classe da tupla X é a classe Ci, seguindo a Equação 2.4.

$$P(X|C_i)\cdot P(C_i) > P(X|C_j)\cdot P(C_j) \quad \text{para } 1 \leq j \leq m,\ j \neq i, \text{ sendo i e j duas classes distintas} \tag{2.4}$$

Assim, o rótulo de classe previsto é a classe *Ci*, para a qual P(**X**|*Ci*) é o máximo.

De acordo com (Faceli *et al.*, 2021), apesar do algoritmo Naive Bayes apresentar uma boa eficácia com grande conjunto de dados, ele possui uma complexidade crescente, o que pode torná-lo lento. Além disso, ele assume todas as variáveis de entrada são independentes entre si, dada a classe. Essa suposição é frequentemente simplista demais para refletir as relações complexas entre as variáveis do mundo real. Em cenários onde as variáveis estão correlacionadas, essa suposição pode levar a resultados imprecisos. Outro problema, é a sua dificuldade de lidar com dados numéricos contínuos, em que quando isso acontece é necessário discretizar esses dados em intervalos, o que pode introduzir perda de informações e afetar a qualidade das previsões. Portanto, o Naive Bayes é uma técnica poderosa e rápida para classificação, mas suas simplificações podem levar a resultados sub-ótimos em algumas situações.

2.1.2.4 Redes Neurais Artificiais

As redes neurais artificiais (RNAs), ou *Artificial Neural Networks* (ANNs) são algoritmos de classificação que utilizam de modelos matemáticos inspirados na organização e no funcionamento dos neurônios no cérebro humano. Essas redes consistem em camadas interconectadas de unidades de processamento chamadas neurônios artificiais ou nós, que transmitem e processam informações por meio de conexões ponderadas (Krogh, 2008).

Dessa maneira, o neurônio é a unidade de processamento fundamental de uma RNA. Na Figura 2.6 é apresentado um modelo simplificado de neurônio artificial. As unidades de processamento desempenham um papel muito simples. Cada terminal de entrada do neurônio, simulando os dendritos[1], recebe um valor. Os valores recebidos são ponderados e combinados por

[1] Dendritos são extensões ramificadas e finas que se projetam a partir do corpo celular (ou soma) de um neurônio. Eles são uma parte essencial das células nervosas, também conhecidas como neurônios, que são os componentes básicos do sistema nervoso.

uma função matemática *f*. A saída da função é a resposta do neurônio para a entrada (Faceli *et al.*, 2021). Para cálculo da entrada total recebida pelo neurônio, primeiro supõe-se um objeto **x** com *d* atributos representado na forma de vetor como *x* = [x_1, x_2, ..., x_d] e um neurônio com d terminais de entrada cujos pesos são w_1, w_2, ..., w_d, que podem ser representados na forma vetorial como w = [w_1, w_2, ..., w_w]. Por fim, há o *b*, que indica o viés (bias) do neurônio. A entrada total recebida pelo neurônio, u, pode ser calculada pela Equação 2.5. Kubat (1999) ressalta que os neurônios podem apresentar conexões de entrada negativas (w_j < 0) ou positivas (w_j > 0). Um valor de peso igual a zero equivale à ausência da conexão associada.

$$u = \sum_{j=1}^{d} x_j \cdot w_j + b \tag{2.5}$$

A saída de um neurônio é definida por meio da aplicação de uma função de ativação à entrada total. Diversas funções podem ser usadas para cálculo da saída, como, por exemplo, linear (Figura 2.7(a)), limiar (Figura 2.7(b)), sigmoidal (Figura 2.7(c)), tangente hiperbólica (Figura 2.7(d)), gaussiana (Figura 2.7(e)) e linear retificada (Figura 2.7(f)).

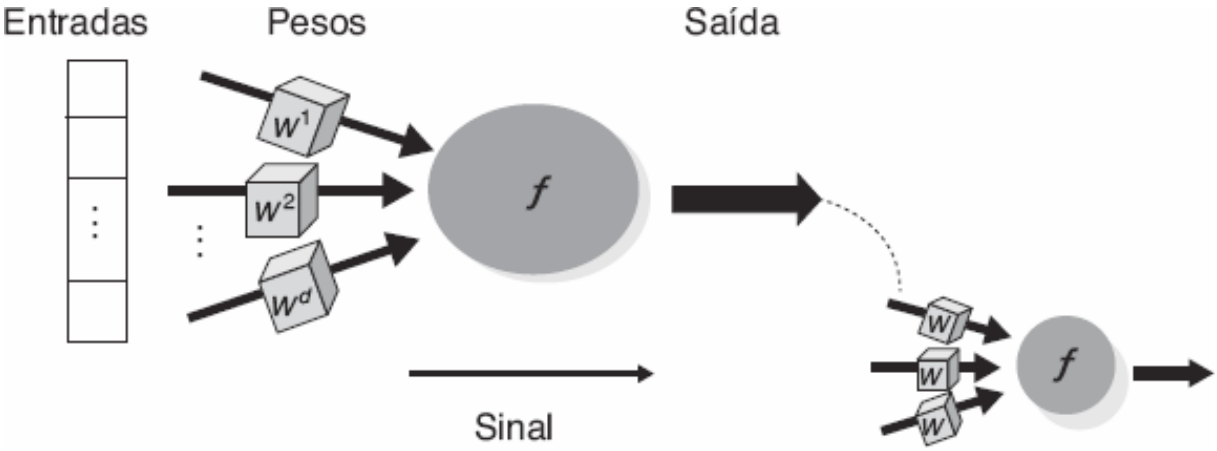

Figura 2.6 – Neurônio artificial (Kubat, 1999)

.

Entretanto, considerando um conjunto de neurônios e não um apenas, segundo Faceli *et al.* (2021), em uma RNA, os neurônios podem estar dispostos em uma ou mais camadas. Quando duas ou mais camadas são utilizadas, um neurônio pode receber em seus terminais de entrada valores de saída de neurônios da camada anterior e/ou enviar seu valor de saída para terminais de entrada de neurônios da camada seguinte. A Figura 2.8 ilustra um exemplo de RNA com três camadas. Essa rede recebe como entrada valores de dois atributos de entrada e gera dois valores em sua saída. Uma rede com mais de uma camada de neurônios recebe o nome de rede multicamadas. A camada de neurônios que gera os valores de saída é chamada de camada de saída. As demais camadas são denominadas camadas intermediárias.

Após o entendimento da arquitetura das RNAs, ainda há a necessidade de entender o funcionamento de seu treinamento. De uma forma simples, de acordo com Kubat (1999), o

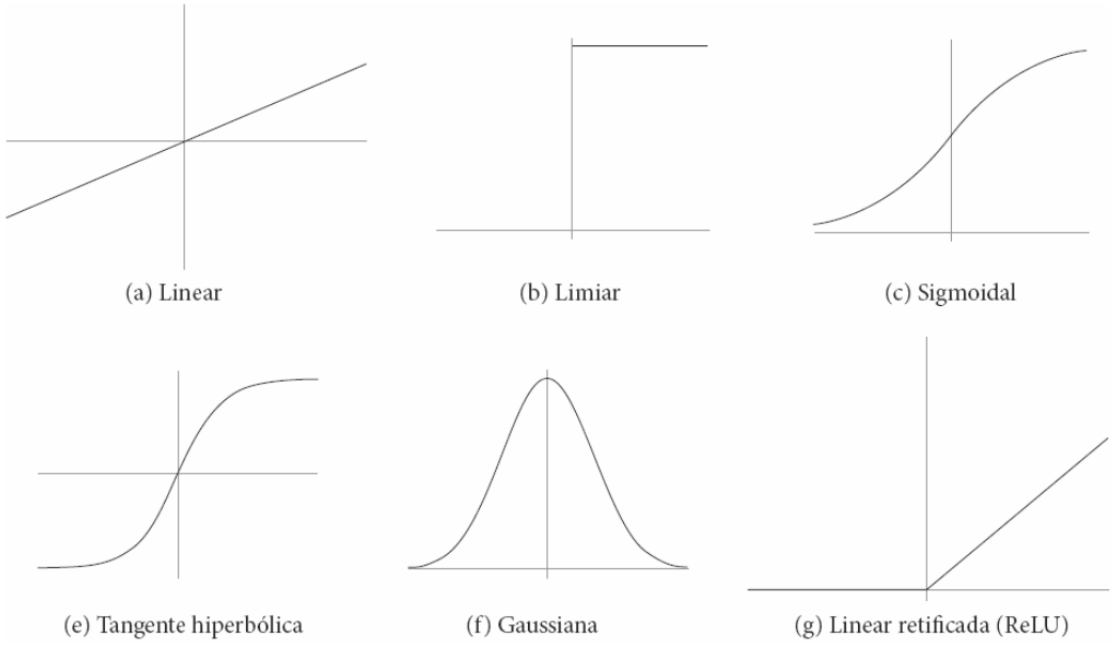

Figura 2.7 – Exemplos de funções de ativação (Faceli *et al.*, 2021)

.

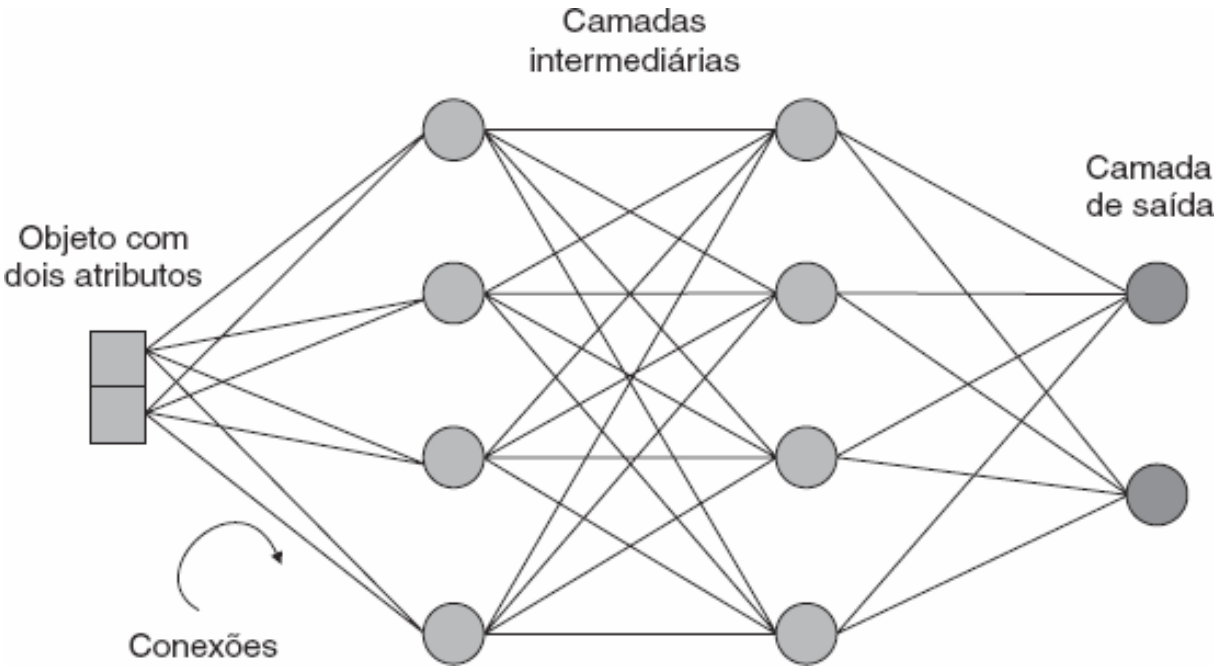

Figura 2.8 – Exemplo de RNA multicamadas típica. (Faceli *et al.*, 2021)

treinamento de RNAs é um processo que ajusta os pesos das conexões de modo a minimizar a diferença entre as saídas previstas e as saídas reais. O algoritmo de *backpropagation*, também conhecido como retropropagação de erro, é amplamente utilizado nesse processo. Ele calcula gradientes de erro em relação aos pesos da rede e os utiliza para atualizar os pesos de maneira iterativa, melhorando o desempenho da RNA ao longo do tempo.

De acordo com Faceli *et al.* (2021), as RNAs são algoritmos de classificação com uma

eficácia muito acima da média devido à sua capacidade de aprender padrões complexos a partir dos dados. Entretanto, geralmente requerem grandes conjuntos de dados para aprender padrões significativos. Com conjuntos de dados pequenos, as RNAs podem sofrer de *overfitting*, onde elas se ajustam demais aos dados de treinamento e têm dificuldade em generalizar para novos dados. Além disso, podem levar muito tempo para treinar, especialmente quando se trata de grandes conjuntos de dados. Isso pode ser um desafio em problemas que exigem resultados rápidos ou onde a iteração frequente é necessária. Por fim, um outro problema é que elas possuem diversos hiperparâmetros que precisam ser ajustados como, por exemplo, o número de camadas, o número de neurônios em cada camada ou taxas de aprendizado. Escolher os hiperparâmetros corretos pode ser um processo complexo e requer experimentação. Portanto, as RNAs são poderosos algoritmos de classificação, mas elas também têm algumas desvantagens importantes. A escolha de se usar RNAs deve considerar a natureza dos dados, os recursos computacionais disponíveis e as características do problema.

2.1.3 Desempenho Acadêmico

O desempenho dos discentes é uma componente crucial das instituições de ensino superior porque é considerado um critério essencial para avaliar a qualidade das instituições de ensino (Polidori; Marinho-Araujo; Barreyro, 2006). É um tema amplamente pesquisado nas áreas da psicologia educacional, sociologia da educação e outras disciplinas relacionadas (Romero; Ventura, 2007). De acordo com York, Gibson e Rankin (2015), a definição de desempenho do discente é resumida em seis componentes mais importantes: “realização acadêmica, satisfação, aquisição de habilidades e competências, persistência , alcance dos objetivos de aprendizagem e sucesso na carreira”.

Dessa maneira, a previsão antecipada do desempenho dos discentes pode ajudar os tomadores de decisão a fornecer as ações necessárias no momento certo e a planejar o treinamento adequado para melhorar a taxa de sucesso do discente (Dutt; Ismail; Herawan, 2017). Vários estudos foram publicados usando métodos de predição para prever o desempenho acadêmico dos discentes. Pode-se observar diferentes níveis de cada estudo como (Alyahyan; Düştegör, 2020):

- nível do diploma: tem como objetivo prever o desempenho dos discentes para obtenção do diploma;
- nível do ano: tem como objetivo prever o desempenho dos discentes até o final de um ano;
- nível do curso: tem como objetivo prever o desempenho dos discentes em um curso específico;
- nível do exame: tem como objetivo prever o desempenho dos discentes em um exame para um curso específico.

A TôSabendo (Subseção 2.1.1) encaixa-se no desempenho do discente no nível de um curso específico, no nível de apenas uma matéria específica ou até mesmo no nível de um exame, já que apresenta questões apresentadas durante todo o curso ou apenas de alguma matéria oferecida.

De acordo com Alyahyan e Düştegör (2020), a decisão mais importante para construção de um modelo preditivo para o desempenho acadêmico é definir o que seria o desempenho acadêmico. No caso da TôSabendo (Subseção 2.1.1), está relacionado a obter as melhores pontuações sobre os Quizzes, indicando que aquele discente está apto para aquela matéria ou para alguma prova específica, como o citado ENADE. Desta forma, é necessário definir os fatores mais influentes, ou seja, que ditam os dados que precisam ser coletados, minerados e usados como parâmetro.

Os fatores que podem impactar na previsão do desempenho dos discentes, apresentadas na Figura 2.9 são desempenho acadêmico prévio (antes da faculdade e na universidade), demografia do discente (sexo, idade, cor, status socioeconômico), atividade de *e-learning* (*logs* da plataforma, como, por exemplo, o Moodle, com quantidade de acessos, quantidade de tarefas realizadas, etc.), atributos psicológicos (ansiedade e estresse, interesse do discente nas aulas, tempo de preocupação com si mesmo), e ambientes (tipo de aula, duração do semestre) (Dutt; Ismail; Herawan, 2017).

Entretanto, os dois principais são desempenho acadêmico prévio e os dados demográficos do discente, que foram apresentados em 69% dos trabalhos de pesquisa (Alyahyan; Düştegör, 2020). Isso está relacionado com o fato de que as notas de avaliação e média cumulativa de notas são os fatores mais comuns utilizados para prever o desempenho do discente em EDM (Subseção 2.1.4). Sendo assim, com mais de 40%, o desempenho acadêmico anterior é o fator mais importante, que é basicamente toda "bagagem" ao qual o discente carrega, desde dados pré-universitários (ensino médio) (Rawal; Lal, 2023) até universitários (Ali *et al.*, 2020). Sobre dados pré-universitários, podem ser utilizados diferentes tópicos, como notas do ensino médio, os interesses em diferentes matérias e resultados do teste de admissão da faculdade. Para dados universitários, pode-se usar as notas dos discentes desde de quando entraram, incluindo a média cumulativa de notas ou média no semestre, marcos dentro do curso e notas de avaliação do curso e da faculdade. Também está nesse meio, Quizzes (como a TôSabendo), trabalhos de pesquisa e frequência.

Em relação aos dados demográficos, é um tema divergente na literatura (Alyahyan; Düştegör, 2020), em que vários estudos indicaram um impacto no desempenho dos discentes, como o gênero (Baashar *et al.*, 2022; Mittleman, 2022), idade (Wang *et al.*, 2022), cor (Gottfried; Kirksey; Fletcher, 2022) e status socioeconômico (Hamoud; Hashim; Awadh, 2018), porém uma maior oposição quanto ao gênero em particular (Garg, 2018).

Figura 2.9 – Conjunto variado de fatores que potencialmente impactam a predição do desempenho acadêmico de discentes (Dutt; Ismail; Herawan, 2017)

.

2.1.4 *Educational Data Mining* (EDM)

A mineração de dados educacionais é um campo de pesquisa emergente independente, preocupado com o desenvolvimento de métodos para explorar os dados únicos, cada vez mais em larga escala que vêm de ambientes educacionais e usar esses métodos para entender melhor os discentes e os ambientes em que eles aprendem (Liñán; Pérez, 2015; Dutt; Ismail; Herawan, 2017). Alguns dos exemplos os quais se observa o maior uso da EDM é em modelos preditivos para prever uma diversidade de resultados educacionais cruciais, como desempenho (Qiu *et al.*, 2022), retenção (Brdesee *et al.*, 2022), sucesso acadêmico (Alturki; Hulpuș; Stuckenschmidt, 2022), satisfação (Alqurashi, 2019), conquistas (Pallathadka *et al.*, 2023) e taxa de abandono (Niyogisubizo *et al.*, 2022); todos seguem um mesmo processo. Apesar de muitas publicações, incluindo estudos de caso, sobre mineração de dados educacionais, ainda é difícil para os educadores, principalmente se eles são iniciantes no campo de mineração de dados, aplicar efetivamente essas técnicas a seus problemas acadêmicos específicos (Alyahyan; Düştegör, 2020). Por isso, necessita-se de um aprendizado completo sobre todo o processo de EDM.

O processo de EDM é um processo iterativo de descoberta de conhecimento que consiste na formulação, teste e refinamento de hipóteses (Moscoso-Zea; Luján-Mora *et al.*, 2016; Sarala; Krishnaiah, 2015). Atualmente, a EDM desempenha um papel significativo na descoberta de padrões de conhecimento sobre os fenômenos educacionais e o processo de aprendizagem (M;

Rahman, 2016), incluindo a compreensão do desempenho dos discentes (Hasib *et al.*, 2022). Como se pode visualizar na Figura 2.10, de forma breve, o processo começa integrando dados brutos de diferentes fontes de dados, a partir de um ambiente educacional. Ocorre um pré-processamento, onde esses dados são limpos, removendo seus ruídos, dados duplicados ou inconsistentes. Esses dados atualizados são transformados em um formato conciso que pode ser entendido por ferramentas de mineração de dados, por meio de técnicas de filtragem e agregação. Assim, os modelos de mineração de dados são treinados e testados, para em seguida, na etapa de análise, identificar os padrões de interesse, que podem ser encontrados para uma melhor entendimento daquele ambiente escolar (Han; Pei; Tong, 2022) e, assim, auxiliar nas decisões dos professores e da própria faculdade. Segundo Romero e Ventura (2007), é importante ressaltar que cada uma das etapas citadas anteriormente requerem várias decisões e configuração de parâmetros, que afetam diretamente a qualidade do resultado obtido.

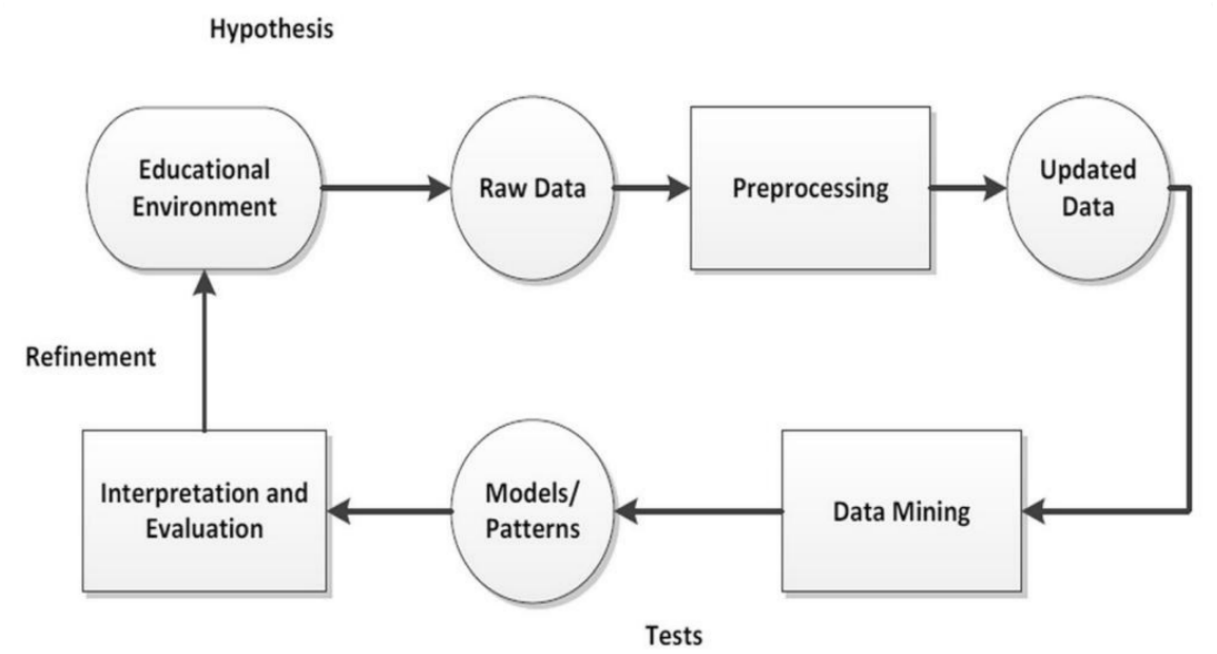

Figura 2.10 – Processo de descoberta de conhecimento em instituições educacionais (Alyahyan; Düştegör, 2020)

.

Portanto, por meio desse conhecimento obtido dos dados, aplicam-se técnicas de predição com maior clareza e menor risco de erro, visto que os dados passam a ficar mais homogêneos e concisos com a necessidade (Alyahyan; Düştegör, 2020), como, por exemplo, quando é necessário prever o desempenho acadêmico de um discente de graduação. Nesse caso, uma nota que é muito abaixo da média de um discente específico em um conjunto grande de dados pode afetar negativamente o desempenho do modelo de mineração de dados. Dessa maneira, há várias formas de evitar isso, como trocar aquele valor por uma média de todas as notas de outros discentes, colocar um valor aleatório de dentro de um intervalo ou até mesmo retirar aquele discente do conjunto de treinamento.

O restante desta Subseção descreve, de maneira geral as etapas da EDM apresentadas na Figura 2.11: Coleta de dados, Preparação inicial dos dados, Análise Estatística, Pré-processamento dos dados, Mineração de dados e, por fim, Análise dos dados, presentes, respectivamente, nas Subseções 2.1.4.1 a 2.1.4.6

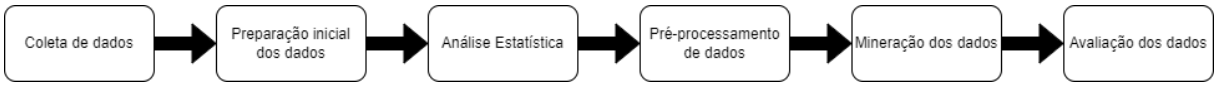

Figura 2.11 – Etapas da EDM (Alyahyan; Düştegör, 2020)

.

2.1.4.1 Coleta de dados

De acordo com Alyahyan e Düştegör (2020) os dados referentes aos discentes podem ser extraídos de várias fontes. Como se comentou na Subseção 2.1.3, o fator mais relevante observado na literatura é o desempenho acadêmico prévio do discente. Pode-se encontrar esses dados na base de dados das faculdades ou usar o Sistema de Informação do Estudante (SIS) universitário, que é muito utilizado para este tipo de pesquisa. O segundo fator mais relevante, que é a demografia do discente, como idade, sexo, cor, também pode ser encontrado nessas duas fontes; entretanto, dados como status socioeconômico podem não ser encontrados explicitamente e, nesse caso, recorre-se a outro meio, como pesquisas com os próprios discentes ou a dedução a partir de dados já existentes. Enquanto que informações relacionadas ao ambiente dos discentes podem ser extraídas do SIS, dados psicológicos provavelmente exigiriam que o discente preenchesse uma pesquisa. Por fim, as atividades de *e-learning* dos discentes podem ser obtidas a partir dos recolhimentos dos *logs* do sistema de *e-learning*, onde os discentes realizam atividades acadêmicas *online*, como envio de trabalhos, exercícios, provas ou simulados

2.1.4.2 Preparação inicial dos dados

Em sua forma original, os dados, ou também chamados de dados cru (*raw data*) usualmente não estão prontos para análise e modelagem (Alyahyan; Düştegör, 2020). Os conjuntos de dados que são obtidos principalmente da fusão de tabelas de diferentes fontes podem conter dados ausentes, dados inconsistentes, dados incorretos, dados codificados incorretamente e dados duplicados (Romero; Ventura, 2013). É por isso que os dados brutos precisam passar por uma preparação inicial e, como aponta (CrowdFlower, 2016), essa preparação inicial é a etapa mais demorada da EDM.

Essa preparação inicial consiste em três etapas: seleção de dados, limpeza de dados e derivação de novas variáveis, as quais estão apresentadas na Figura 2.12 . As próximas Subseções 2.1.4.2.1, 2.1.4.2.2 e 2.1.4.2.3 tratam, respectivamente, dessas três etapas.

Figura 2.12 – Preparação inicial dos dados (Alyahyan; Düştegör, 2020)

.

2.1.4.2.1 Seleção de dados

Nem todos os dados que foram coletados na primeira etapa (Subseção 2.1.4.1) são importantes para o objetivo desejado. Por isso, é sempre importante a partir de uma visualização sobre os dados e o seu objetivo, retirar ou adicionar mais dados para melhorar o modelo de mineração de dados. De acordo com Alyahyan e Düştegör (2020), isso é necessário, porque a dimensão dos dados coletados pode ser significativa, principalmente ao usar realizações acadêmicas anteriores, por exemplo quando se têm dados até mesmo do ensino médio daquele discente, como também dos anos de graduação concluídos. Isso pode impactar negativamente a complexidade computacional (Seifert, 2004). Ademais, incluir todos os dados coletados na análise pode gerar resultados de previsão abaixo do ideal, especialmente no caso de redundância de dados ou dependência de dados (Martins *et al.*, 2019). Dessa maneira, é crucial determinar quais atributos são importantes ou precisam ser incluídos na análise. Isso requer um bom entendimento dos objetivos da EDM, bem como do próprio domínio dos dados (Pyle, 1999)

Além disso, segundo Visalakshi e Radha (2014), com a seleção de dados, o modelo de mineração de dados fica mais fácil de ser entendido até mesmo visualmente com um número reduzido de recursos. Com isso, entende-se melhor o significado de cada atributo, instância e variável, testando-os por menos tempo, beneficiando o treinamento e facilitando a avaliação do modelo de mineração de dados (Alyahyan; Düştegör, 2020). Segundo García, Luengo e Herrera (2015), Nisbet, Elder e Miner (2009), Liñán e Pérez (2015), existem dois tipos de seleção de dados, que são a seleção vertical (atributos/variáveis) e a seleção horizontal (instância/registros).

2.1.4.2.2 Limpeza de dados

Quando se extrai os dados dessas fontes citadas na Subseção 2.1.4.1, muitas vezes eles podem vir inconsistentes, possuindo ruídos (como um valor muito grande ou muito pequeno) ou até mesmo com valores ausentes (Berry; Linoff, 2004). Quando um valor está em uma distância anormal dos outros valores no conjunto de dados, ele é chamado de *outlier*, ou também de anomalia. Valores ausentes e valores *outliers* são muito comuns no campo da EDM (Dutt; Ismail; Herawan, 2017). Por isso é muito importante saber como lidar com eles, tratando-os sem

comprometer a qualidade da previsão do modelo de mineração de dados (McCarthy *et al.*, 2022). É necessário considerar vários métodos de acordo com o contexto do problema a ser solucionado.

Quanto aos dados *outliers*, eles podem ser facilmente identificados por meios visuais, criando um histograma, diagramas ou gráficos e procurando por valores muito altos ou muito baixos (Alyahyan; Düştegör, 2020). Uma vez identificados, os *outliers* podem ser removidos dos dados de modelagem. Outra possibilidade é converter a variável numérica em uma variável categórica (ou seja, agrupar os dados) ou até mesmo deixar os valores discrepantes nos dados (McCarthy *et al.*, 2022).

Já em relação aos dados ausentes, existem duas estratégias para lidar com eles. A primeira (Brown; Kros, 2003) é uma exclusão *listwise*, que consiste em excluir o registro (exclusão de linha, quando os valores ausentes são poucos) ou o atributo/variável (exclusão de coluna, quando os valores ausentes são muitos). A segunda estratégia (Nisbet; Elder; Miner, 2009), imputação, deriva o valor ausente do restante dos dados, como, por exemplo, mediana, média, um valor constante para valor numérico, um valor selecionado aleatoriamente da distribuição de valores ausentes (Alyahyan; Düştegör, 2020) ou até mesmo transformá-lo em uma variável categórica.

2.1.4.2.3 Derivação de novas variáveis

Novas variáveis podem ser derivadas de variáveis existentes combinando-as (Nisbet; Elder; Miner, 2009). De acordo com Alyahyan e Düştegör (2020), um exemplo disso seria a média de todas as notas de um discente durante o semestre ou o GPA (*Grade Point Average*), que seria obtido no banco de dados da faculdade. Nesse caso, essa média só diz respeito à nota do discente em um semestre e não diz a respeito sobre como ele está ao longo de vários. Nesse caso, não é possível saber se ele está indo muito bem, passando por dificuldades ou estável. Logo, calcular a diferença da média entre dois semestres consecutivos pode adicionar uma informação extra para o que se deseja.

Portanto, esse é um processo que pode vir a melhorar o modelo de mineração de dados futuramente quando feito com uma boa base no conhecimento do domínio de dados (Feelders; Daniels; Holsheimer, 2000).

2.1.4.3 Análise Estatística

O processo de análise estatística em EDM envolve o uso de técnicas estatísticas para explorar, resumir e interpretar os dados (Chatfield, 1995). Essa análise estatística ajuda a identificar padrões, tendências e relações nos dados, fornecendo *insights* valiosos para a mineração de dados antes de passar para tarefas mais complicadas. De acordo com Alyahyan e Düştegör (2020), existem diversas estatísticas a serem aplicadas sobre os dados e que dependem do tipo de dado, como, para dados categóricos, a frequência e a moda, e, para dados contínuos, a média, mediana, desvio padrão, variância, intervalo, Kurtosis e correlação P.

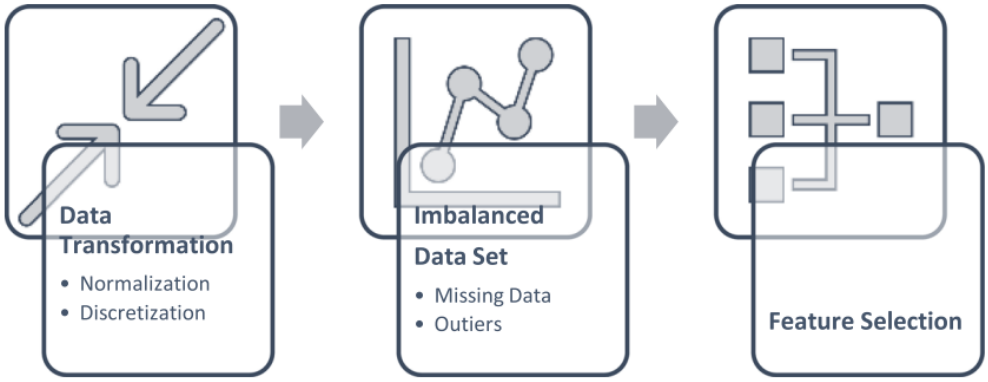

Figura 2.13 – Pré-processamento dos dados (Alyahyan; Düştegör, 2020)

.

Essa etapa pode ajudar também nas próximas etapas do processo de mineração, como no pré-processamento de dados para identificação de *outliers*, observar padrões de dados ausentes, estudar a distribuição de cada variável e identificar a relação entre variáveis independentes e a variável de destino (Chatfield, 1995). Além disso tudo, a análise estatística pode ser utilizada na fase de interpretação para explicar os resultados do modelo de mineração de dados (Pyle, 1999).

2.1.4.4 Pré-processamento de dados

Esta é a última etapa antes de passar para a implementação do modelo de mineração de dados. Ela é dividida em 3 etapas: transformação de dados, gerenciamento de conjuntos de dados desequilibrados e seleção de recursos. Essas 3 etapas estão apresentadas na Figura 2.13 e são explicadas a seguir.

A transformação de dados é um processo necessário para eliminar dissimilaridades no conjunto de dados tornando-se mais apropriado para a mineração de dados (Osborne, 2002). Nela, podem ocorrer as seguintes operações: normalização de atributos numéricos, discretização, conversão em variáveis numéricas e combinação de níveis. Apesar de existirem todos esses métodos, eles não levam necessariamente a melhores resultados. É necessário testá-los de diferentes formas e combinações, avaliando o desempenho do modelo de mineração de dados e identificando os melhores resultados (Alyahyan; Düştegör, 2020).

Os conjuntos de dados desequilibrados referem-se a um conjunto de dados em que as classes-alvo ou categorias de interesse não estão representadas igualmente (Qazi; Raza, 2012), ou seja, existe uma disparidade significativa no número de registros entre as diferentes classes do conjunto de dados, como, por exemplo, o número de discentes reprovados versus discentes aprovados. De acordo com Khoshgoftaar, Golawala e Hulse (2007), essa falta de equilíbrio pode impactar negativamente o desempenho dos algoritmos de classificação. A reamostragem (sub ou superamostragem) é a solução para esse problema (Chawla *et al.*, 2002), aumentando ou reduzindo instâncias dos dados de forma aleatória ou com técnicas para balanceamento

Por fim, tem-se a seleção de recurso, que é a etapa feita depois de todos os dados estarem preparados e prontos para a modelagem. É um processo de identificar e escolher um subconjunto relevante de recursos (variáveis) de um conjunto de dados (Liu; Motoda, 2012). Essa etapa é essencial na EDM, pois pode melhorar a precisão do modelo de mineração de dados, reduzir a dimensionalidade dos dados, eliminar ruídos ou redundâncias desnecessárias e reduzir o tempo de computação (Chandrashekar; Sahin, 2014). Segundo Kohavi e John (1997), os métodos de seleção de recursos são divididos em filtro e *wrapper*. Os métodos de filtro funcionam como um pré-processamento para classificar os recursos. Dessa maneira, os recursos com alto grau de classificação são identificados e aplicados ao preditor. Nos métodos *wrapper*, o critério para selecionar o recurso é o desempenho do modelo de predição, o que significa que o preditor é agrupado em um algoritmo de pesquisa que encontrará um subconjunto que forneça o desempenho mais alto.

2.1.4.5 Mineração dos dados

De forma simples, a etapa da mineração de dados visa escolher e utilizar um modelo, seja ele de predição ou descritivo, treinando e o testando com os dados tratados anteriormente, com o objetivo de observar e interpretar os seus resultados, retirando informações valiosas (Alyahyan; Düştegör, 2020). No caso da EDM, as informações valiosas seriam, por exemplo, se o discente irá bem em uma matéria, se há chances dele sair da faculdade, como está performance em uma plataforma de ensino *online*. Os modelos de predição encontram-se apresentados e explicados na Subseção 2.1.2. Já modelos descritivos são usados para produzir padrões que descrevem a estrutura fundamental, as relações e a interconectividade dos dados minerados (Peng *et al.*, 2008).

Conhecendo seus parâmetros, existem diversas maneiras de encontrar a melhor forma de usar esses modelos para obter as melhores informações; porém, a mais simples e fácil é a abordagem de tentativa e erro (Ruano *et al.*, 2010), que consiste em realizar numerosos experimentos modificando os valores dos parâmetros até encontrar os parâmetros de desempenho mais benéficos.

2.1.4.6 Avaliação dos dados

A avaliação dos modelos de classificação ocorre por meio de diferentes métricas, cada uma com diferentes significados. De forma geral, elas se baseiam em identificar o que o modelo apontou como verdadeiro o que é realmente verdadeiro, o que o modelo apontou como falso o que é realmente falso, e o que ele errou. A partir disso, é possível montar uma matriz, chamada de matriz de confusão, que sumariza o comportamento do modelo, representada pela Figura 2.14:

- *True positive* (TP): ocorre quando o modelo prediz o valor positivo e o valor correto é positivo;

		Valor Predito	
		Sim	**Não**
Real	**Sim**	Verdadeiro Positivo (TP)	Falso Negativo (FN)
	Não	Falso Positivo (FP)	Verdadeiro Negativo (TN)

Figura 2.14 – Possíveis valores em uma matriz de confusão (Diego Nogare, 2020)

.

- *True negative* (TN): ocorre quando o modelo prediz o valor negativo e o valor correto também é negativo;
- *False positive* (FP): ocorre quando o modelo prediz o valor positivo mas o valor correto é negativo;
- *False negative* (FN): ocorre quando o modelo prediz o valor negativo mas o valor correto é positivo.

A partir dessa contagem de erros e acertos, é possível estabelecer razões numéricas que definem a eficácia do modelo, como acurácia, precisão, revocação, F1-measure e ROC curve, apresentados a seguir:

- **Acurácia** (Equação 2.6): mede a proporção de acertos em relação a todas as predições realizadas;

$$\text{Acurácia} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{2.6}$$

- **Precisão** (Equação 2.7): mede, de todos os que foram detectados como positivos, quantos destes foram corretamente preditos;

$$\text{Precisão} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{2.7}$$

- **Revocação** (*Recall*) (Equação 2.8): mede a proporção de acertos em relação a todos os que deveriam ser positivos:

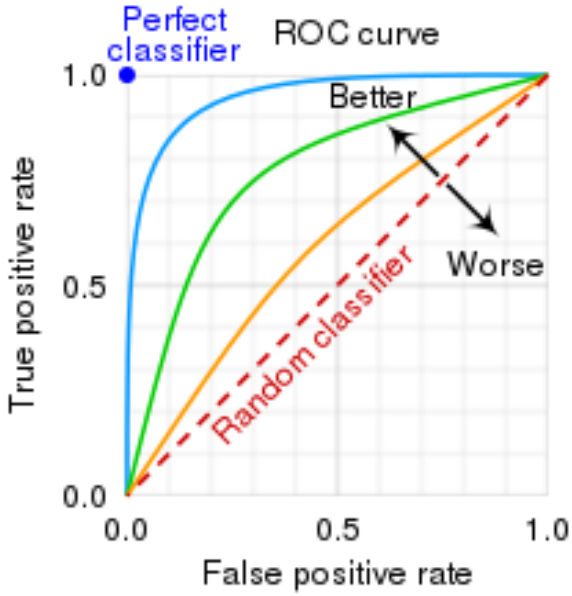

Figura 2.15 – Representação da *ROC curve* (Wikipedia, 2003)

.

$$\text{Revocação (}\textit{Recall}\text{)} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{2.8}$$

- ***F1-Measure*** (Equação 2.9): mede o qual preciso e qual robusto é o modelo.

$$\text{F1-Measure} = 2 \times \frac{\text{Precisão} \times \text{Revocação}}{\text{Precisão} + \text{Revocação}} \tag{2.9}$$

- ***Receiver operating characteristic curve (ROC curve)*** (Figura 2.15): curva plotada em um eixo de coordenadas como taxa de TP versus taxa de FP, onde a taxa TP está no eixo Y e a taxa FP está no eixo X. A aréa sob a curva (AUC) indica que, se próximo de 1, o modelo tem alta capacidade de separação de classes; se próximo de 0, o modelo não tem capacidade de separação de classes.

2.2 Trabalhos Relacionados

Nesta seção, é apresentado o embasamento científico também necessário para a definição da metodologia deste trabalho por meio da descrição de trabalhos relacionados. Uma grande quantidade de estudos são focados em prever o desempenho de discentes do ensino superior utilizando diversos parâmetros e modelos de predição. Alguns são focados especialmente para plataforma de ensino *online*, como Bravo-Agapito, Romero e Pamplona (2021); outros são mais focados em realização de exames *online* (Tomasevic; Gvozdenovic; Vranes, 2020); e a maioria no desempenho dos discentes na formação acadêmica de forma generalizada, como Yağcı (2022) e Kumar *et al.* (2011). São apresentados desses trabalhos quais estratégias foram utilizadas, desde

os dados preparados para aplicação dos algoritmos dos modelos de predição, os próprios modelos de predição e seus respectivos resultados.

Alyahyan e Düştegör (2020) efetuaram uma revisão completa sobre os usos dos algoritmos de modelos de predição para prever desempenho dos discentes na área acadêmica. Foi verificado que a maioria dos artigos para previsão do desempenho dos discentes está focada na aplicação algoritmos de classificação: árvores de decisão, Naive Bayes, RNAs e KNNs, com 78% de recorrência, sendo as árvores de decisão as mais recorrentes com 44%. Em segundo lugar, há os algoritmos de regressão logística e linear com 3%. Por fim, há os algoritmos de Clusterização com 2%.

Visando uma melhor organização dos trabalhos relacionados, as Subseções 2.2.1 e 2.2.2 apresentam, respectivamente, trabalhos sobre previsão de desempenho semestral e previsão de desempenho em exames *online*

2.2.1 Previsão de desempenho na formação acadêmica

Yağcı (2022) experimentou diversos modelos de predição do desempenho de discentes em um semestre utilizando algoritmos de classificação. Para isso, o autor utilizou uma base de dados do SIS contendo registros de 1854 discentes de diferentes cursos de uma universidade estadual da Turquia que realizaram a matéria de Língua Turca 1. Foram utilizados, como mencionado, o Naive Bayes, Random Forest, KNN, RNA e Regressão Logística. Após tratamento dos dados, selecionando os parâmetros e variáveis da melhor maneira e discretizando as notas dos exames finais desses discentes, os modelos foram treinados e testados. Como se pode visualizar na Figura 2.2, experimentos práticos considerando tais modelos e as métricas AUC, acurácia, *F1-measure*, precisão e *Recall*, mostraram que a RNA e a Random Forest obtiveram os melhores resultados. Se observou que algoritmos mais simples como Naive Bayes e KNN não obtiveram resultados tão bons. Já a regressão logística obteve resultados medianos. Talvez os resultados não tenham sido tão satisfatórios para os modelos de Naive Bayes e KNN, por conta da pouca quantidade de variáveis ou até mesmo variáveis pouco significativas utilizadas para o treinamento dos mesmos no intuito de predizer a nota final do discente, entendendo se ele foi ou não aprovado naquele semestre; no caso, melhores atributos poderiam ser utilizados para aperfeiçoar essa previsão, como dados demográficos do discente ou mais dados prévios do discente, como a média acumulada de notas.

Complementado esse último artigo, Kumar *et al.* (2011) utilizaram critérios diferentes quanto à performance do discente em um único semestre para criação de uma estratégia de predição que utiliza apenas de modelos implementados por meio de algoritmos de árvores de decisão, o qual não havia no artigo de Yağcı (2022). Como conjunto de dados para essa tarefa, foi utilizado um conjunto de 115 discentes de ciência da computação da faculdade de PRIST, o que é um número de instâncias bem reduzido em comparação ao trabalho anterior, porém os critérios de predição foram bem escolhidos, com notas de 5 disciplinas do primeiro semestre da

Model	(AUC)	Classification accuracy (CA)	F1	Precision	Recall
Random Forest	0.860	0.746	0.721	0.752	0.746
Neural Network	0.863	0.746	0.723	0.748	0.746
SVM	0.804	0.735	0.704	0.735	0.735
Logistic Regression	0.826	0.717	0.685	0.700	0.717
Naïve Bayes	0.810	0.713	0.692	0.706	0.713
kNN	0.810	0.699	0.694	0.691	0.699

Tabela 2.2 – AUC, Acurácia, *F1-measure*, Precisão e *Recall* dos modelos (Yağcı, 2022)

.

faculdade, com marcas internas (provas) e externas (trabalhos) nessas disciplinas. Foram testados e comparados dois algoritmos de árvore de decisão: J48 e ID3 apenas com as marcas externas. Os resultados foram satisfatórios para os dois modelos que utilizaram dessas árvores de decisão, com acurácia de 92.2% e 88.8%. Assim, o J48 foi mais preciso que o ID3. Entretanto, pensando em uma maior quantidade de discentes, é importante adicionar mais instâncias para o treinamento e teste do modelo implementado por árvore de decisão. Como é apontado por Carraro (2019), possivelmente colocando mais instâncias variadas para os dados de teste, ocorreria um *overfitting* e a classificação não obteria bons resultados. Além disso, seria necessário a adição de mais variáveis e atributos de entrada.

Portanto, esses dois artigos possuem objetivos muito semelhantes com este trabalho, visto que apresentam e testam duas estratégias de predição diferentes com diversos modelos de predição para prever o desempenho de discentes academicamente. Assim, antes mesmo da utilização desses modelos na estratégia proposta para a plataforma TôSabendo (Subseção 2.1.1), é importante entender, para o contexto da plataforma TôSabendo, quais deles podem obter resultados piores ou melhores e por quais motivos; entretanto, tanto em Yağcı (2022) quanto em Kumar *et al.* (2011), há pouca diferenciação de parâmetros usados para cada modelo, assim como poucas variáveis de entrada, o que pode ter dificultado a predição para os modelos citados (Alyahyan; Düştegör, 2020).

Além disso, especialmente pelo fato de testarem o desempenho de discentes em sua formação acadêmica, ainda que não seja focado especificamente para plataformas de ensino *online*, já é possível entender os benefícios os quais essa previsão fornece para as IESs. Assim como aponta (Han; Pei; Tong, 2022), esses estudos podem ajudar as IESs a estabelecerem uma estrutura de análise de aprendizagem e a contribuir para os processos de tomada de decisão.

2.2.2 Previsão de desempenho em exames *online*

Considerando a previsão de desempenho em exames *online*, Tomasevic, Gvozdenovic e Vranes (2020) realizaram a predição de desempenho de discentes de forma bem completa.

A previsão do desempenho do discente no exame é vista como particularmente importante para identificar os discentes que provavelmente serão reprovados no exame final, de modo que assistência adicional ou tutoria possa ser fornecida a tempo, ou até mesmo uma oferta de caminhos de aprendizado personalizados ou materiais de avaliação para discentes individuais. Ao mesmo tempo, pode fornecer informações relevantes aos educadores para que possam ser recrutados professores adicionais ou planejadas intervenções para apoiar os discentes (individuais ou grupos), ou para identificar cursos e programas de ensino que precisam ser melhorados. Portanto, encontra-se relacionado ao escopo deste trabalho que é identificar o desempenho de discentes em uma plataforma de ensino *online*, visando ajudar o discente. Para isso, Tomasevic, Gvozdenovic e Vranes (2020) utilizaram um grande conjunto de dados, que se encontra publicamente disponível, denominado de *Open University Learning Analytics Dataset* (OULAD) Kuzilek, Hlosta e Zdrahal (2017) para realizar uma análise abrangente e comparação de técnicas de modelos de predição de última geração. Nessa base de dados, os dados foram recolhidos no âmbito dos módulos (cursos específicos) indicados como “DDD_2013J” e “DDD_2014B”, que representam um período acadêmico daquele curso abrangendo um assunto de aprendizagem específico e representando um conjunto de sessões (cada uma terminando com uma avaliação intermediária), onde cada um desses cursos termina com o exame final que determina (em combinação com as avaliações anteriores) se o discente foi aprovado ou reprovado no curso. Existem na base 3166 discentes inscritos (excluindo os discentes que não realizaram o exame final).

Como pode-se visualizar na Tabela 2.3, a análise consistiu em fatores como demografia do discente, desempenho do discente nas avaliações do curso e nos dados de engajamento do discente. Mais precisamente, o desempenho do discente considerou a pontuação nas avaliações intermediárias (no caso da estrutura de dados de entrada apresentada foram seis avaliações intermediárias no módulo correspondente) e no exame final do curso, e o número de tentativas de exame. Isso se assemelha muito a forma como a TôSabendo funciona, onde os Quizzes, contendo os intermediários e um final com todos os assuntos apresentados anteriormente, podem ser realizados quantos vezes que for necessário, sendo que o *Quizz* final é o mais importante de todos e diz realmente se o discente “está sabendo”. Além disso, de acordo com a Tabela 2.3, os dados demográficos foram levados em consideração como um conjunto de parâmetros de entrada descritivos e potencialmente relevantes, pois podem conter informações que não são desprezíveis. As instâncias contendo os valores ausentes não foram levadas em consideração e foram excluídas dos dados de interesse. Outros dados disponíveis, que geralmente têm diferentes faixas de valores, foram dimensionados e normalizados para serem iguais e para equilibrar a influência de diferentes recursos relacionados ao discente.

Todo o conjunto de dados de interesse foi dividido aleatoriamente em subconjuntos designados para fins de treinamento, validação e teste. Ou seja, no caso em que o algoritmo subjacente exigia apenas dois subconjuntos de dados servindo para fins de treinamento e teste, o conjunto de dados inicial foi dividido na proporção de 80%:20%, respectivamente. Por outro lado, quando o algoritmo subjacente precisava de três subconjuntos de dados diferentes, como

Feature element ID	1	2	3	4-9	10-15	16	17
Feature type	**DEMOGRAPHIC**			**ENGAGEMENT**	**PERFORMANCE**		
Feature description	Gender	Highest Education	Age	<Sum Of Clicks> (per assessment)	<Scores per Assessment>	No of Attempts	Final Exam Score
Feature value range	[0,1]	[0, 0.25, 0.5,0.75,1]	[0,0.5,1]	[0-1]	[0-1]	[0-1]	[0-1]
Feature value description	0: male 1: female	0: no formal 0.25: below A level 0.5: A level or equivalent 0.75: HE qualif. 1: Post graduate	0: <35 0.5: 35-55 1: >55	0-N scaled to [0-1]	0-100 scaled to 0-1	0-N scaled to [0-1]	0-N scaled to [0-1]

Tabela 2.3 – Estrutura dos dados analisados (características relacionadas ao discente) (Tomasevic; Gvozdenovic; Vranes, 2020)

.

para treinamento, validação e teste, a proporção de 60%:20%:20% foi aplicada. Desta forma, a intenção foi comparar a eficácia das técnicas analisadas na mesma porção do conjunto de dados inicial (ou seja, 20%).

Como no caso dos métodos de classificação, foi necessário classificar o discente em dois grupos: “reprovado” ou “aprovado”, os resultados do exame final (ou seja, a pontuação real do exame final representada pelo valor no intervalo de 0 a 100) tiveram que ser previamente convertidos em dois valores, ou seja, duas classes, “0” ou “1”. Tal categorização foi necessária por tratar-se de uma tarefa de classificação binária (os discentes podem passar ou não no exame). Por outro lado, para **tarefas de regressão**, foi utilizada a nota real (na faixa de 0 a 100) que o discente obteve no exame final. Em ambos os casos, considerou-se aprovado no exame o discente que obteve mais de 40 (do total de 100); caso contrário, considerou-se reprovado no exame.

Para a resolução das tarefas de classificação, foram considerados os algoritmos KNN, SVM[2], ANN, Árvore de Decisão, Naive Bayes e Regressão Logística[3]. Como uma métrica quantitativa geral, o valor F1 foi utilizado. Para a análise de regressão, foram utilizados KNN, SVM, ANN, Árvore de Decisão, Regressão Bayesiana e Regressão Linear. Para avaliar e comparar os algoritmos de regressão, o *Root Mean Square Error* (RMSE)[4] foi escolhido como uma métrica quantitativa. Além disso, para garantir maior validade da avaliação e das métricas quantitativas, os resultados apresentados foram obtidos pela média das métricas calculadas a partir de 10 tentativas independentes realizadas para cada modelo analisado. Como pode-se visualizar nas Tabelas 2.4 e 2.5 com os resultados para classificação e regressão, respectivamente, a partir de

[2] SVM busca encontrar um hiperplano no espaço de características que melhor separa as classes de dados. O hiperplano é escolhido de forma que maximize a margem entre as duas classes, ou seja, a distância entre os pontos mais próximos de cada classe que estão mais próximos do hiperplano. Esses pontos são chamados de vetores de suporte.

[3] A regressão logística modela a relação entre as variáveis independentes (ou características) e a probabilidade de pertencer a uma classe usando a função *logit*. A função *logit* transforma a probabilidade linear em uma escala que varia de menos infinito a mais infinito, mapeando-a para o intervalo de 0 a 1.

[4] RMSE é uma métrica usada para avaliar a qualidade do ajuste de um modelo em problemas de regressão. Calcula a diferença média entre as previsões feitas pelo modelo e os valores reais dos dados, elevando essa diferença ao quadrado, calculando a média desses quadrados e, por fim, tirando a raiz quadrada desse valor médio.

	D	E	P	D + E	D + P	E + P	D + E + P
k-NN (no weights)	0.6173	0.9146	0.94	0.8886	0.9406	0.939	0.9423
k-NN (distance weights)	0.6136	0.9124	0.9344	0.8906	0.938	0.9423	0.9453
SVM (linear kernel)	0.7202	0.7202	0.9288	0.934	0.9382	0.9608	0.9622
SVM (RBF kernel)	0.7202	0.942	0.9377	0.932	0.946	0.9565	0.9604
ANN (2x1)	0.7127	0.9505	0.9404	0.9487	0.947	0.9662	0.9645
Decision trees	0.6436	0.9454	0.9386	0.9473	0.9388	0.9507	0.9507
Naïve Bayes	0.567	0.8458	0.9207	0.8497	0.9236	0.9172	0.9135
Regularized logistic regression	0.6739	0.8896	0.9317	0.8927	0.9335	0.9336	0.9442

Tabela 2.4 – Comparação de desempenho (F1) para previsão de resultado do exame final – classificação (D – demografia, E – engajamento, P – dados de desempenho). (Tomasevic; Gvozdenovic; Vranes, 2020)

.

diferentes parâmetros para cada modelo de predição, eles foram treinados e testados com todas combinações de fatores (dados de entrada) pra predição e individualmente: D – demografia, E – engajamento, P – dados de desempenho. Nessas Tabelas (2.4 e 2.5), os valores grifados de verde indicam dados de entrada ótimos para determinado modelo, os valores grifados de amarelo indicam o modelo ótimo para os dados de entrada fornecidos e os valores grifados de vermelho indicam o melhor modelo geral.

Frente ao exposto, com base nos resultados apresentados para a tarefa de classificação, percebeu-se que a maior eficácia (ou seja, o maior valor F1) foi obtida explorando todos os três tipos de dados disponíveis (D + E + P) na maioria dos modelos (KNN com e sem ponderação de distância, SVM com *kernels* lineares e RBF, árvores de decisão e regressão logística), sendo que a abordagem Naive Bayes apresentou os melhores resultados para combinação de demografia e desempenho (D + P), enquanto combinação de engajamento e desempenho (E + P) deu a maior eficácia nos modelos de ANN e, novamente, nas árvores de decisão. Para a tarefa de regressão, pode-se ver que a maior eficácia (ou seja, o menor valor de RMSE) foi obtida com a combinação de dados de engajamento e desempenho (E + P) em casos de ANN, árvores de decisão, SVM com *kernel* RBF e KNN com distâncias ponderadas; já ao considerar o conjunto de todos os três tipos de dados disponíveis (D + E + P), os melhores resultados foram obtidos com os modelos de SVM com *kernel* linear, regressão bayesiana e regressão linear. Logo, entende-se que combinar mais variáveis e atributos podem ajudar a aperfeiçoar o modelo de predição ao invés do uso de apenas um dos fatores de desempenho existentes, como o desempenho prévio.

Como se observou no artigo Yağcı (2022) da Subseção 2.1.3, é interessante notar que a maior precisão geral foi obtida com RNA, tanto para classificação quanto para regressão, porém dessa vez, em Tomasevic, Gvozdenovic e Vranes (2020), com mais dados. Em tal artigos, os dados utilizados foram dados de engajamento e dados de desempenho passado, o que significa que os dados demográficos não mostraram influência significativa na precisão neste caso específico. De acordo com Garg (2018), utilizar ou não dados demográficos junto com esses outros

	D	E	P	D + E	D + P	E + P	D + E + P
k-NN (no weights)	36.5897	20.0686	15.6062	23.0584	16.1149	15.6144	16.0487
k-NN (distance weights)	37.4798	19.9535	15.9446	22.4458	16.2916	15.5655	15.9699
SVM (linear kernel)	42.0458	34.9426	17.0883	34.6037	16.9518	16.7734	16.7158
SVM (RBF kernel)	40.1914	18.4868	15.8578	20.6338	15.8598	14.7474	14.9454
ANN (2x2x1)	36.3446	15.0223	15.3591	15.3355	14.7255	12.1256	12.3498
Decision trees	37.7511	16.807	16.8671	16.2454	14.1399	12.997	13.3465
Bayesian linear regression	36.9372	30.8343	16.8663	30.8946	30.8946	16.5688	16.4089
Regularized linear regression	36.5821	30.2297	16.7771	29.7517	16.6695	16.2897	16.2187

Tabela 2.5 – Comparação de desempenho (RMSE) para previsão do resultado do exame final – regressão (D – demografia, E – engajamento, P – dados de desempenho). (Tomasevic; Gvozdenovic; Vranes, 2020)

dados de engajamento e desempenho gera divisões de opiniões na literatura. No caso dos dados demográficos existirem na base de dados, é recomendável utilizá-los; caso contrário, não são de grande importância e não há necessidade de buscá-los.

Portanto, com base nesse artigo principal desta seção e dos demais artigos apresentados na seção 2.2.1, observa-se que os tipos de dados mais eficazes pra realizar a previsão geral do desempenho dos discentes são o desempenho prévio e sua demografia. Para o desempenho prévio de discentes, os dados mais comumente utilizados são notas de avaliações, exames e atividades extraclasse, e a média geral do mesmo naquele curso ou em uma disciplina. Já para a demografia do discente, os dados mais comumente utilizados são o gênero e a idade do mesmo. Apesar do desempenho prévio demonstrar ser o mais impactante na previsão do desempenho do discente no conjunto dos artigos apresentados, combinar dois ou mais tipos de dados demonstrou auxiliar ainda mais na previsão como identificado por Tomasevic, Gvozdenovic e Vranes (2020). Em relação aos modelos de predição para previsão do desempenho dos discentes, os mais eficazes foram os que utilizaram os algoritmos de classificação, a saber:

1. redes neurais artificiais (RNAs) ou *artificial neural networks* (ANNs) a partir do treinamento utilizando *backpropagation*, em que diferentes RNAs *feedforward*, com uma e duas camadas intermediárias, foram testadas;
2. árvores de decisão, considerando diferentes parâmetros mínimos de nós pais e nós folhas;
3. Naive Bayes com parâmetros extraídos do conjunto de dados de treinamento;
4. KNNs sem e com ponderação de distância e com um número de vizinhos variando de [0-50].

Dessa forma, esses tipos de dados e modelos de predição, com os devidos parâmetros, foram testados para desenvolver e avaliar a estratégia de predição proposta, focando na previsão do desempenho de discentes na plataforma TôSabendo.

3 Desenvolvimento

Como já mencionado anteriormente (vide Seção 1.2), este trabalho possui, como objetivo geral, a proposta, o desenvolvimento e a validação de uma estratégia para a predição do desempenho de discentes na plataforma TôSabendo, visando compreender mais amplamente o desempenho dos estudantes ao abordarem as questões dos Quizzes da plataforma, o que pode permitir aos educadores intercederem, eficaz e antecipadamente, no processo de ensino-aprendizagem. Para tanto, vários estudos foram realizados para prever o desempenho de discentes. Dentre eles, diferentes modelos de predição e diferentes técnicas de EDM foram abordados para atingir esse objetivo.

A partir desses estudos, este capítulo tem, como finalidade, apresentar detalhadamente a estratégia proposta e a explicação de cada decisão tomada para a composição da mesma. Desta forma, encontra-se assim delineado: a Seção 3.1 apresenta as arquiteturas de funcionamento associadas à estratégia proposta para alcançar o objetivo geral deste trabalho e a Seção 3.2 descreve a remodelagem do banco de dados da TôSabendo no intuito de permitir a futura integração da estratégia de predição proposta à plataforma.

3.1 Arquiteturas de funcionamento

Esta seção apresenta as arquiteturas desenvolvidas para compreender a estratégia de predição proposta deste trabalho e o respectivo uso da mesma para a futura atualização da plataforma TôSabendo. Para isso, foi feita uma divisão em duas arquiteturas: uma dedicada à verificação de modelos de predição para a plataforma TôSabendo, e outra destinada à análise desses modelos.

Para se definir os bons modelos de predição que foram utilizados na estratégia de predição proposta neste trabalho, foi definida, inicialmente, uma arquitetura que verifica distintos modelos de predição seguindo processos de EDM (vide Seção 2.1.4). Tal arquitetura encontra-se descrita na Figura 3.1. Observa-se, de forma geral, que o fluxo tem início com um breve processamento de dados, seguido da aplicação e análise de diversos modelos de predição. Foi considerado, para toda essa arquitetura, que os discentes podem ser divididos em dois grupos: o de discentes novatos, os quais acabaram de adentrar no IES, e os discentes veteranos, os quais possuem pelo menos 1 período concluído. Por conta disso, os passos a seguir da arquitetura foram realizados para cada grupo:

- Passo 1: consiste no fornecimento de dados sintéticos de discentes da plataforma, que são usados para validação de modelos de predição. Os dados sintéticos podem ser criados

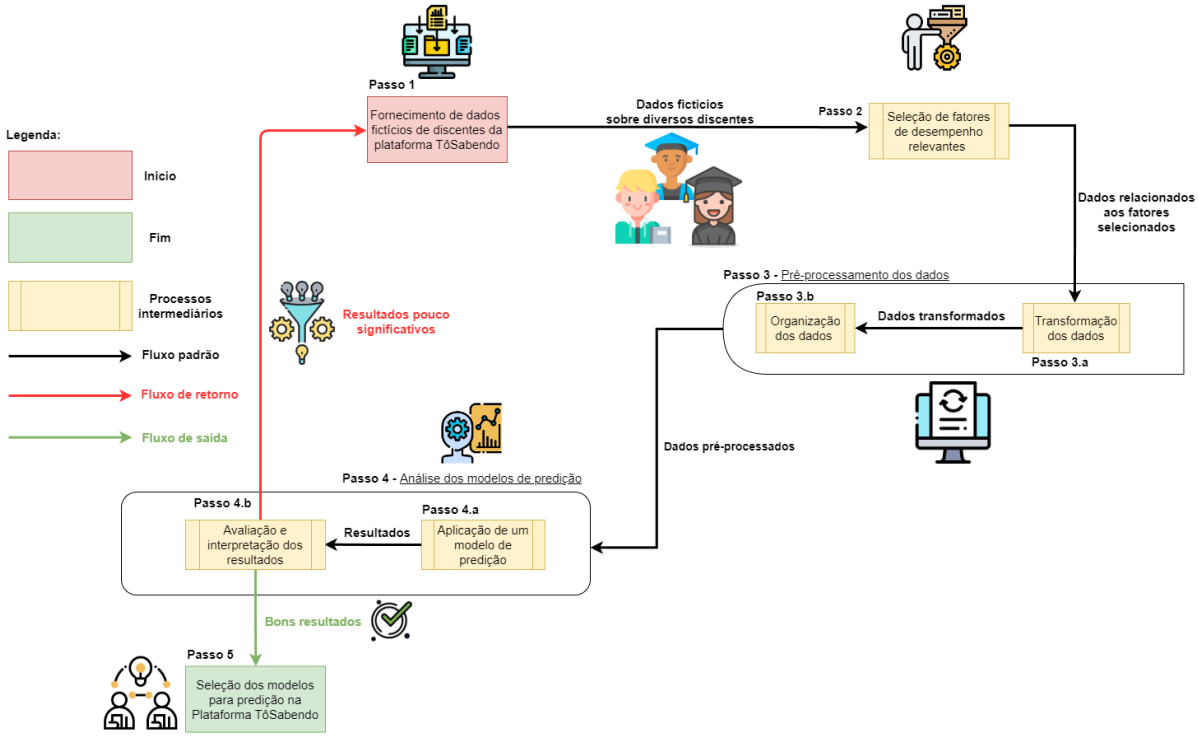

Figura 3.1 – Verificação de modelos de predição para a plataforma TôSabendo

manualmente considerando atributos do esquema relacional do banco de dados, assim como dados adicionais, como o desempenho prévio do discente (histórico escolar e acadêmico).

- Passo 2: consiste na seleção, dentre os dados fornecidos pelo Passo 1, daqueles considerados mais importantes (fatores) para uma determinada predição. Como apresentado na Seção 2.1.3, diversos fatores podem influenciar diretamente no desempenho de discentes, como, por exemplo, desempenho acadêmico prévio, demografia do discente e o ambiente escolar. Neste passo, levando em conta que existem dois grupos de discentes, a seleção dos fatores será diferente para os dois, como, por exemplo, o discente novato ainda não possui notas acadêmicas e, por conta disso, é necessário usar apenas o seu desempenho pré-acadêmico com notas escolares, enquanto o discente veterano possui tanto notas escolares quanto acadêmicas.

- Passo 3: consiste no pré-processamento dos dados, que é uma etapa anterior à implementação e aplicação dos modelos de predição. Nela, os dados devem passar por um último tratamento antes de serem usados para treinamento e teste dos modelos. Este pré-processamento de dados ocorre a a partir de processos que envolvem a transformação dos dados (Passo 3.a) e a organização dos dados (Passo 3.b), descritos a seguir.

- Passo 3.a: consiste na transformação de dados, que é necessária para eliminar dissimilaridades no conjunto de dados. Para isso, é feita uma normalização dos atributos numéricos, para que todos contribuam igualmente para os modelos, evitando a predominância de atributos com valores grandes, como, por exemplo, o reescalonamento de atributos no

intervalo de 0 a 1. Além disso, é feita uma conversão dos dados para variáveis numéricas, com codificação de rótulos para essas variáveis (por exemplo, usando um valor de 0 até N), e com variáveis "*dummy*", que utilizam 0 ou 1 pra representar variáveis categóricas, sendo 1 a presença de uma categoria e 0 caso contrário. Após essas transformações nos dados, eles são encaminhados para o Passo 3.b, onde são organizados.

- Passo 3.b: consiste na organização de dados oriundos do Passo 3.b que se encontram desequilibrados, mesmo depois de transformados. Um exemplo disso é uma disparidade significativa entre o número de registros de diferentes classes como, por exemplo, discentes reprovados versus discentes aprovados em uma disciplina. Assim, é necessário fazer uma reamostragem, aumentando ou reduzindo instâncias de dados de forma aleatória ou com técnicas para balanceamento, como a técnica SMOTH, que gera instâncias de dados sintéticas para a classe minoritária, o que evita, assim, o problema de *overfitting* que é gerado pela criação de dados sintéticos.

- Passo 4: consiste da análise de modelos de predição, que utilizam os dados pré-processados advindos do Passo 3 a fim de serem aplicados, avaliados e interpretados. Dessa forma, compreende-se quais são os melhores modelos de predição a serem adicionados na plataforma TôSabendo, sendo um para discentes novatos e outro para discentes veteranos. Essa análise de modelos de predição envolve dois passos principais, Passo 4.a e Passo 4.b, expostos a seguir.

- Passo 4.a: consiste na aplicação de um modelo de predição com os dados pré-processados do Passo 3, os quais são utilizados como dados de entrada. Esse modelo de predição é implementado por meio de um algoritmo de classificação e é treinado e testado, gerando resultados. Esses resultados são encaminhados para o Passo 4.b, onde são avaliados.

- Passo 4.b: consiste na avaliação e interpretação dos resultados oriundos do Passo 4.a, aplicando algumas métricas de avaliação (Seção 2.1.4.6) para em seguida interpretar se o modelo obteve bons resultados a partir de tabelas e gráficos contendo os resultados atuais, resultados anteriores e os de outros modelos. Assim, se o modelo obtém resultados pouco significativos, ele pode passar novamente por todo ajuste dos dados, desde o fornecimento inicial (Passo 1), como apresentado pelo fluxo de retorno; se o modelo obtém bons resultados e melhores que os de outros, ele é considerado um bom modelo de predição, como apresentado pelo fluxo de saída. Dessa forma, alguns modelos são descartados e outros encaminhados para o Passo 5, para serem incorporados à plataforma TôSabendo.

- Passo 5: consiste na seleção dos bons modelos de predição, um para discentes novatos e outro para veteranos, para ser utilizado futuramente na plataforma TôSabendo a fim de realizar a predição do desempenho na mesma sempre que os discentes cadastrarem pela primeira vez.

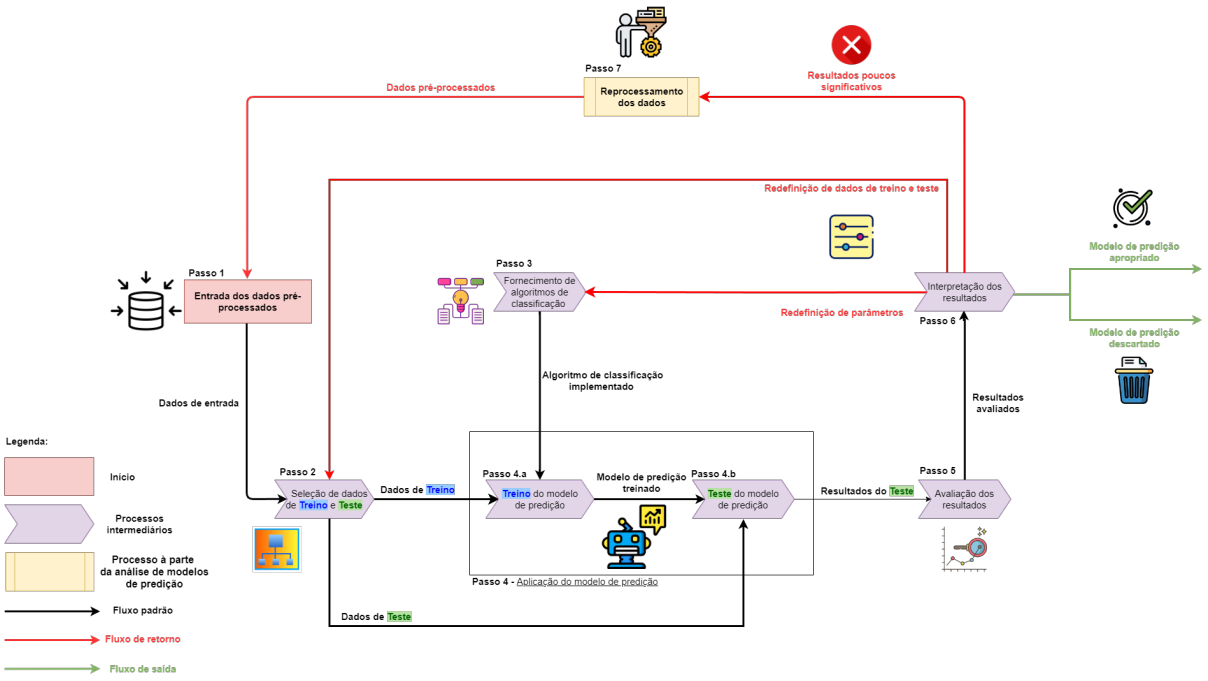

Figura 3.2 – Análise de modelos de predição

Particularmente, considerando apenas o Passo 4 da Figura 3.1, foi desenvolvida uma arquitetura (vide Figura 3.2) de como o mesmo foi realizado. Ainda levando em conta os dois grupos de discentes, essa arquitetura visou encontrar, aplicada separadamente, o melhor modelo de predição para o discente novato e para o veterano, seguindo os seguintes passos:

- Passo 1: consiste na entrada de dados que passaram por todo um tratamento e são usados como entrada para análise do modelo de predição.
- Passo 2: consiste na divisão dos dados de entrada oriundos do Passo 1, selecionando quais dados são de treinamento e quais são de teste. Essa seleção pode ser feita de diversas maneiras, porém, inicialmente, é utilizada a mais trivial, que é realizar a separação em porcentagens, como, por exemplo, 80% dos dados para treinamento e 20% para teste ou em outras proporções de 70% e 30%. Com esses dados de treino e teste selecionados, eles já podem ser usados em suas respectivas funções de treinar e testar o modelo de predição, nos Passos 4.a e 4.b respectivamente.
- Passo 3: consiste no fornecimento de um algoritmo de classificação implementado para desenvolver o modelo de predição desejado. Os algoritmos de classificação usados encontram-se descritos na Seção 2.1.2, como, por exemplo, árvore de decisão, KNN, Naive Bayes ou RNA.
- Passo 4: consiste na aplicação do modelo de predição em questão com os dados de entrada selecionados em treino e teste, oriundos do Passo 2, e também com o algoritmo de classifi-

cação implementado, oriundo do Passo 3. Para isso, o modelo deve passar pelos Passos 4.a e 4.b, descritos a seguir.

- Passo 4.a: consiste no treinamento (*fitting*) do modelo de predição com os dados de treino oriundos do Passo 2, para o mesmo "aprender" sobre padrões de discentes. Esse modelo de predição treinado é testado no Passo 4.b.

- Passo 4.b: consiste no teste do modelo de predição treinado a partir dos dados de teste, oriundos do Passo 2, gerando uma série de resultados que são as próprias predições do modelo, indicando se os discentes têm ou não um bom desempenho na plataforma. Esses resultados são avaliados no Passo 5.

- Passo 5: consiste na avaliação dos resultados do teste e treinamento do modelo de predição, oriundos do Passo 4.b. Para isso, são aplicadas medidas de eficácia (vide Seção 2.1.4.6) como, por exemplo, a acurácia ou a *F1-measure*, as quais são estabelecidas por meio de uma matriz de confusão. Essa avaliação indica, por meio de comparações, quais dados foram preditos de maneira correta, ou seja, se o discente que foi predito como com um bom desempenho realmente se saiu bem na plataforma ou se o discente que foi predito como com um desempenho ruim realmente não se saiu bem. Os resultados avaliados são encaminhados para um interpretação no Passo 6.

- Passo 6: consiste na interpretação manual dos resultados avaliados oriundos do passo 5 por meio da visualização de gráficos e tabelas com os valores encontrados das métricas. Por meio dessa interpretação, podem surgir 3 fluxos de retorno: o primeiro considera que os parâmetros dos algoritmos de classificação foram mal definidos e, por isso, precisam ser redefinidos, retornando ao Passo 3. O segundo considera apenas que os dados de treino e teste foram mal selecionados e, por isso, precisam ser divididos novamente no Passo 2 para encontrar o melhor conjunto. E o último e mais trabalhoso considera que o modelo obteve resultados pouco significativos, ou seja, o treino do modelo não obteve bons resultados em nenhuma das ocasiões, seja alterando os parâmetros ou alterando a divisão dos dados; nesse caso, é necessário realizar todo reprocessamento dos dados (Passo 7). Como fluxo de saída, após interpretar que os resultados do modelo foram sempre piores que o de outros modelos, ainda que refazendo os Passos 2 e 3 e realizando o Passo 7, descarta-se o uso do mesmo, enquanto que, se os resultados forem significativos e bem melhores que de outros modelos, será apropriado este modelo de predição para a plataforma TôSabendo, seja ele para o discente novato ou veterano.

- Passo 7: consiste no reprocessamento de dados, que envolve os Passos 1 a 3 da arquitetura presente na Figura 3.1. Portanto, é um novo tratamento dos dados na busca por resultados aprimorados, em que os dados tratados posteriormente são empregados como entrada para o modelo (Passo 1).

Frente ao exposto, após realizar todos passos da arquitetura da Figura 3.1 e da Figura 3.2, tornou-se possível a integrar à futura plataforma TôSabendo funcional os modelos de predição para previsão do desempenho dos discentes da mesma.

3.2 Remodelagem do banco de dados da TôSabendo

Para manipulação dos dados da Plataforma TôSabendo, foi necessário remodelar o banco de dados a fim de adicionar novas informações relacionadas à predição de desempenho dos discentes, corrigir algumas inconsistências e, por fim, implantá-lo em algum sistema gerenciador de bancos de dados. Desta forma, as Subseções 3.2.1 e 3.2.2 descrevem, respectivamente, como o esquema conceitual EER foi atualizado com as novas informações relevantes e como o banco de dados foi implantado em um sistema gerenciador.

3.2.1 Atualização do esquema conceitual EER

Para atualização do esquema EER, foi empregada a ferramenta TerraER[1] para incorporação de novos dados, como, por exemplo, histórico de discentes, com suas notas e disciplinas cursadas, e predições realizadas para discentes. Como se pode visualizar na Figura 3.3, novas entidades, atributos e relacionamentos foram incorporados com foco na predição de desempenho do discente. Dentre as entidades, há o Histórico Acadêmico, que é associado apenas aos discentes (jogadores) que já cursaram pelo menos um período durante a faculdade, ou seja, os discentes veteranos, já que os históricos fornecem notas média e coeficientes, não ainda presentes para discentes de 1º período. Além disso, completando o Histórico Acadêmico, há a entidade Disciplina Cursada, identificada por seu atributo "código", como, por exemplo, BCC392 referente a disciplina Monografia 1, e um atributo "semestre", por exemplo, 2023/1 referente ao primeiro semestre do ano de 2023. Para cada disciplina de um histórico acadêmico de um discente, é indicado sua média naquele semestre e se o discente foi aprovado.

Além do histórico acadêmico, considerou-se o histórico escolar do discente no ensino médio como um importante fator de desempenho, principalmente para discentes novatos, que ainda não possuem notas acadêmicas. Dessa maneira, ponderou-se que uma entidade Histórico Escolar deve guardar a nota média, a escola do ensino médio, a data de conclusão e as notas médias nas disciplinas mais relevantes quando relacionados a cursos de IESs, que são português e matemática. É necessário ressaltar que cada discente possui apenas um histórico acadêmico e um histórico escolar. Por isso, o atributo que os identifica unicamente é o próprio "id" do discente, ou seja, seu CPF.

Ademais, adicionou-se uma entidade Predição especificamente para guardar as predições realizadas para o desempenho dos discentes na plataforma. Dessa forma, considerou-se que um

[1] O TerraER é uma ferramenta de modelagem de código aberto projetada para apoiar os discentes na criação de modelos Entidade-Relacionamento. Disponível em http://www.terraer.com.br/.

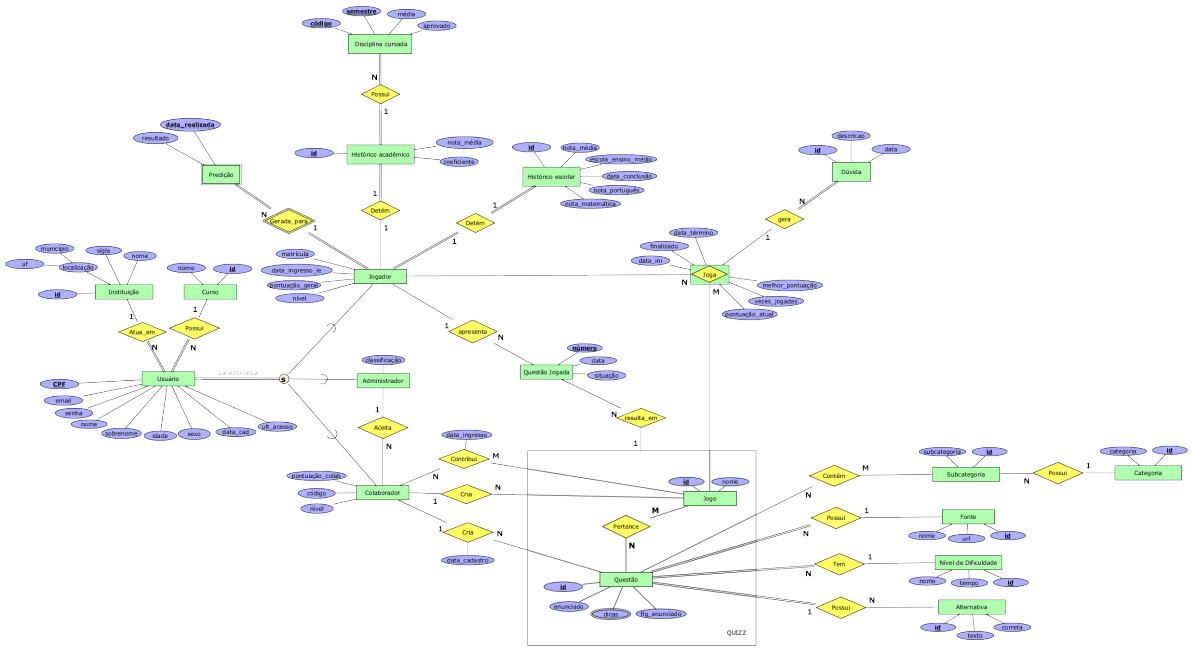

Figura 3.3 – Esquema conceitual EER atualizado

discente pode ter várias predições, que se identificam unicamente pelo atributo “CPF” do mesmo junto com a data que predição foi realizada. O atributo “resultado” identifica o resultado da predição realizada pelos modelos de predição, podendo ser “bom desempenho”, “desempenho médio” ou “desempenho inferior”.

Por fim, foram alterados alguns dados e definições no banco de forma a deixá-lo mais consistente, a saber:

- alteração da chave primária da entidade Usuário de *e-mail* para CPF, já que é um atributo que também único e é mais fácil de manipulá-lo em consultas e escritas no banco;
- exclusão da entidade Localidade e substituição por um atributo “localização” na entidade Instituição, pois não há necessidade de guardar a localização do usuário, mas, sim, da IES que o mesmo frequenta;
- adição do atributo “idade” na entidade Usuário, referente a idade do mesmo, como forma de usá-la como um fator de demografia do discente na predição;
- complementação da entidade Jogador (discente) com os atributos “matrícula” e “data de ingresso a IES”, com o objetivo de obter seu histórico e entender se o mesmo é novato ou veterano;
- adição de uma nova entidade Dúvida, que indica as dúvidas dos discentes geradas quando os mesmos estiverem jogando algum *Quizz*, guardando a descrição e a data em que foi realizada.

```
model Usuario {
    cpf            String   @id @db.VarChar(12)
    email          String   @unique @db.VarChar(100)
    senha          String   @db.VarChar(100)
    nome           String   @db.Text
    sobrenome      String   @db.Text
    sexo           String   @db.Char(2)
    idade          Int
    data_cadastro  DateTime @default(now())
    updated_at     DateTime @updatedAt
    ultimo_acesso  DateTime
    id_curso       Int
    id_instituicao Int

    curso       Curso       @relation(fields: [id_curso], references: [id], onDelete: Restrict, onUpdate: Restrict)
    instituicao Instituicao @relation(fields: [id_instituicao], references: [id], onDelete: Restrict, onUpdate: Restrict)

    Jogador       Jogador[]
    Colaborador   Colaborador[]
    Administrador Administrador[]

    @@map("usuarios")
}
```

Figura 3.4 – *Model* da tabela Usuário

Por meio do esquema conceitual EER (Figura 3.3) concluído, modelou-se o esquema relacional, o qual se encontra descrito no Anexo B e foi utilizado para implantação do banco de dados.

3.2.2 Implantação do banco de dados

O novo banco de dados foi implantado no sistema gerenciador de bancos *PostgreeSQL* seguindo o esquema relacional do Anexo B. Para isso, utilizou-se uma ferramenta chamada Prisma: uma *Object-Relational Mapping* (ORM) para linguagens de programação *Typescript* e *Node.js*, linguagens que serão utilizadas para implementar o *back-end* da plataforma em sua próxima versão

Uma ORM facilita e automatiza a integração e comunicação entre sistemas de gerenciamento de banco de dados relacionais e linguagens de programação orientadas a objetos (Brunno Kriger, 2023), como é o caso do *Typescript*. Ela desempenha um papel fundamental na persistência de dados, que é a capacidade de armazenar e recuperar informações de forma consistente e eficiente em um banco de dados. O principal diferencial de uma ORM, como Prisma, é que ela permite a abstração dos modelos de dados, ou seja, os desenvolvedores interagem com os dados em um nível mais alto de abstração, tratando os registros do banco de dados como objetos em vez de linhas em tabelas. Além disso, aumenta a produtividade, já que reduz a quantidade de código necessário para manipular dados, visto que muitas tarefas relacionadas a CRUD são tratadas automaticamente pela ORM.

Para criação de cada uma das tabelas do banco no Prisma, implementou-se um *model*, que apresenta todos os atributos e seus tipos, incluindo chaves primárias e restrições de integridade referencial. Após os *models* serem criados, realizou-se uma operação de migração, que transforma

```sql
-- CreateTable
CREATE TABLE "usuarios" (
    "cpf" VARCHAR(12) NOT NULL,
    "email" VARCHAR(100) NOT NULL,
    "senha" VARCHAR(100) NOT NULL,
    "nome" TEXT NOT NULL,
    "sobrenome" TEXT NOT NULL,
    "sexo" CHAR(2) NOT NULL,
    "idade" INTEGER NOT NULL,
    "data_cadastro" TIMESTAMP(3) NOT NULL DEFAULT CURRENT_TIMESTAMP,
    "updated_at" TIMESTAMP(3) NOT NULL,
    "ultimo_acesso" TIMESTAMP(3) NOT NULL,
    "id_curso" INTEGER NOT NULL,
    "id_instituicao" INTEGER NOT NULL,

    CONSTRAINT "usuarios_pkey" PRIMARY KEY ("cpf")
);

-- AddForeignKey
ALTER TABLE "usuarios" ADD CONSTRAINT "usuarios_id_curso_fkey" FOREIGN KEY ("id_curso") REFERENCES "cursos"("id") ON DELETE RESTRICT ON UPDATE RESTRICT;

-- AddForeignKey
ALTER TABLE "usuarios" ADD CONSTRAINT "usuarios_id_instituicao_fkey" FOREIGN KEY ("id_instituicao") REFERENCES "instituicoes"("id") ON DELETE RESTRICT ON UPDATE RESTRICT;
```

Figura 3.5 – *Querys* de criação da tabela Usuário

todos esses *models* em *querys* de criação de tabelas e de restrições, conectando com o *PostgreeSQL* e criando as tabelas no banco de dados desejado.

Um exemplo de *model* que representa a tabela de "Usuário" é representada na Figura 3.4 e suas respectivas *querys* na Figura 3.5. O restante do esquema físico com os *models* representando as respectivas tabelas é apresentado no Anexo C.

4 Experimentação prática

Neste capítulo, são apresentados e analisados os experimentos para validação da estratégia de predição proposta do desempenho dos discentes na TôSabendo, seguindo as arquiteturas apresentadas na Seção 3.1. Desta forma, este capítulo encontra-se assim delineado: a Seção 4.1 apresenta as métricas que foram utilizadas para avaliação dos resultados dos modelos, a Seção 4.2 descreve detalhadamente como foram realizados os experimentos a fim de interpretar os resultados dos modelos e identificar a melhor estratégia de predição e a Seção 4.3 apresenta e avalia os resultados obtidos por meio dos experimentos realizados.

4.1 Métricas de Avaliação

Considerando que a estratégia de predição a ser avaliada utiliza modelos de classificação, conforme detalhado na Seção 3.1, a avaliação baseou-se na utilização da acurácia e precisão como principais métricas. Estas foram empregadas para analisar a performance dos modelos diante de diferentes conjuntos de dados e hiperparâmetros.

Como previamente discutido na Subseção 2.1.4.6, a acurácia, sendo uma medida simples e intuitiva, é amplamente utilizada na avaliação de modelos de predição educacional. Ao representar a proporção de previsões corretas em relação ao total de observações, a acurácia oferece uma visão global do desempenho do modelo. Quanto a precisão, é uma métrica que se concentra na proporção de verdadeiros positivos em relação ao total de previsões positivas do modelo. Sua inclusão na avaliação é justificada pela necessidade de identificar a capacidade do modelo em evitar falsos positivos. Em contextos educacionais, minimizar previsões incorretas é crucial para a eficácia das estratégias de intervenção, tornando a precisão uma métrica complementar valiosa.

Dessa forma, a acurácia e precisão foram escolhidas como métricas principais para a avaliação dos modelos implementados. Essas técnicas combinadas proporcionam uma abordagem abrangente para avaliar a eficácia dos modelos, permitindo intervenções educacionais mais direcionadas e eficazes.

4.2 Descrição experimental

Para a avaliação da estratégia de predição proposta foram testados diferentes conjunto de dados e modelos de predição a fim de analisar e interpretar quais são os melhores dados a serem utilizados em uma determinada predição e quais os modelos que melhor se saem com esses dados. Considerando que a plataforma será usufruída por duas categorias de discentes, novatos e veteranos, os experimentos foram realizados para cada um dessas categorias.

Quanto aos modelos de predição avaliados são os mesmos expostos na Seção 2.1.2. A escolha deles deve-se ao fato de que utilizam dos algoritmos de classificação que mais aparecem em modelos de predição na literatura de previsão de desempenho acadêmico de acordo com (Alyahyan; Düştegör, 2020). Os mesmos foram implementados e avaliados na linguagem de programação *python*, que oferece diversas bibliotecas que apresentam os algoritmos para construção desses modelos já prontos, sendo apenas necessário entender e selecionar os seus hiperparâmetros. As bibliotecas utilizadas foram o ***scikit-learn*** e, especificamente para as redes neurais, o ***tensorflow***. Além disso, há outras bibliotecas que facilitam o *plot* de tabelas e gráficos para interpretação dos resultados, como a ***matplotlib***.

Dessa forma, dois experimentos bases foram realizados com os conjuntos de dados estabelecidos e os modelos. O primeiro experimento realizado foi para selecionar qual modelo teve os melhores resultados com novatos e qual teve para veteranos. Já no segundo experimento, tais melhores modelos passaram por uma seleção de hiperparâmetros para otimizar ainda mais seus desempenhos. Tais experimentos realizados encontram-se apresentados nas próximas Subseções: a Subseção 4.2.1 referente ao teste para determinação dos melhores modelos e a Subseção 4.2.2 referente ao teste dos diferentes hiperparâmetros.

4.2.1 Determinação dos melhores modelos

Para investigar o desempenho ótimo de cada modelo, independentemente da seleção de hiperparâmetros, foram gerados 20 conjuntos de dados distintos, consistindo em 10 conjuntos para novatos e 10 conjuntos para veteranos. Cada conjunto compreendeu 1000 instâncias representativas de diferentes discentes. O modelo que demonstre a superioridade geral seria escolhido para a respectiva categoria de discente e implementado na plataforma TôSabendo para a realização de predições relacionadas ao desempenho acadêmico desses indivíduos.

Em relação a cada um conjunto de dados utilizado, como a plataforma TôSabendo ainda não está em funcionamento e os dados que haviam até o momento são pouco úteis para predição, necessitou-se criar dados sintéticos. Optou-se por gerar esses dados de forma aleatória em vez de criá-los manualmente por diversas razões que contribuem para a eficácia e robustez dos modelos de predição. Algumas dessas razões são:

- a geração aleatória permite abranger uma ampla gama de cenários possíveis, considerando diversas combinações de variáveis e valores. Isso ajuda a garantir que os modelos sejam treinados e testados em situações diversas, tornando-os mais generalizáveis e capazes de lidar com a variabilidade natural dos dados reais;
- a criação de dados manualmente pode introduzir vieses inconscientes e limitações na representação do verdadeiro espectro de dados. A aleatoriedade na geração de dados procura evitar esse viés, garantindo uma representação mais equilibrada e realista das possíveis situações que os modelos podem encontrar, reduzindo, assim, o risco de sobreajuste

(*overfitting*), em que o modelo se ajusta demais a um certo conjunto de dados e, com novos dados, o mesmo obteria péssimos resultados.

Na criação desses conjuntos de dados aleatórios, a seleção dos atributos para a predição do desempenho dos discentes na plataforma TôSabendo foi embasada na Seção 2.1.3. Os atributos escolhidos foram o desempenho acadêmico prévio e dados demográficos. Para os discentes novatos, o desempenho acadêmico prévio incluiu as médias em português e matemática no ensino médio, além da média geral e quantidade de faltas. Em contrapartida, para os discentes veteranos, o desempenho acadêmico prévio foi representado pela média geral e pelo coeficiente. Em ambas as categorias, os dados demográficos contemplaram o gênero e a idade dos discentes. Com base nessas características, foram atribuídos rótulos a cada discente, indicando se seu desempenho foi classificado como bom, médio ou ruim.

Após a criação das instâncias dos discentes juntamente com seus respectivos rótulos, os dados foram submetidos a um processo de pré-processamento visando assegurar qualidade e relevância no aprendizado dos modelos. Conforme abordado na Seção 2.1.4 referente à EDM, é importante observar que não foi necessário realizar uma preparação inicial ou análise estatística dos dados, já que os dados foram criados manualmente e são sintéticos, não sendo necessária a seleção de dados, a limpeza de dados e a criação de novas variáveis.

Quanto ao pré-processamento, a primeira etapa consistiu na normalização da idade dos discentes para o intervalo entre 0 e 1, proporcionando uma escala única e tornando-as comparáveis, facilitando, assim, o treinamento de modelos sensíveis à escala. Em seguida, os rótulos, inicialmente em formato de texto ou categorias, foram convertidos em valores numéricos, possibilitando que algoritmos lidem de maneira mais eficiente com esse tipo de dado. Por fim, realizou-se a padronização das características, removendo a média e dimensionando para a variação unitária. Esse procedimento coloca todas as características em uma escala comum, contribuindo para a consistência e a interpretação efetivas pelos modelos.

A partir dos dados pré-processados, procedeu-se à divisão dos mesmos, alocando 70% para o conjunto de treinamento e 30% para o conjunto de teste em cada um dos 20 conjuntos gerados. Essa abordagem resultou em conjuntos de treinamento e teste totalmente aleatórios e distintos entre si.

Ao avançar para o treinamento dos modelos, visando reforçar a validação e assegurar uma avaliação mais robusta, incorporou-se a técnica de *K-Fold Cross Validation*. Conforme ilustrado na Figura 4.1, essa abordagem envolve a subdivisão do conjunto de dados de treinamento em k subconjuntos (**folds**), com o modelo sendo treinado k vezes. A cada iteração, k-1 *folds* são utilizados como conjunto de treinamento, enquanto o *fold* remanescente funciona como conjunto de validação. Essa prática permite uma avaliação abrangente do desempenho do modelo, atenuando a sensibilidade a variações nos dados de treinamento e validação. Além disso, contribui para a seleção de um modelo mais generalizável e confiável, prevenindo o sobreajuste. Essa

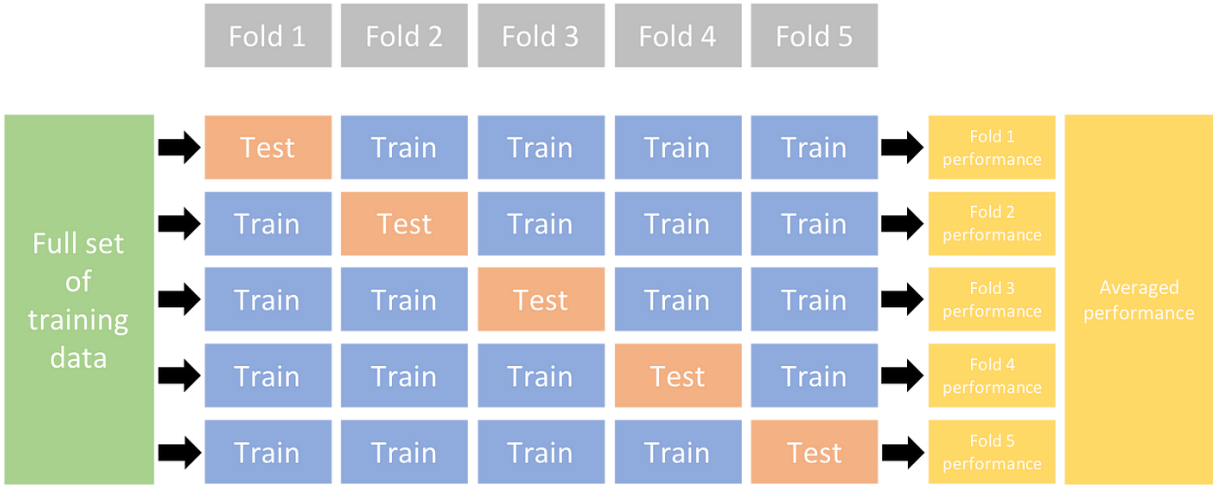

Figura 4.1 – Funcionamento do *K-Fold Cross Validation* (Glenn Jocher, Burhan Q., 2023)

técnica foi aplicada e avaliada em todos os conjuntos de dados.

Assim, após o treinamento e teste dos modelos, foi calculada uma média global do desempenho de cada modelo nos conjuntos de treinamento e teste. Com base na interpretação desses resultados, um modelo foi escolhido para a categoria de discentes novatos, enquanto outro foi selecionado para a categoria de discentes veteranos.

4.2.2 Testagem de hiperparâmetros

Após a seleção dos modelos de predição para novatos e veteranos, realizou-se uma testagem de diferentes hiperparâmetros com objetivo de otimizar os modelos escolhidos e aprimorar seus resultados. Para essa tarefa, criou-se um novo conjunto de dados para novatos e outro para veteranos com as mesmas características e tamanho dos criados para o experimento da Subseção 4.2.1. Com esses conjuntos, realizou um *Grid-search* juntamente com um *K-fold cross-validation*.

Como ilustrado na Figura 4.2, o *Grid-search* é uma técnica que se caracteriza pela busca exaustiva de uma combinação específica de hiperparâmetros dentro de um conjunto previamente definido. Na instância apresentada na figura, essa busca diz respeito a dois hiperparâmetros distintos: o *learning-rate* (taxa de aprendizado) e o *batch-size* (tamanho dos lotes de treinamento) de uma Rede Neural Artificial (RNA). Vale ressaltar que, em alguns casos, um desses parâmetros pode ter uma influência menos significativa que o outro, justificando a importância de avaliar todas as combinações possíveis. De modo geral, esse processo de busca é conduzido em uma grade, em que cada dimensão representa um hiperparâmetro, e os pontos de interseção indicam as diferentes combinações de valores desses parâmetros.

Para simplificar a integração entre o ***Grid-search*** e o ***K-fold Cross Validation***, adotou-se a função ***GridSearchCV*** da biblioteca ***scikit-learn*** em *Python*. Essa escolha justifica-se pelo fato de o ***GridSearchCV*** automatizar tanto o treinamento do modelo quanto a avaliação do desempenho para todas as combinações de hiperparâmetros, simplificando consideravelmente o processo de

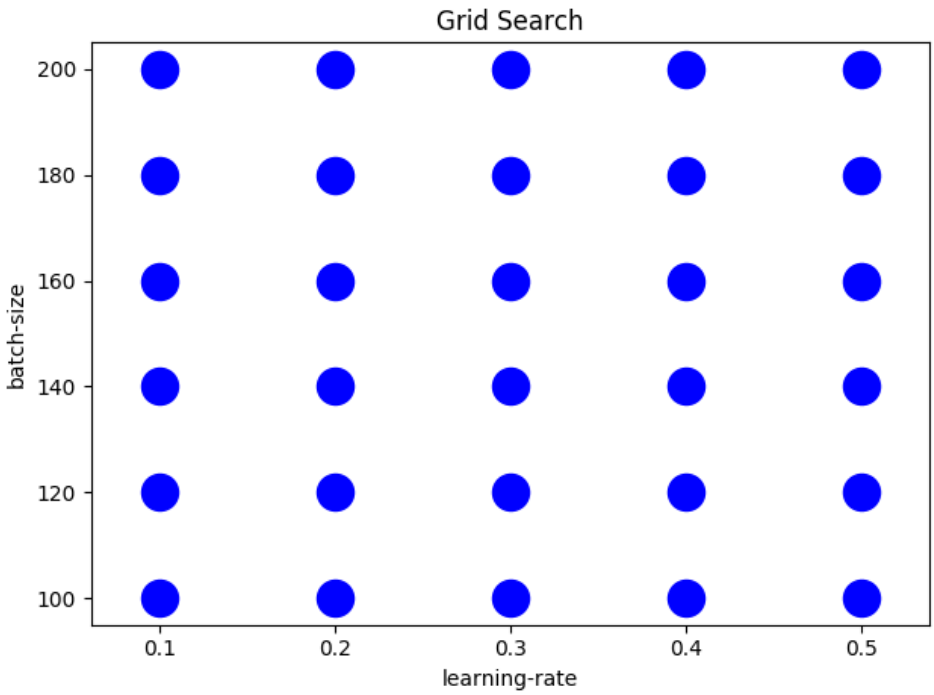

Figura 4.2 – *Grid-search* para combinação de hiperparâmetros de uma RNA

testagem. Além disso, a utilização da validação cruzada contribui para mitigar o sobreajuste, pois avalia o modelo em diferentes conjuntos de dados de validação durante o treinamento. Após a execução do ***GridSearchCV***, a função identifica e retorna o modelo que apresenta a melhor combinação de hiperparâmetros, com base nas métricas de avaliação escolhidas, que foram a acurácia e a precisão. Esse processo sistemático e automatizado assegura a seleção de modelos otimizados e alinhados aos critérios específicos de desempenho estabelecidos.

Assim, para cada um dos modelos com as melhores configurações selecionadas, foram realizados testes utilizando subconjuntos isolados (subconjunto de teste) para avaliar a capacidade dos modelos em lidar com novos dados. Em situações em que os resultados não atenderam às expectativas, foi conduzido um novo teste com o *GridSearchCV*, explorando diferentes valores de hiperparâmetros que não haviam sido previamente considerados na experimentação inicial. Após a execução meticulosa do *GridSearchCV*, a análise e interpretação repetida dos resultados e a constatação de ausência de melhorias significativas, os modelos treinados e testados, que proporcionaram os resultados mais satisfatórios, serão empregados para a predição do desempenho dos discentes na plataforma TôSabendo. Esse processo rigoroso de seleção garante a utilização de modelos robustos, ajustados às nuances dos dados, e confere confiabilidade nas predições do desempenho acadêmico.

4.3 Análise dos Resultados Obtidos

Nesta seção, são apresentados e analisados os resultados obtidos por meio da experimentação prática realizada, envolvendo a descrição experimental apresentada na Seção 4.2. A Subseção 4.3.1 descreve os resultados do primeiro experimento quanto aos diferentes conjuntos e a Subseção 4.3.2 descreve os resultados do segundo experimento quanto a testagem de diversos hiperparâmetros nos modelos selecionados no primeiro experimento.

4.3.1 Determinação dos melhores modelos

No contexto do primeiro experimento, que abrange os quatro modelos de predição implementados, os resultados de acurácia e precisão média nos conjuntos de treinamento e teste, provenientes da avaliação em 10 conjuntos distintos, são apresentados nas Tabelas 4.1 e 4.2, para discentes novatos e veteranos, respectivamente. Os melhores resultados dentre todos os modelos foram sublinhados de **verde**. Vale ressaltar que, neste experimento, a seleção dos hiperparâmetros não constituiu a métrica primordial para a identificação do modelo ótimo.

Tabela 4.1 – Desempenho Médio dos Modelos (%) para discentes novatos

Modelo	**Acurácia**		**Precisão**	
	Treino	**Teste**	**Treino**	**Teste**
Árvore de Decisão	**98.65**	**99.03**	**98.68**	**99.04**
KNN	95.21	95.83	95.28	95.89
Naive Bayes	86.14	85.27	86.42	85.39
Redes Neurais	93.64	94.17	93.85	94.33

Tabela 4.2 – Desempenho Médio dos Modelos (%) para discentes veteranos

Modelo	**Acurácia**		**Precisão**	
	Treino	**Teste**	**Treino**	**Teste**
Árvore de Decisão	**99.70**	**99.77**	**99.71**	**99.77**
KNN	96.66	96.97	96.60	96.99
Naive Bayes	94.85	94.10	95.67	94.96
Redes Neurais	97.35	97.87	97.35	97.97

Inicialmente, observou-se a ausência de sobreajuste em todos os modelos, uma vez que foram treinados utilizando validação cruzada nos 10 conjuntos de dados tanto para discentes novatos quanto para veteranos. Dessa forma, ao serem avaliados nos conjuntos de teste, os modelos mantiveram um desempenho consistente, indicando uma capacidade robusta de generalização. Essa constatação assegura que os modelos não apenas se adaptaram de maneira excessiva aos dados de treinamento, mas também conseguiram generalizar eficazmente para novos conjuntos de dados.

Além disso, diante da eficaz generalização dos modelos nas duas categorias de discentes, destaca-se a Árvore de Decisão, evidenciando a mais alta acurácia e precisão em ambos os conjuntos de treino e teste. O KNN também apresenta um desempenho robusto, seguido pelas Redes Neurais. No entanto, o Naive Bayes demonstra um desempenho ligeiramente inferior em comparação com os outros modelos. Esses resultados sugerem que, considerando que os dados utilizados para predição são mais simples e criados artificialmente, a Árvore de Decisão, ao capturar padrões mais simples, obteve os melhores resultados. Apesar de as redes neurais demonstrarem robustez na captação de padrões complexos, o problema em questão não exibe padrões extremamente complexos ou não lineares. Portanto, modelos mais simples, como a Árvore de Decisão, mostram-se suficientes para a tarefa em questão. Essa análise ressalta a importância de selecionar modelos adequados à natureza dos dados e padrões subjacentes ao problema em foco.

Portanto, frente aos resultados apresentados e da análise efetuada, conclui-se que a previsão do desempenho de discentes, tanto para novatos quanto para veteranos, na plataforma TôSabendo, revela maior eficácia ao empregar modelos mais simplificados, a exemplo das Árvores de Decisão. A implementação da validação cruzada, como medida preventiva contra o sobreajuste, evidenciou a robustez dos modelos, assegurando a sua habilidade de generalização para novos conjuntos de dados.

4.3.2 Testagem de hiperparâmetros

Com a escolha da Árvore Decisão sendo o principal modelo a partir da execução do primeiro experimento, tanto para discentes novatos quanto para veteranos, restou a tarefa de identificar os hiperparâmetros ideais que possam aprimorar ainda mais os resultados da predição. A intenção é empregar esses modelos, devidamente treinados e testados, futuramente e pela primeira vez na plataforma TôSabendo.

Conforme mencionado na Subseção 4.2.2, a avaliação foi conduzida por meio do método *grid_search_cv*, que engloba a exploração de diversos hiperparâmetros em conjunto com a técnica de *k-fold cross-validation*. Nos dois modelos que empregam árvores de decisão, os hiperparâmetros submetidos a teste foram:

- *criterion*: define a função para medir a qualidade da divisão em um nó. Os valores testados foram “gini” e “entropy”. “Gini” utiliza o critério de impureza Gini, enquanto “entropy” usa a entropia. A escolha entre eles afeta como a árvore decide onde dividir os nós.
- *splitter*: define a estratégia usada para escolher a divisão em cada nó. Os valores testados foram “best” e “random”. “Best” escolhe a melhor divisão, enquanto “random” escolhe uma divisão aleatória. Isso influencia a aleatoriedade na construção da árvore.

- *max_depth*: controla a profundidade máxima da árvore, evitando o *overfitting*. Os valores testados foram *None*, 10, 20, 30.
- *min_samples_split*: define o número mínimo de amostras necessárias para que um nó seja dividido em dois, ajudando a controlar a complexidade da árvore. Os valores testados foram 2, 4, 5, 10.
- *min_samples_leaf*: define o número mínimo de amostras em uma folha, influenciando a granularidade da árvore. Os valores testados foram 1, 2, 4.

Após os testes, a melhor configuração para o modelo para discentes e para veteranos é apresentada, respectivamente, nas Tabelas 4.3 e 4.4.

Tabela 4.3 – Melhores Hiperparâmetros na árvore de decisão para novatos

Hiperparâmetro	**Melhor Valor**
Criterion	entropy
Splitter	best
Max Depth	None
Min Samples Split	1
Min Samples Leaf	2

Tabela 4.4 – Melhores Hiperparâmetros na árvore de decisão para veteranos

Hiperparâmetro	**Melhor Valor**
Criterion	gini
Splitter	best
Max Depth	None
Min Samples Split	1
Min Samples Leaf	10

Com a seleção desses hiperparâmetros, os resultados referentes à acurácia e à precisão nos conjuntos de treinamento e teste para novatos e veteranos encontram-se nas Tabelas 4.5 e 4.6, respectivamente. Os dados apresentados indicam um desempenho aparentemente excepcional para os modelos de classificação aplicados aos grupos de Novatos e Veteranos. Ambos os conjuntos de treinamento e teste demonstraram acurácia e precisão de 100% ou quase isso, sugerindo uma notável capacidade de aprendizado e generalização nos modelos. Contudo, tais resultados podem suscitar preocupações acerca da possibilidade de sobreajuste nos dados de treinamento, mesmo com a implementação do *k-fold cross-validation*.

Tabela 4.5 – Resultados de Treinamento e Teste (%) para Novatos

Métrica	**Treinamento**	**Teste**
Acurácia	100.0	99.33
Precisão	100.0	99.33

Tabela 4.6 – Resultados de Treinamento e Teste (%) para Veteranos

Métrica	**Treinamento**	**Teste**
Acurácia	100.0	100.0
Precisão	100.0	100.0

Frente ao exposto, é crucial ponderar que o desempenho dos modelos pode variar dependendo da natureza dos dados, do tamanho da amostra e das características específicas dos discentes para predição, sobretudo ao considerar dados sintéticos. Nesse contexto, torna-se necessário conduzir futuros testes para avaliar se a Árvore de Decisão permanecerá como a escolha mais acertada para as duas categorias de discentes. A utilização de dados reais dos discentes que utilizarão a plataforma pode potencialmente favorecer as redes neurais, capazes de identificar padrões de dados mais complexos, resultando em desempenho superior. No entanto, diante esses resultados iniciais, as árvores de decisão avaliadas e interpretadas tanto para discentes novatos e veteranos serão as empregadas para a previsão do desempenho dos discentes na futura plataforma TôSabendo funcional.

5 Conclusões

Neste capítulo, são apresentadas as conclusões sobre o trabalho desenvolvido (vide Seção 5.1) e as perspectivas de trabalho futuro (vide Seção 5.2).

5.1 Conclusão

Como apresentado, este trabalho propôs, desenvolveu e validou uma estratégia para predição de desempenho de discentes na plataforma de ensino *gamificada* denominada TôSabendo, considerando até então a implementação da plataforma realizada por (França *et al.*, 2021) e (Ferreira, 2022). Por meio de tal estratégia, será possível predizer comportamentos de discentes na plataforma para que professores forneçam aos mesmos algum tipo de intervenção com recursos próprios da TôSabendo e/ou recursos didáticos em sala de aula.

A fim de avaliar essa estratégia, proposta na Seção 3.1 por meio de arquiteturas de funcionamento e posteriormente desenvolvida, foram realizados experimentos iniciais para avaliar o melhor modelo de predição tanto para os discentes novatos quanto para os veteranos a partir de distintos conjuntos de dados. Com esses modelos de predição selecionados, em seguida, foi realizada uma testagem de hiperaparâmetros para identificar quais os melhores para predição de desempenho das duas categorias de discentes na plataforma TôSabendo.

Diante dos resultados obtidos no primeiro experimento, torna-se evidente a eficácia dos modelos mais simplificados, em particular, a Árvore de Decisão. A análise abrangeu métricas como acurácia e precisão nos conjuntos de treinamento e teste, revelando que esses modelos conseguiram manter um desempenho consistente, indicando uma capacidade robusta de generalização. O destaque para a Árvore de Decisão, que apresentou a mais alta acurácia e precisão em ambos os conjuntos de treino e teste para as duas categorias de discentes, sugere sua adequação para a tarefa em questão. Ao capturar padrões mais simples nos dados de predição, a Árvore de Decisão demonstrou-se superior, especialmente quando comparada a modelos mais complexos, como Redes Neurais. Considerando ainda os trabalhos relacionados da Seção 2.2, observa-se que os resultados foram distintos deste experimento, visto que as redes neurais em todos os trabalhos se sobressaíram como os melhores modelos. Isso demonstrou que a previsão do desempenho de discentes na plataforma TôSabendo revela maior eficácia ao empregar modelos mais simplificados, especialmente quando confrontados com dados menos complexos e padrões mais simples. Essa análise reforça a importância de selecionar modelos adequados à natureza dos dados e a padrões subjacentes ao problema em foco, enfatizando a relevância da abordagem adotada nos experimentos.

Quanto ao segundo experimento, dedicado à busca e à escolha dos melhores hiperparâme-

tros para as Árvores de Decisão para predição de desempenho de discentes novatos e veteranos, o mesmo proporcionou um ajuste refinado dos modelos, culminando em resultados notáveis nos conjuntos de treinamento e teste. Os valores de acurácia e precisão atingiram 100% ou valores muito próximos, indicando uma capacidade bem significativa de aprendizado e generalização para as duas categorias de discentes. No entanto, a ressalva sobre a possibilidade de sobreajuste nos dados de treinamento destaca a importância contínua da avaliação e adaptação dos modelos, especialmente ao considerar a transição para dados reais dos usuários da plataforma.

Por meio desses estudos e experimentos, alcançou-se alguns dos objetivos específicos estabelecidos para este trabalho. O primeiro desses objetivos consistia no entendimento aprofundado do processo de EDM (Seção 2.1.4) em uma plataforma de ensino *online* que utiliza informações dos discentes. Ao longo da pesquisa, foram mapeadas e compreendidas detalhadamente as nuances desse processo, identificando as etapas críticas e os pontos de intervenção necessários para otimizar a predição do desempenho acadêmico.

Além disso, explorou-se a possibilidade de aplicação da predição do desempenho dos discentes na plataforma para distintas disciplinas e cursos de Instituições de Ensino Superior (IESs) para intervenção antecipada dos educadores. Essa capacidade de prever o desempenho dos estudantes em diversas áreas acadêmicas torna a plataforma TôSabendo uma ferramenta valiosa para complementar o ensino, oferecendo *insights* úteis para melhorar a qualidade da educação em diferentes cursos e disciplinas.

Dessa forma, apesar desses resultados satisfatórios, é crucial reconhecer que o desempenho dos modelos pode ser influenciado por diversos fatores, incluindo a natureza dos dados, o tamanho da amostra e as características específicas dos discentes. Assim, enfatiza-se a necessidade de manter uma abordagem flexível e adaptativa, considerando que a escolha ideal dos modelos pode variar em diferentes cenários. Portanto, a continuidade da avaliação e potenciais ajustes são essenciais para garantir a eficácia dos modelos na predição do desempenho dos discentes na futura plataforma TôSabendo funcional.

5.2 Trabalhos Futuros

Nesta seção, são apresentadas algumas perspectivas de trabalho futuro. Desta forma, pretende-se:

1. tornar a plataforma TôSabendo funcional novamente, envolvendo seu *back-end* e *front-end*;
2. incorporar os modelos de predição a plataforma TôSabendo, criando uma aplicação que conecte os modelos treinados ao *back-end* para gerar respostas aos professores e alunos quanto a predição. Um exemplo de uma arquitetura de funcionamento referente a essa incorporação é apresentado no Apêndice A;

3. criar um sistema de retreinamento automático dos modelos de predição a partir dos novos dados dos discentes;

4. realizar novos experimentos considerando a testagem de hiperparâmetros nos modelos para tuná-los antes de selecionar o melhor;

5. realizar novos experimentos a partir de dados reais dos discentes, até mesmo com atributos próprios da TôSabendo, como quantidade de *quizzes* resolvidos e suas pontuações, e quantidade de acessos a plataforma;

6. experimentar como os dados sair-se-ão com modelos que utilizem de algoritmos de regressão para predizer a pontuação que os discentes terão na plataforma e não ter apenas categorias como resultados da predição;

7. aplicar testes de significância estatísticas nos resultados apresentados para verificar um possível sobreajuste gerado pela criação de dados sintéticos.

Referências

ALA'RAJ, M. *et al.* A deep learning model for behavioural credit scoring in banks. **Neural Computing and Applications**, Springer, p. 1–28, 2022.

ALI, Z. M. *et al.* The application of data mining for predicting academic performance using k-means clustering and naïve bayes classification. **International Journal of Psychosocial Rehabilitation**, v. 24, n. 03, p. 2143–2151, 2020.

ALQURASHI, E. Predicting student satisfaction and perceived learning within online learning environments. **Distance education**, Taylor & Francis, v. 40, n. 1, p. 133–148, 2019.

ALTURKI, S.; HULPUș, I.; STUCKENSCHMIDT, H. Predicting academic outcomes: A survey from 2007 till 2018. **Technology, Knowledge and Learning**, Springer, p. 1–33, 2022.

ALYAHYAN, E.; DÜŞTEGÖR, D. Predicting academic success in higher education: literature review and best practices. **International Journal of Educational Technology in Higher Education**, Springer, v. 17, p. 1–21, 2020.

AZANK, F. **Modelos de Predição | KNN**. 2019. Acessado em 1 de junho de 2023. Disponível em: <https://medium.com/turing-talks/turing-talks-10-introdu%C3%A7%C3%A3o-%C3%A0-predi%C3%A7%C3%A3o-a75cd61c268d>.

BAASHAR, Y. *et al.* Toward predicting student's academic performance using artificial neural networks (anns). **Applied Sciences**, MDPI, v. 12, n. 3, p. 1289, 2022.

BERRY, M. J.; LINOFF, G. S. **Data mining techniques: for marketing, sales, and customer relationship management**. Washington, DC: John Wiley & Sons, 2004.

BRAMER, M.; BRAMER, M. Data for data mining. **Principles of data mining**, Springer, p. 9–19, 2016.

BRAVO-AGAPITO, J.; ROMERO, S. J.; PAMPLONA, S. Early prediction of undergraduate student's academic performance in completely online learning: A five-year study. **Computers in Human Behavior**, Elsevier, v. 115, p. 106595, 2021.

BRDESEE, H. S. *et al.* Predictive model using a machine learning approach for enhancing the retention rate of students at-risk. **International Journal on Semantic Web and Information Systems (IJSWIS)**, IGI Global, v. 18, n. 1, p. 1–21, 2022.

BREIMAN, L. **Classification and regression trees**. New York, NY, USA: Routledge, 2017.

BROWN, M. L.; KROS, J. F. Data mining and the impact of missing data. **Industrial Management & Data Systems**, MCB UP Ltd, v. 103, n. 8, p. 611–621, 2003.

Brunno Kriger. **ORM (OBJECT RELATIONAL MAPPER) – SAIBA O QUE É E IMPORTÂNCIA NA PROGRAMAÇÃO**. 2023. Acessado em 14 de agosto de 2023. Disponível em: <{https://www.devmedia.com.br/orm-object-relational-mapper/1905}.>

CARRARO, L. M. **Modelos de Predição | Introdução à Predição**. 2019. Acessado em 01 de junho de 2023. Disponível em: <https://medium.com/turing-talks/turing-talks-10-introdu%C3%A7%C3%A3o-%C3%A0-predi%C3%A7%C3%A3o-a75cd61c268d>.

CESTNIK, B.; KONONENKO, I.; BRATKO, I. Assistant 86: A knowledge-elicitation tool for sophisticated users. progress in machine learning. In: **Proc. of EWSL**. Ljubijana, Yugoslavia: Sigma Press, 1987. v. 87.

CHANDRASHEKAR, G.; SAHIN, F. A survey on feature selection methods. **Computers & Electrical Engineering**, Elsevier, v. 40, n. 1, p. 16–28, 2014.

CHATFIELD, C. Model uncertainty, data mining and statistical inference. **Journal of the Royal Statistical Society Series A: Statistics in Society**, Oxford University Press, v. 158, n. 3, p. 419–444, 1995.

CHAWLA, N. V. *et al.* Smote: synthetic minority over-sampling technique. **Journal of artificial intelligence research**, v. 16, p. 321–357, 2002.

CROWDFLOWER. **Data Science Report**. 2016. Acessado em 30 de junho de 2023. Disponível em: <https://visit.figure-eight.com/rs/416-ZBE-142/images/CrowdFlower_DataScienceReport_2016.pdf>.

CUNNINGHAM, P.; DELANY, S. J. k-nearest neighbour classifiers-a tutorial. **ACM computing surveys (CSUR)**, ACM New York, NY, USA, v. 54, n. 6, p. 1–25, 2021.

Diego Nogare. **Performance de Machine Learning – Matriz de Confusão**. 2020. Acessado em 11 de julho de 2023. Disponível em: <{https://diegonogare.net/2020/04/performance-de-machine-learning-matriz-de-confusao}.>

DUTT, A.; ISMAIL, M. A.; HERAWAN, T. A systematic review on educational data mining. **Ieee Access**, Ieee, v. 5, p. 15991–16005, 2017.

EINSTEIN, A. Cosmic religion: with other opinions and aphorism. **Covici-Freide**, New York (EUA), 1931.

FACELI, K. *et al.* **Inteligência artificial: uma abordagem de aprendizado de máquina**. Rio de Janeiro, RJ: LTC, 2021.

FEELDERS, A.; DANIELS, H.; HOLSHEIMER, M. Methodological and practical aspects of data mining. **Information & Management**, Elsevier, v. 37, n. 5, p. 271–281, 2000.

FERREIRA, C. O. Desenvolvimento de uma estratégia de machine learning para aprimoramento da plataforma tôsabendo. In: UFOP. Ouro Preto, MG, 2022.

FIELDING, R. T. **Architectural styles and the design of network-based software architectures**. Irvine, CA: University of California, Irvine, 2000.

FILIPOUSKI, A.; MARCHI, D.; SIMÕES, L. J. Língua portuguesa e literatura. **RIO GRANDE**, 2009.

FRANÇA, T. F. *et al.* Tôsabendo: A platform to create engaging teaching and learning experiences. In: IEEE. **2021 XVI Latin American Conference on Learning Technologies (LACLO)**. Ouro Preto, MG, 2021. p. 275–281.

GAMA, J. Árvores de decisão. **Palestra ministrada no Núcleo da Ciência de Computação da Universidade do Porto, Porto**, 2002.

GARCÍA, S.; LUENGO, J.; HERRERA, F. **Data preprocessing in data mining**. New York City, NY: Springer, 2015. v. 72.

GARG, R. Predicting student performance of different regions of punjab using classification techniques. **International Journal of Advanced Research in Computer Science**, v. 9, n. 1, p. 236–241, 2018.

GERDS, T. A.; CAI, T.; SCHUMACHER, M. The performance of risk prediction models. **Biometrical Journal: Journal of Mathematical Methods in Biosciences**, Wiley Online Library, v. 50, n. 4, p. 457–479, 2008.

GIARDINETTO, J. R. B.; MARIANI, J. M. Jogos, brinquedos e brincadeiras: O processo ensino-aprendizagem da matemática na educação infantil. **MATEMÁTICA E EDUCAÇÃO INFANTIL**, UNIVERSIDADE ESTADUAL PAULISTA" JÚLIO DE MESQUITA FILHO, 2005.

GIROTO, C. R. M.; POKER, R. B.; OMOTE, S. **As tecnologias nas práticas pedagógicas inclusivas**. São Paulo, SP: Editora Oficina Universitária, 2012.

Glenn Jocher, Burhan Q. **K-Fold Cross Validation with Ultralytics**. 2023. Acessado em 16 de dezembro de 2023. Disponível em: <{https://docs.ultralytics.com/guides/kfold-cross-validation/#conclusio}.>

GOMEDE, E. *et al.* Application of computational intelligence to improve education in smart cities. **Sensors**, MDPI, v. 18, n. 1, p. 267, 2018.

GOTTFRIED, M.; KIRKSEY, J. J.; FLETCHER, T. L. Do high school students with a same-race teacher attend class more often? **Educational evaluation and policy analysis**, SAGE Publications Sage CA: Los Angeles, CA, v. 44, n. 1, p. 149–169, 2022.

HAMOUD, A.; HASHIM, A. S.; AWADH, W. A. Predicting student performance in higher education institutions using decision tree analysis. **International Journal of Interactive Multimedia and Artificial Intelligence**, v. 5, p. 26–31, 2018.

HAN, J.; PEI, J.; TONG, H. **Data mining: concepts and techniques**. Waltham. MA: Morgan kaufmann, 2022.

HASIB, K. M. *et al.* A machine learning and explainable ai approach for predicting secondary school student performance. In: IEEE. **2022 IEEE 12th Annual Computing and Communication Workshop and Conference (CCWC)**. Las Vegas, NV, 2022. p. 0399–0405.

HASTIE, T. *et al.* **The elements of statistical learning: data mining, inference, and prediction**. New York, NY, USA: Springer, 2009. v. 2.

JAVEED, A. *et al.* Machine learning for dementia prediction: A systematic review and future research directions. **Journal of medical systems**, Springer, v. 47, n. 1, p. 17, 2023.

JI, L.; ZHANG, X.; ZHANG, L. Research on the algorithm of education data mining based on big data. In: CSEI. **2020 IEEE 2nd International Conference on Computer Science and Educational Informatization (CSEI)**. Zhengzhou, China, 2020. v. 9, n. 3, p. 344–350.

JOYCE, J. L.; HARRIS, L. Artificial intelligence (ai) and education. **Focus, Congressional Research service, August. Retrived from www. crs. gov**, 2018.

KARIMI, H.; HUANG, J.; DERR, T. A deep model for predicting online course performance. **Association for the Advancement of Artificial Intelligence**, 2020.

KHOSHGOFTAAR, T. M.; GOLAWALA, M.; HULSE, J. V. An empirical study of learning from imbalanced data using random forest. In: IEEE. **19th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2007)**. Patras, Greece, 2007. v. 2, p. 310–317.

KIMURA, H. **Por que o GraphQL vem tomando espaço do REST?** 2021. Acessado em 30 de junho de 2023. Disponível em: <https://prensa.li/prensa/por-que-o-graphql-vem-tomando-espaco-do-rest/>.

KOHAVI, R.; JOHN, G. H. Wrappers for feature subset selection. **Artificial intelligence**, Elsevier, v. 97, n. 1-2, p. 273–324, 1997.

KROGH, A. What are artificial neural networks? **Nature biotechnology**, Nature Publishing Group US New York, v. 26, n. 2, p. 195–197, 2008.

KUBAT, M. Neural networks: a comprehensive foundation by simon haykin, macmillan, 1994, isbn 0-02-352781-7. **The Knowledge Engineering Review**, Cambridge University Press, Cambridge, v. 13, n. 4, p. 409–412, 1999.

KUHN, M.; JOHNSON, K. *et al.* **Applied predictive modeling**. New York, NY: Springer, 2013. v. 26.

KUMAR, S. A. *et al.* Efficiency of decision trees in predicting student's academic performance. Citeseer, 2011.

KUZILEK, J.; HLOSTA, M.; ZDRAHAL, Z. Open university learning analytics dataset. **Scientific data**, Nature Publishing Group, v. 4, n. 1, p. 1–8, 2017.

LANTZ, B. **Machine learning with R: expert techniques for predictive modeling**. Birmingham, England: Packt publishing ltd, 2019.

LIÑÁN, L. C.; PÉREZ, Á. A. J. Educational data mining and learning analytics: differences, similarities, and time evolution. **RUSC. Universities and Knowledge Society Journal**, v. 12, n. 3, p. 98–112, 2015.

LIU, H.; MOTODA, H. **Feature selection for knowledge discovery and data mining**. New York, NY, USA: Springer Science & Business Media, 2012. v. 454.

M, A.; RAHMAN, A. M. J. M. Z. A review on data mining techniques and factors used in educational data mining to predict student amelioration. In: **2016 International Conference on Data Mining and Advanced Computing (SAPIENCE)**. Ernakulam, India: IEEE, 2016. p. 122–133.

MARTINS, M. P. *et al.* A data mining approach for predicting academic success–a case study. In: SPRINGER. **Information Technology and Systems: Proceedings of ICITS 2019**. New York, NY, USA: Springer, 2019. p. 45–56.

MCCARTHY, R. V. *et al.* **Applying predictive analytics**. Hamden, CT: Springer, 2022.

MITTLEMAN, J. Intersecting the academic gender gap: The education of lesbian, gay, and bisexual america. **American Sociological Review**, SAGE Publications Sage CA: Los Angeles, CA, v. 87, n. 2, p. 303–335, 2022.

MONARD, M. C.; BARANAUSKAS, J. A. Indução de regras e árvores de decisão. **Sistemas Inteligentes-fundamentos e aplicações**, sn, v. 1, p. 115–139, 2003.

MOSCOSO-ZEA, O.; LUJÁN-MORA, S. *et al.* Datawarehouse design for educational data mining. In: IEEE. **2016 15th International Conference on Information Technology Based Higher Education and Training (ITHET)**. Istanbul, Turkey, 2016. p. 1–6.

MURPHY, K. P. **Machine learning: a probabilistic perspective**. Cambridge, Massachusetts: MIT press, 2012.

NISBET, R.; ELDER, J.; MINER, G. D. **Handbook of statistical analysis and data mining applications**. Santa Barbara, Goleta, California: Academic press, 2009.

NIYOGISUBIZO, J. *et al.* Predicting student's dropout in university classes using two-layer ensemble machine learning approach: A novel stacked generalization. **Computers and Education: Artificial Intelligence**, Elsevier, v. 3, p. 100066, 2022.

OSBORNE, J. Notes on the use of data transformations. **Practical assessment, research, and evaluation**, v. 8, n. 1, p. 6, 2002.

PALLATHADKA, H. *et al.* Classification and prediction of student performance data using various machine learning algorithms. **Materials today: proceedings**, Elsevier, v. 80, p. 3782–3785, 2023.

PENG, Y. *et al.* A descriptive framework for the field of data mining and knowledge discovery. **International Journal of Information Technology & Decision Making**, World Scientific, v. 7, n. 04, p. 639–682, 2008.

POLIDORI, M. M.; MARINHO-ARAUJO, C. M.; BARREYRO, G. B. **SINAES: perspectivas e desafios na avaliação da educação superior brasileira**. Campinas, SP: SciELO Brasil, 2006. 425–436 p. Acessado em 03 de julho de 2023.

PRENSKY, M. Digital natives, digital immigrants part 2: Do they really think differently? **On the horizon**, MCB UP Ltd, v. 9, n. 6, p. 1–6, 2001.

PYLE, D. **Data preparation for data mining**. San Francisco, CA: morgan kaufmann, 1999.

QAZI, N.; RAZA, K. Effect of feature selection, smote and under sampling on class imbalance classification. In: IEEE. **2012 UKSim 14th International Conference on Computer Modelling and Simulation**. Cambridge, UK, 2012. p. 145–150.

QIU, F. *et al.* Predicting students' performance in e-learning using learning process and behaviour data. **Scientific Reports**, Nature Publishing Group UK London, v. 12, n. 1, p. 453, 2022.

QUINLAN, J. R. Induction of decision trees. **Machine learning**, Springer, v. 1, p. 81–106, 1986.

QUINLAN, J. R. *et al.* Bagging, boosting, and c4. 5. In: **Aaai/Iaai, vol. 1**. Sydney, Australia: AAAI, 1996. v. 96, p. 725–730.

RASCHKA, S.; MIRJALILI, V. **Python machine learning second edition**. Birmingham, England: Packt Publishing.–, 2017.

RAWAL, A.; LAL, B. Predictive model for admission uncertainty in high education using naïve bayes classifier. **Journal of Indian Business Research**, Emerald Publishing Limited, v. 15, n. 2, p. 262–277, 2023.

ROMERO, C.; VENTURA, S. Educational data mining: A survey from 1995 to 2005. **Expert systems with applications**, Elsevier, v. 33, n. 1, p. 135–146, 2007.

ROMERO, C.; VENTURA, S. Data mining in education. **Wiley Interdisciplinary Reviews: Data mining and knowledge discovery**, Wiley Online Library, v. 3, n. 1, p. 12–27, 2013.

ROMERO, C.; VENTURA, S. Educational data mining and learning analytics: An updated survey. **Wiley interdisciplinary reviews: Data mining and knowledge discovery**, Wiley Online Library, v. 10, n. 3, p. e1355, 2020.

RUANO, M. *et al.* A systematic approach for fine-tuning of fuzzy controllers applied to wwtps. **Environmental Modelling & Software**, Elsevier, v. 25, n. 5, p. 670–676, 2010.

SAMUEL, A. L. Some studies in machine learning using the game of checkers. **IBM Journal of research and development**, IBM, v. 3, n. 3, p. 210–229, 1959.

SANTOS, H. G. d. **Comparação da performance de algoritmos de machine learning para a análise preditiva em saúde pública e medicina**. Tese (Doutorado) — Universidade de São Paulo, São Paulo, SP, 2018.

SARALA, V.; KRISHNAIAH, J. Empirical study of data mining techniques in education system. **International Journal of Advances in Computer Science and Technology (IJACST)**, Citeseer, v. 4, n. 1, p. 15–21, 2015.

SEIFERT, J. W. Data mining: An overview. **National security issues**, p. 201–217, 2004.

SEIXAS, L. da R. *et al.* Gamificaçao como estratégia no engajamento de estudantes do ensino fundamental. In: **Anais do Simpósio Brasileiro de Informática na Educação**. Recife, PE: CBIE, 2014. v. 25, n. 1, p. 559.

SEKEROGLU, B.; DIMILILER, K.; TUNCAL, K. Student performance prediction and classification using machine learning algorithms. In: **Proceedings of the 2019 8th International Conference on Educational and Information Technology**. New York, NY, USA: ICEIT, 2019. p. 7–11.

SOUSA, R. P. d. *et al.* **Tecnologias digitais na educação**. Campina Grande, PB: Eduepb, 2011.

TOMASEVIC, N.; GVOZDENOVIC, N.; VRANES, S. An overview and comparison of supervised data mining techniques for student exam performance prediction. **Computers & education**, Elsevier, v. 143, p. 103676, 2020.

VALENTINI, C. B.; SOARES, E. M. d. S. Aprendizagem em ambientes virtuais-compartilhando ideias e construindo cenários. Educs, 2005.

VISALAKSHI, S.; RADHA, V. A literature review of feature selection techniques and applications: Review of feature selection in data mining. In: IEEE. **2014 IEEE international conference on computational intelligence and computing research**. Coimbatore, India, 2014. p. 1–6.

WANG, A. I.; TAHIR, R. The effect of using kahoot! for learning–a literature review. **Computers & Education**, Elsevier, v. 149, p. 103818, 2020.

WANG, Y. *et al.* Academic performance under covid-19: The role of online learning readiness and emotional competence. **Current psychology**, Springer, p. 1–14, 2022.

WIENER, A.; CAMPOS, A. de. Kolligo: gamificação na educação para experiência de aprendizagem mais engajadoras. In: **Anais dos Workshops do Congresso Brasileiro de Informática na Educação**. Florianopolis, SC: CBIE, 2019. v. 8, n. 1, p. 1180.

Wikipedia. **Receiver operating characteristic**. 2003. https://en.wikipedia.org/wiki/Receiver_operating_chara Acessado em 11 de julho de 2023.

XU, H. *et al.* Novel key indicators selection method of financial fraud prediction model based on machine learning hybrid mode. **Mobile Information Systems**, Hindawi, v. 2022, 2022.

YAĞCI, M. Educational data mining: prediction of students' academic performance using machine learning algorithms. **Smart Learning Environments**, Springer, v. 9, n. 1, p. 11, 2022.

YORK, T. T.; GIBSON, C.; RANKIN, S. Defining and measuring academic success. **Practical assessment, research, and evaluation**, v. 20, n. 1, p. 5, 2015.

ZHOU, W. *et al.* Rest api design patterns for sdn northbound api. In: IEEE. **2014 28th international conference on advanced information networking and applications workshops**. Victoria, BC, Canada, 2014. p. 358–365.

ZHOU, Z.-H. **Machine learning**. Nanjing, Jiangsu, China: Springer Nature, 2021.

Apêndices

APÊNDICE A – Incorporação de modelos de predição à plataforma TôSabendo

Com a escolha dos modelos de predição a serem utilizados na plataforma TôSabendo, torna-se necessário incorporá-los à plataforma, sendo um voltado para os discentes novatos e outro para os discentes veteranos. Para isso, inicialmente é necessário remodelar o banco de dados e o *back-end* da plataforma por meio da adição dos dados essenciais aplicados nos melhores modelos. A Figura A.1 ilustra a arquitetura de funcionamento da Plataforma TôSabendo considerando modelos de predição incorporados. O fluxo em azul indica escrita de dados no banco, o fluxo em rosa indica uma consulta ao banco e o fluxo em preto indica um processo isolado de operações no banco. De acordo com a Figura A.1, os passos que demonstram o futuro funcionamento da plataforma com os modelos de predição incorporados são:

- Passo 1: consiste no cadastro de discentes quando entram na plataforma TôSabendo, onde há o fornecimento de valores de atributos que já estavam presentes no antigo banco de dados (vide Seção 2.1.1.2) como, por exemplo, *e-mail*, senha, sexo, nome, e fornecimento de dados adicionais, a serem utilizados para treinamento dos modelos de predição, como dados do histórico escolar e acadêmico, adquiridos possivelmente por meio de bancos de dados das próprias IESs. Além disso, é preciso ainda discernir se o discente é novato ou veterano. Com tal cadastro, esses dados dos novos discentes são armazenados no banco de dados da plataforma TôSabendo.

- Passo 2: consiste na seleção e aplicação de um dos modelos de predição treinados considerando os dados do novo discente úteis para a predição, sendo consultados e obtidos do banco de dados. Ressalta-se que os dados adquiridos dos discentes novatos são diferentes dos veteranos. Então, a consulta no banco é diferente. Além disso, diferentes modelos são aplicados. Porém, no fim, todos os resultados de predição têm o mesmo significado perante o novo discente: ter ou não um bom desempenho na plataforma. Portanto, a predição é realizada, retornada e armazenada no banco de dados da plataforma.

- Passo 3: consiste no aperfeiçoamento dos modelos de predição por meio de um retreinamento de tempos em tempos dos mesmos com os dados de diversos novos discentes que cadastrarem na plataforma e já passaram por uma predição. Para isso, estabelece-se uma quantidade miníma de novos discentes, tanto para o modelo de predição para novatos quanto para o de veteranos. Portanto, o novo modelo treinado é trocado pelo antigo até então utilizado no Passo 2.

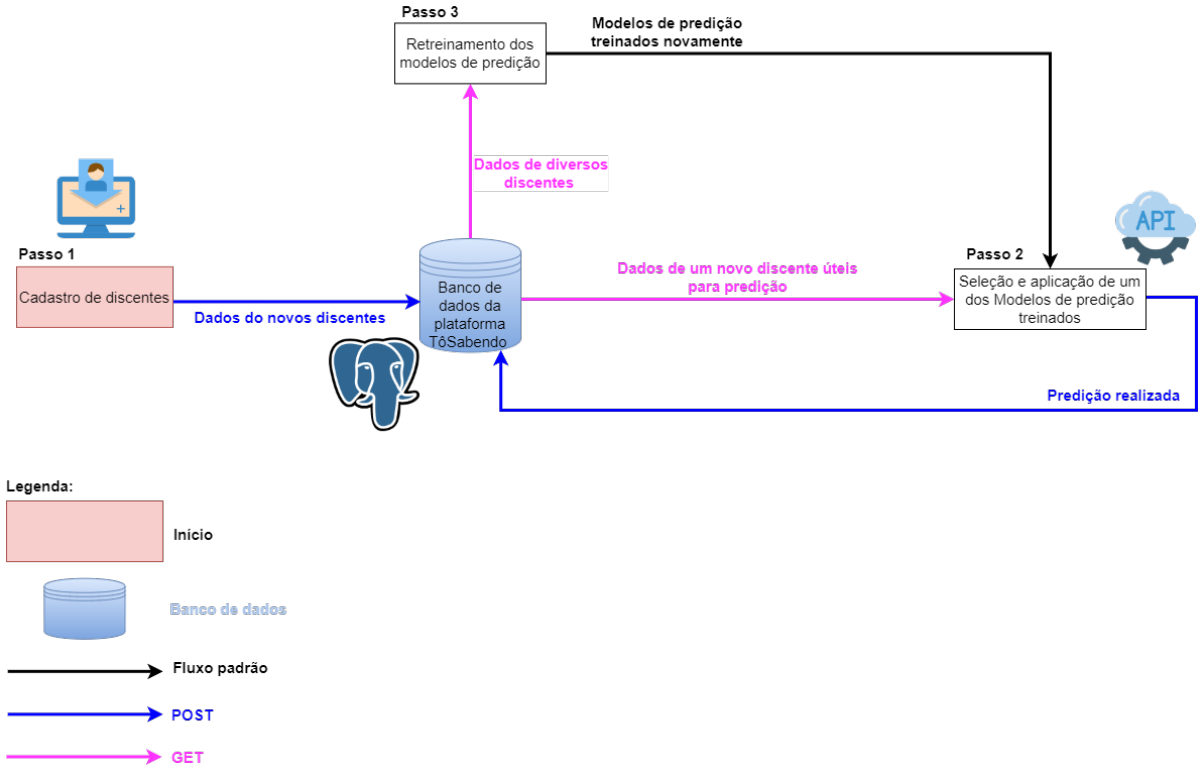

Figura A.1 – Funcionamento da Plataforma TôSabendo com modelos de predição incorporados

Portanto, com modelos de predição de desempenho incorporados a uma versão funcional da Plataforma TôSabendo, professores poderão ter um conhecimento previsto de como um determinado discente pode se sair em Quizzes voltados para um ou mais determinados conteúdos programáticos, no intuito de poder aperfeiçoar o ensino e as atividades de tais conteúdos em sala de aula e também na própria plataforma.

Anexos

ANEXO A – Antigo Esquema relacional da plataforma TôSabendo

Este anexo apresenta o antigo esquema relacional da plataforma TôSabendo projetado por (Ferreira, 2022) a partir do esquema conceitual EER da Figura 2.2.

Usuario(email, senha, nome, sobrenome, sexo, senha, data_cadastro, ultimo_acesso, curso, instituicao, localidade)

Usuario[curso] → Curso[id] **Bloqueio**

Usuario[instituição] → Instituicao[id] **Bloqueio**

Usuario[localidade] → Localidade[id] **Bloqueio**

Curso(id, nome)

Instituicao(id, nome, sigla)

Localidade(id, uf, cidade)

Jogador(id_usuario, biografia, pontuacao_geral, nivel)

Jogador[id_usuario] → Usuario[email] **Propagação**

Colaborador(id_usuario, codigo, pontuacao_colab, nivel, acesso_admin)

Colaborador[id_usuario] → Usuário[email] **Propagação**

Colaborador[acesso_admin] → Admin[id_usuario] **Substituição por nulo**

Administrador(id_usuario, classificacao)

Administrador[id_usuario] → Usuário[email] **Propagação**

Jogo(id, nome, criador)

Jogo[criador] → Colaborador[id_usuario] **Substituição por Default**

Jogo_Colaborador(jogo, colaborador, data_ingresso)

Jogo_Colaborador[jogo] → Jogo[id] **Bloqueio**

Jogo_Colaborador[colaborador] → Colaborador[id_usuario] **Bloqueio**

Questao(id, enunciado, fig_enunciado, data_cadastro, criador, fonte, nivel_dificuldade)

Questão[criador] → Colaborador[id_usuario] **Substituição por Default**

Questão[fonte] → Fonte[id] **Bloqueio**

Questão[nivel_dificuldade] → Nivel_Dificuldade[id] **Bloqueio**

Dica_Questao(questao, dica)

Dica_Questao[questao] → Questao[id] **Propagação**

Quizz(jogo, questao)

Quizz[jogo] → Jogo[id] **Bloqueio**

Quizz[questao] → Questão[id] **Bloqueio**

Jogador_Joga(id_jogador, id_jogo, vezes_jogadas, data_inicio, data_termino, pontuacao_atual, melhor_pontuacao, finalizado)

Jogador_Joga[id_jogador] → Jogador[id_usuario] **Bloqueio**

Jogador_Joga[id_jogo] → Jogo[id] **Bloqueio**

Questao_Jogada(numero, id_jogador, id_jogo, id_questao, situacao, data)

Questao_Jogada[id_jogador] → Jogador[id_usuario] **Propagação**

Questao_Jogada[id_jogo, id_questao] → Quiz[jogo, questao] **Propagação**

Categoria(id, categoria)

Subcategoria(id, subcategoria, id_categoria)

Subcategoria[id_categoria] → Categoria[id] **Propagação**

Subcategoria_Questao(<u>questao</u>, <u>subcategoria</u>)

Subcategoria_Questao[questao] → Questao[id] **Bloqueio**

Subcategoria_Questao[subcategoria] → Subcategoria[id] **Bloqueio**

Alternativa(<u>id</u>, letra, correta, questao)

Alternativa[questao] → Questao[id] **Propagação**

Nivel_Dificuldade(<u>id</u>, nome, tempo)

Fonte(<u>id</u>, nome, url)

ANEXO B – Novo Esquema relacional da plataforma TôSabendo

Este anexo apresenta o novo esquema relacional da plataforma TôSabendo projetado a partir do novo esquemca conceitual EER apresentado na Figura 3.3. As alterações, em relação ao esquema relacional antigo descrito no Anexo A, encontram-se em vermelho.

Usuario(cpf, email, senha, nome, sobrenome, sexo, idade, senha, data_cadastro, ultimo_acesso, id_curso, id_instituicao)

Usuario[id_curso] → Curso[id] **Bloqueio**

Usuario[id_instituição] → Instituicao[id] **Bloqueio**

Curso(id, nome)

Instituicao(id, nome, sigla, uf, municipio)

Jogador(id_usuario, matricula, data_ingresso_ie, pontuacao_geral, nivel)

Jogador[id_usuario] → Usuario[email] **Propagação**

Colaborador(id_usuario, codigo, pontuacao_colab, nivel, id_admin)

Colaborador[id_usuario] → Usuário[email] **Propagação**

Colaborador[id_admin] → Admin[id_usuario] **Substituição por Nulo**

Administrador(id_usuario, classificacao)

Administrador[id_usuario] → Usuário[email] **Propagação**

Predicao(id_jogador, data_realizada, resultado)

Predicao[id_jogador] → Jogador[id_usario] **Propagação**

Historico_Academico(id, nota_media, coeficiente)

Historico_Academico[id] → Jogador[id_usario] **Propagação**

Historico_Escolar(id, nota_media, escola_ensino_medio, data_conclusao, nota_portugues, nota_matematica)

Historico_Escolar[id] → Jogador[id_usario] **Propagação**

Disciplina_Cursada(codigo, semestre, media, aprovado, id_historico_academico)

Disciplina_Cursada[id_historico_academico] → Jogador[id_usario] **Propagação**

Jogo(id, nome, id_criador)

Jogo[id_criador] → Colaborador[id_usuario] **Substituição por Default**

Contribuicao(id_jogo, id_colaborador, data_ingresso)

Contribuicao[id_jogo] → Jogo[id] **Bloqueio**

Contribuicao[id_colaborador] → Colaborador[id_usuario] **Bloqueio**

Questao(id, enunciado, fig_enunciado, data_cadastro, id_criador, id_fonte, id_nivel_dificuldade)

Questão[id_criador] → Colaborador[id_usuario] **Substituição por Default**

Questão[id_fonte] → Fonte[id] **Bloqueio**

Questão[id_nivel_dificuldade] → Nivel_Dificuldade[id] **Bloqueio**

Dica_Questao(id_questao, dica)

Dica_Questao[id_questao] → Questao[id] **Propagação**

Quizz(id_jogo, id_questao)

Quizz[id_jogo] → Jogo[id] **Bloqueio**

Quizz[id_questao] → Questão[id] **Bloqueio**

Jogador_Joga(id_jogador, id_jogo, vezes_jogadas, data_inicio, data_termino, pontuacao_atual, melhor_pontuacao, finalizado)

Jogador_Joga[id_jogador] → Jogador[id_usuario] **Bloqueio**

Jogador_Joga[id_jogo] → Jogo[id] **Bloqueio**

Duvida(id, descricao, data, id_jogador, id_jogo)

Duvida[id_jogador, id_jogo] → Jogador_Joga[id_jogador, id_jogo] **Bloqueio**

Questao_Jogada(numero, situacao, data id_jogador, id_jogo, id_questao)

Questao_Jogada[id_jogador] → Jogador[id_usuario] **Propagação**

Questao_Jogada[id_jogo, id_questao] → Quiz[id_jogo, id_questao] **Propagação**

Categoria(id, categoria)

Subcategoria(id, subcategoria, id_categoria)

Subcategoria[id_categoria] → Categoria[id] **Propagação**

Subcategoria_Questao(id_questao, id_subcategoria)

Subcategoria_Questao[id_questao] → Questao[id] **Bloqueio**

Subcategoria_Questao[id_subcategoria] → Subcategoria[id] **Bloqueio**

Alternativa(id, letra, correta, id_questao)

Alternativa[id_questao] → Questao[id] **Propagação**

Nivel_Dificuldade(id, nome, tempo)

Fonte(id, nome, url)

ANEXO C – Esquema físico do banco de dados da plataforma TôSabendo

Este anexo apresenta o esquema físico da plataforma TôSabendo, implementado na ORM denominada Prisma e implantado no SGDB PostgreSQL, a partir do novo esquema relacional apresentado no Anexo B.

```
model Usuario {
    cpf            String   @id @db.VarChar(12)
    email          String   @unique @db.VarChar(100)
    senha          String   @db.VarChar(100)
    nome           String   @db.Text
    sobrenome      String   @db.Text
    sexo           String   @db.Char(2)
    idade          Int
    data_cadastro  DateTime @default(now())
    updated_at     DateTime @updatedAt
    ultimo_acesso  DateTime
    id_curso       Int
    id_instituicao Int

    curso       Curso       @relation(fields: [id_curso], references: [id], onDelete: Restrict, onUpdate: Restrict)
    instituicao Instituicao @relation(fields: [id_instituicao], references: [id], onDelete: Restrict, onUpdate: Restrict)

    Jogador       Jogador[]
    Colaborador   Colaborador[]
    Administrador Administrador[]

    @@map("usuarios")
}
```

Figura C.1 – *Model* da tabela Usuário

```
model Curso {
    id      Int       @id @default(autoincrement())
    nome    String    @db.Text
    Usuario Usuario[]

    @@map("cursos")
}
```

Figura C.2 – *Model* da tabela Curso

```
model Instituicao {
    id          Int       @id @default(autoincrement())
    nome        String    @db.Text
    sigla       String    @db.Char(8)
    uf          String    @db.Char(3)
    municipio   String    @db.VarChar(50)
    Usuario     Usuario[]

    @@map("instituicoes")
}
```

Figura C.3 – *Model* da tabela Instituição

```
model Jogador {
    id_usuario       String   @id
    matricula        String   @db.VarChar(15)
    data_ingresso_ie DateTime
    pontuacao_geral  Float?
    nivel            Int      @default(0)

    usuario Usuario @relation(fields: [id_usuario], references: [cpf], onDelete: Cascade, onUpdate: Cascade)

    Predicao           Predicao[]
    HistoricoAcademico HistoricoAcademico[]
    HistoricoEscolar   HistoricoEscolar[]
    JogadorJoga        JogadorJoga[]
    QuestaoJogada      QuestaoJogada[]

    @@map("jogadores")
}
```

Figura C.4 – *Model* da tabela Jogador

```
model Colaborador {
    id_usuario      String  @id
    codigo          String  @db.VarChar(15)
    pontuacao_colab Float?
    nivel           Int     @default(0)
    id_admin        String?

    usuario       Usuario       @relation(fields: [id_usuario], references: [cpf], onDelete: Cascade, onUpdate: Cascade)
    administrador Administrador? @relation(fields: [id_admin], references: [id_usuario], onDelete: SetNull, onUpdate: SetNull)

    Jogo         Jogo[]
    Contribuicao Contribuicao[]
    Questao      Questao[]

    @@map("colaboradores")
}
```

Figura C.5 – *Model* da tabela Colaborador

```
model Administrador {
    id_usuario    String @id
    classificacao Int    @default(0)

    usuario Usuario @relation(fields: [id_usuario], references: [cpf], onDelete: Cascade, onUpdate: Cascade)

    Colaborador Colaborador[]

    @@map("administradores")
}
```

Figura C.6 – *Model* da tabela Administrador

```
model Predicao {
    id_jogador     String
    data_realizada DateTime @default(now())
    resultado      String   @db.VarChar(30)

    jogador Jogador @relation(fields: [id_jogador], references: [id_usuario], onDelete: Cascade, onUpdate: Cascade)

    @@id([id_jogador, data_realizada])
    @@map("predicoes")
}
```

Figura C.7 – *Model* da tabela Predição

```
model HistoricoAcademico {
    id          String @id
    nota_media  Float
    coeficiente Float

    jogador Jogador @relation(fields: [id], references: [id_usuario], onDelete: Cascade, onUpdate: Cascade)

    DisciplinaCursada DisciplinaCursada[]

    @@map("historicos_academicos")
}
```

Figura C.8 – *Model* da tabela Histórico_Acadêmico

```
model HistoricoEscolar {
    id                  String   @id
    nota_media          Float
    escola_ensino_medio String   @db.VarChar(80)
    data_conclusao      DateTime
    nota_portugues      Float
    nota_matematica     Float

    jogador Jogador @relation(fields: [id], references: [id_usuario], onDelete: Cascade, onUpdate: Cascade)

    @@map("historicos_escolares")
}
```

Figura C.9 – *Model* da tabela Histórico_Escolar

```
model DisciplinaCursada {
    codigo               String  @db.VarChar(15)
    semestre             String  @db.VarChar(10)
    media                Float
    aprovado             Boolean @default(true)
    id_historico_academico String

    historicoAcademico HistoricoAcademico @relation(fields: [id_historico_academico], references: [id], onDelete: Cascade, onUpdate: Cascade)

    @@id([codigo, semestre])
    @@map("disciplinas_cursadas")
}
```

Figura C.10 – *Model* da tabela Disciplina_Cursada

```
model Jogo {
    id         String @id @default(uuid())
    nome       String @db.Text
    id_criador String @default("Anônimo")

    criador Colaborador @relation(fields: [id_criador], references: [id_usuario], onDelete: SetDefault, onUpdate: SetDefault)

    Contribuicao Contribuicao[]
    Quizz        Quizz[]
    JogadorJoga  JogadorJoga[]

    @@map("jogos")
}
```

Figura C.11 – *Model* da tabela Jogo

```
model Contribuicao {
    id_jogo        String
    id_colaborador String
    data_ingresso  DateTime @default(now())

    jogo        Jogo        @relation(fields: [id_jogo], references: [id], onDelete: Restrict, onUpdate: Restrict)
    colaborador Colaborador @relation(fields: [id_colaborador], references: [id_usuario], onDelete: Restrict, onUpdate: Restrict)

    @@id([id_jogo, id_colaborador])
    @@map("contribuicoes")
}
```

Figura C.12 – *Model* da tabela Contribuição

```
model Questao {
    id                  String   @id @default(uuid())
    enunciado           String?  @db.VarChar(1000)
    fig_enunciado       String?  @db.VarChar(200)
    data_cadastro       DateTime @default(now())
    id_criador          String   @default("Anônimo")
    id_fonte            String
    id_nivel_dificuldade String

    criador          Colaborador      @relation(fields: [id_criador], references: [id_usuario], onDelete: SetDefault, onUpdate: SetDefault)
    fonte            Fonte            @relation(fields: [id_fonte], references: [id], onDelete: Restrict, onUpdate: Restrict)
    nivelDificuldade NivelDificuldade @relation(fields: [id_nivel_dificuldade], references: [id], onDelete: Restrict, onUpdate: Restrict)

    DicaQuestao         DicaQuestao[]
    Quizz               Quizz[]
    SubcategoriaQuestao SubcategoriaQuestao[]
    Alternativa         Alternativa[]

    @@map("questoes")
}
```

Figura C.13 – *Model* da tabela Questão

```
model DicaQuestao {
    id_questao String
    dica       String @db.VarChar(500)

    questao Questao @relation(fields: [id_questao], references: [id], onDelete: Cascade, onUpdate: Cascade)

    @@id([id_questao, dica])
    @@map("dicas_questoes")
}
```

Figura C.14 – *Model* da tabela Dica_Questão

```
model Quizz {
    id_jogo    String
    id_questao String

    jogo    Jogo    @relation(fields: [id_jogo], references: [id], onDelete: Restrict, onUpdate: Restrict)
    questao Questao @relation(fields: [id_questao], references: [id], onDelete: Restrict, onUpdate: Restrict)

    QuestaoJogada QuestaoJogada[]

    @@id([id_jogo, id_questao])
    @@map("quizzes")
}
```

Figura C.15 – *Model* da tabela Quizz

```
model JogadorJoga {
    id_jogador       String
    id_jogo          String
    vezes_jogadas    Int       @default(0)
    data_inicio      DateTime  @default(now())
    data_termino     DateTime?
    pontuacao_atual  Float?
    melhor_pontuacao Float?
    finalizado       Boolean   @default(false)

    jogador Jogador  @relation(fields: [id_jogador], references: [id_usuario], onDelete: Restrict, onUpdate: Restrict)
    jogo    Jogo     @relation(fields: [id_jogo], references: [id], onDelete: Restrict, onUpdate: Restrict)
    Duvida  Duvida[]

    @@id([id_jogador, id_jogo])
    @@map("jogadores_jogam")
}
```

Figura C.16 – *Model* da tabela Jogador_Joga

```
model Duvida {
    id         String   @id @default(uuid())
    descricao  String   @db.VarChar(50)
    data       DateTime @default(now())
    id_jogador String
    id_jogo    String

    jogadorJoga JogadorJoga @relation(fields: [id_jogador, id_jogo], references: [id_jogador, id_jogo], onDelete: Restrict, onUpdate: Restrict)

    @@map("duvidas")
}
```

Figura C.17 – *Model* da tabela Duvida

```
model QuestaoJogada {
    numero     String   @id @default(uuid())
    situacao   String   @db.VarChar(30)
    data       DateTime @default(now())
    id_jogador String
    id_jogo    String
    id_questao String

    jogador Jogador @relation(fields: [id_jogador], references: [id_usuario], onDelete: Cascade, onUpdate: Cascade)
    quizz   Quizz   @relation(fields: [id_jogo, id_questao], references: [id_jogo, id_questao], onDelete: Cascade, onUpdate: Cascade)

    @@map("questoes_jogadas")
}
```

Figura C.18 – *Model* da tabela Questão_Jogada

```
model Categoria {
    id         Int    @id @default(autoincrement())
    categoria String @db.VarChar(50)

    Subcategoria Subcategoria[]

    @@map("categorias")
}
```

Figura C.19 – *Model* da tabela Categoria

```
model Subcategoria {
    id           Int    @id @default(autoincrement())
    subcategoria String @db.VarChar(50)
    id_categoria Int

    categoria Categoria @relation(fields: [id_categoria], references: [id], onDelete: Cascade)

    SubcategoriaQuestao SubcategoriaQuestao[]

    @@map("subcategorias")
}
```

Figura C.20 – *Model* da tabela Subcategoria

```
model SubcategoriaQuestao {
    id_questao      String
    id_subcategoria Int

    questao      Questao      @relation(fields: [id_questao], references: [id], onDelete: Restrict, onUpdate: Restrict)
    subcategoria Subcategoria @relation(fields: [id_subcategoria], references: [id], onDelete: Restrict, onUpdate: Restrict)

    @@id([id_questao, id_subcategoria])
    @@map("subcategorias_questoes")
}
```

Figura C.21 – *Model* da tabela Subcategoria_Questão

```
model Alternativa {
    id         String  @id @default(uuid())
    letra      String  @db.Char(2)
    correta    Boolean
    id_questao String

    questao Questao @relation(fields: [id_questao], references: [id], onDelete: Cascade, onUpdate: Cascade)

    @@map("alternativas")
}
```

Figura C.22 – *Model* da tabela Alternativa

```
model NivelDificuldade {
    id    String @id @default(uuid())
    nome  String @db.Text
    tempo Int?

    Questao Questao[]

    @@map("niveis_dificuldade")
}
```

Figura C.23 – *Model* da tabela Nivel_Dificuldade

```
model Fonte {
    id   String @id @default(uuid())
    nome String @db.Text
    url  String @db.VarChar(400)

    Questao Questao[]

    @@map("fontes")
}
```

Figura C.24 – *Model* da tabela Fonte