DreamLearning: Um Site para o Estudo e Aplicação de Aprendizado de Máquina

Eduardo dos Santos Rocha

Universidade Federal de Uberlândia
Faculdade de Computação
Bacharelado em Sistemas de Informação

Monte Carmelo - MG
2025

Eduardo dos Santos Rocha

DreamLearning: Um Site para o Estudo e Aplicação de Aprendizado de Máquina

Trabalho de Conclusão de Curso apresentado à Faculdade de Computação da Universidade Federal de Uberlândia, Minas Gerais, como requisito exigido parcial à obtenção do grau de Bacharel em Sistemas de Informação.

Área de concentração: Sistemas de Informação

Orientador: Fernanda Maria da Cunha Santos
Coorientador: Sara Luzia de Melo

Monte Carmelo - MG
2025

Este trabalho é dedicado, primeiramente, a Deus, à minha mãe, minha irmã, minha avó, meu avô, minha tia e meu primo, por sempre conhecerem minhas lutas e me apoiarem.

Agradecimentos

Agradeço imensamente à minha família, começando pela minha mãe, que foi mãe e pai ao mesmo tempo e sempre cuidou de mim e da minha irmã com todo amor e dedicação. À minha avó, que nunca deixou de apoiar minhas ideias por mais malucas que fossem e que sempre sonhou em ver os netos formados. Ao meu avô, que foi um verdadeiro exemplo de pai e cuidou de todos nós com muito zelo. À minha irmã, parceira de vida, que sempre esteve ao meu lado, apoiando cada passo que dei. Não poderia deixar de agradecer também à minha tia e ao meu primo, que sempre me incentivaram e me deram apoio, principalmente quando precisei viajar para Uberlândia para apresentar artigos e trabalhos.

Minha gratidão se estende à Professora Orientadora Fernanda Maria da Cunha Santos, que acreditou na minha ideia de TCC desde o início e me acompanhou com dedicação e paciência ao longo desse processo. Agradeço também à Professora Coorientadora Sara, que, desde o primeiro dia em que apresentei este projeto, demonstrou entusiasmo e tem sido uma grande apoiadora, inclusive durante o estágio.

Agradeço a Deus por ter me dado força e equilíbrio para seguir firme até o fim e concluir a faculdade sem nenhuma reprovação. Aos meus amigos, por estarem sempre por perto, nos momentos bons e nos difíceis. E, por último, mas não menos importante, agradeço ao Rock Lee, que me serviu de inspiração e motivação em muitas madrugadas de estudo.

"Você não vai ser substituido pela inteligência artificial, mas sim por quem usa melhor a inteligência artificial"

(Juliano Kimura)

Resumo

Este trabalho apresenta o desenvolvimento de uma plataforma de aprendizagem, denominada DreamLearning, voltada ao ensino e à aplicação de algoritmos de aprendizado de máquinas. Esta proposta busca preencher lacunas existentes no ensino dessa área, para estudantes de graduação, a fim de implementar uma ferramenta acessível, interativa, baseada em software de código aberto, no idioma português. Para isto, a metodologia adotada seguiu princípios da engenharia de software, incluindo a definição e modelagem clara dos requisitos, bem como uso de tecnologias abertas, tais como Scikit-learn, Flask e Bootstrap. Como resultados, a plataforma desenvolvida permite a realização de tarefas de classificação, regressão e análise associativa, empregando algoritmos supervisionados e não supervisionados. Entre as funcionalidades disponíveis no ambiente educacional, destacam-se o treinamento e teste de modelos, ajustes de hiper parâmetros, validação cruzada, visualização de métricas de desempenho e, por fim, a exportação do código-fonte gerado. Adicionalmente, foi implementado um agente conversacional com o objetivo de sanar dúvidas dos estudantes durante o processo de aprendizagem. A plataforma também oferece videoaulas, elaboradas com base em metodologias pedagógicas, buscando proporcionar uma melhor compreensão dos conteúdos, além de visualizações gráficas e acesso ao histórico das simulações realizadas. Testes de usabilidade foram realizados com estudantes de graduação, onde foi observado que o DreamLearning é uma ferramenta promissora no ensino de algoritmos de aprendizado de máquinas. A plataforma proposta, de maneira acessível e interativa, apresenta aos estudantes conceitos fundamentais da área, permitindo a simulação prática e acesso ao código dos algoritmos utilizados.

Palavras-chave: Aprendizado de máquina, Aprendizado Supervisionado, Aprendizado Não Supervisionado, Software Educacional, Software de Código Aberto.

Abstract

This paper presents the development of a learning platform, called DreamLearning, aimed at teaching and applying machine learning algorithms. This proposal seeks to fill existing gaps in the teaching of this area for undergraduate students, in order to implement an accessible, interactive tool, based on open source software, in the Portuguese language. To this end, the adopted methodology followed principles of software engineering, including the clear definition and modeling of requirements, as well as the use of open technologies, such as Scikit-learn, Flask and Bootstrap. As a result, the developed platform allows the performance of classification, regression and associative analysis tasks, using supervised and unsupervised algorithms. Among the functionalities available in the educational environment, we highlight the training and testing of models, hyperparameter adjustments, cross-validation, visualization of performance metrics and, finally, the export of the generated source code. Additionally, a conversational agent was implemented with the objective of clarifying students' doubts during the learning process. The platform also offers video lessons, developed based on pedagogical methodologies, seeking to provide a better understanding of the content, in addition to graphical visualizations and access to the history of the simulations performed. Usability tests were conducted with undergraduate students, where it was observed that DreamLearning is a promising tool for teaching machine learning algorithms. The proposed platform, in an accessible and interactive way, introduces students to fundamental concepts of the area, allowing practical simulation and access to the code of the algorithms used.

Keywords: Machine Learning, Supervised Learning, No Supervised Learning, Educational Software, Open Source Software.

Lista de ilustrações

Lista de tabelas

Lista de siglas

AM Aprendizado de Máquinas

API Interface de Programação de Aplicativos - *Application Programming Interface*

CatBoost Aumento Categórico - *Categorical Boosting*

CSV Valores separados por vírgula - *Comma Separated Values*

CSS Folhas de Estilo em Cascata - *Cascading Style Sheets*

CPU Unidade central de processamento - *Central Processing Unit*

CNN Redes Neurais Convolucionais - *Convolutional Neural Network*

EFB Pacote de recursos exclusivos - *Exclusive Feature Bundling*

GOSS Amostragem unilateral baseada em gradiente - *Gradient-Based One-Side Sampling*

HTTP Protocolo de Transferência de Hipertexto - *Hypertext Transfer Protocol*

HTML Linguagem de Marcação de HiperTexto - *HyperText Markup Language*

IA Inteligência Artificial

IDE Ambiente de desenvolvimento integrado - *Integrated Development Environment*

KNN K-ésimo Vizinho mais Próximo - *K-nearest neighbors algorithm*

LightGBM Modelo de aumento de gradiente claro - *Light Gradient Boosting Model*

ML4ALL Aprendizado de Máquina Para Todos - *Ma-chine Learning for All*

REA Recursos Educacionais Abertos - *Open Educational Resources*

OSER Recursos Educacionais de Código Aberto - *Open Source Educational Resources*

SUS Escala de Usabilidade do Sistema - *System Usability Scale*

SVM Máquina de Vetores de Suporte - *Support Vector Machine*

TI Tecnologia da informação

UML Linguagem de Modelagem Unificada - *Unified Modeling Language*

UI Interface do Usuário - *User Interface*

UX Experiência do Usuário - *User Experience*

XGBoost Aumento de gradiente extremo - *eXtreme Gradient Boosting*

Sumário

Capítulo 1

Introdução

A era digital é fruto dos trabalhos de pioneiros como Alan Turing, amplamente reconhecido como o "pai da computação"[1]. Sua criação, a "Máquina de Turing" (TURING, 1936) no século XX, estabeleceu a base para os computadores modernos e marcou o início de uma nova era no processamento de informações. Décadas depois, o legado de Turing reflete diretamente nas tecnologias do século XXI, como a Inteligência Artificial (IA) e os algoritmos de *Machine Learning (ML)*, que em português significa Aprendizado de Máquinas (AM). O AM é um campo da IA que possibilita que os computadores aprendam a partir de um conjunto de dados, permitindo que eles identifiquem padrões e façam previsões sem a necessidade de programação explícita para cada tarefa.

Levando em consideração a utilização de técnicas de AM na educação, uma busca realizada na plataforma "Google Acadêmico" [2] utilizando os termos de pesquisa "*machine learning AND education*" no período de 2021 a 2024, resultou em 1.047.000 publicações. Além dessa combinação de termos, outras buscas realizadas no mesmo intervalo de tempo, tais como "*Machine Learning AND Education AND Undergraduate*", "*Machine Learning AND Education AND Undergraduate AND Open Source Software*" e "*Machine Learning AND Education AND Undergraduate AND Open Source Software AND Portuguese*", resultaram em, respectivamente, 104.300, 106.900 e 66.700 publicações. Diante dos números expostos, destaca-se a presença de técnicas de AM no desenvolvimento de pesquisas exploratórias, principalmente no desenvolvimento de produtos e tecnologias com ênfase no setor educacional. No entanto, devido à escassez de publicações em português sobre o tema, a busca foi predominantemente realizada em inglês.

O trabalho de Çağlayan (2019), foi exposto o ensino de algoritmos de AM utilizando a plataforma Orange[3], como uma ferramenta com interface gráfica, e a linguagem C, como uma ferramenta sem interface gráfica. O estudo avaliou o nível de interesse de 11 alunos pelo tema, bem como a percepção e o conhecimento dos estudantes, que estavam nos

[1] <https://www.ebiografia.com/alan_turing/>
[2] <https://scholar.google.com/>
[3] <https://orangedatamining.com/>

primeiros períodos do curso de Engenharia da Computação. Os estudantes foram divididos em dois grupos (A e B), e ambos receberam o mesmo conteúdo teórico sobre o algoritmo K-ésimo Vizinho mais Próximo (KNN). Desse modo, um grupo aprendeu o algoritmo em linguagem C apenas por uma Ambiente de desenvolvimento integrado (IDE), sem interface gráfica, enquanto outro grupo utilizou a plataforma Orange[3]. Os resultados, obtidos por meio de questionários, indicaram um maior nível de interesse dos alunos em utilizar a ferramenta Orange[3], devido sua interface visual intuitiva, abstraindo a necessidade de aprendizado de código para iniciantes.

Na revisão da literatura realizada por Kučak, Juričić e Đambić (2018), observaram que há uma carência de pesquisas que objetivam o desenvolvimento de soluções que utilizam métodos de computação para promover o aprendizado. Nota-se apenas soluções direcionados a problemas relacionados à educação, como exemplos, avaliação de alunos, previsão de desempenho, dentre outros.

Segundo Naik e Samant (2016), as principais ferramentas computacionais utilizadas para verificar resultados de algoritmos de AM são o Weka[4], RapidMiner[5], Knime[6] e Orange[3]. Esses softwares facilitam a aplicação dos algoritmos em diversas áreas do conhecimento, pois os problemas propostos pelo ensino superior que demanda uma abordagem multidisciplinar, envolvendo diferentes níveis e contextos educacionais. No entanto, essas ferramentas não oferecem acesso ao código-fonte dos algoritmos selecionados pelos usuários.

1.1 Objetivos

O principal objetivo desse trabalho é proporcionar o ensino acessível e eficiente dos principais algoritmos de AM, supervisionado e não supervisionado, para estudantes de graduação, tendo como idioma principal a língua portuguesa, bem como o compartilhamento de recursos educacionais que facilitem a aprendizagem. Assim sendo, desenvolvido um site como ferramenta educacional criativa, com recursos gráficos e seguindo as normas regidas pelo software livre e engenharia de software, capaz de integrar os algoritmos de AM na formação dos cursos superiores. Além da utilização de um agente conversacional para auxiliar na compreensão do aplicativo. Logo, o objetivo geral se desdobra nos seguintes objetivos específicos:

- ❏ Analisar os algoritmos de AM e escolher quais são os mais adequados a proposta do projeto.
- ❏ Determinar as tecnologias necessárias para a criação do site.

[4] <https://ml.cms.waikato.ac.nz/weka/>
[5] <https://rapidminer.qsoft.com.br/>
[6] <https://www.knime.com/>

- ❏ Desenvolver uma documentação técnica detalhada, incluindo a modelagem de diagramas de Linguagem de Modelagem Unificada (UML) com foco na criação de um software livre.
- ❏ Desenvolvimento de um agente conversacional, por meio de um chatbot, para auxiliar na compreensão e utilização do aplicativo.
- ❏ Aplicação rápida e simplificada do teste Escala de Usabilidade do Sistema (SUS) para avaliar a usabilidade da primeira versão pública do sistema.

Para alcançar os objetivos propostos, utilizou-se a linguagem Python 3.12.4 com as bibliotecas Scikit-learn[8], CatBoost[7], XGBoost[9], LightGBM[10] e mlxtend[11]. Para a análise gráfica, utilizou-se a biblioteca Plotly[12]. A interface gráfica foi desenvolvida com Flask, Bootstrap e NES.css, utilizando HTML e JavaScript para o *front-end*. E para o agente conversacional utilizou-se o a Interface de Programação de Aplicativos (API) do Google Gemini.

1.2 Motivação

No início do segundo semestre de 2024, foi realizada uma pesquisa exploratória com foco em aplicativos voltados para o ensino de AM, como Weka[4], RapidMiner[5], KNIME[6] e Orange[3]. A investigação revelou a ausência de versões web acessíveis dessas ferramentas, bem como a limitada disponibilidade de conteúdos totalmente em língua portuguesa. Além disso, observou-se a falta de recursos educacionais integrados que não apenas ensinem, mas também permitam a prática dos conceitos de forma interativa. Diante desse cenário, surgiu a motivação para o desenvolvimento da plataforma DreamLearning, uma aplicação web que visa auxiliar usuários na compreensão e aplicação prática de algoritmos de AM supervisionados e não supervisionados. A plataforma oferece vídeo aulas, código-fonte aberto, recursos interativos em português e um agente conversacional que responde dúvidas, orienta o uso da ferramenta e contribui para o processo de aprendizagem de forma dinâmica e acessível.

1.3 Organização do Trabalho

A estrutura deste trabalho consiste em quatro capítulos organizados conforme descrição abaixo:

[8] <https://scikit-learn.org/stable/modules/tree.html>
[7] <https://catboost.ai/>
[9] <https://xgboost.readthedocs.io/en/stable/>
[10] <https://lightgbm.readthedocs.io/en/latest/pythonapi/lightgbm.LGBMClassifier.html>
[11] <https://rasbt.github.io/mlxtend/api_modules/mlxtend.frequent_patterns/apriori/>
[12] <https://plotly.com/>

- Capítulo 1: Introdução, que esclarece de forma mais detalhada a contextualização do projeto, os problemas abordados, os objetivos que orientaram seu desenvolvimento e os resultados esperados;
- Capítulo 2: Revisão teórica, abordando os principais assuntos teóricos que alicerçam este estudo, além de descrever os trabalhos relacionados que fortaleceram a ideia e objetivos propostos;
- Capítulo 3: Procedimentos metodológicos, onde são apresentados os métodos empregados na condução da pesquisa;
- Capítulo 4: Apresentação das considerações finais do trabalho e trabalhos futuros.

Capítulo 2

Fundamentação Teórica

Este capítulo apresenta os conceitos fundamentais que compõem a Fundamentação Teórica, abordando o significado de IA e AM, bem como os algoritmos utilizados no desenvolvimento da proposta deste trabalho, detalhados nas Seções 2.1 a 2.10. Na sequência, a Seção 2.11 expõe os trabalhos correlatos à abordagem proposta, enquanto a Seção 2.12 traz uma análise comparativa desses estudos.

2.1 Inteligência Artificial e Aprendizado de Máquina

A origem etimológica do termo Inteligência Artificial é apresentada por Fernandes (2004). A palavra "inteligência" deriva do latim, onde seu significado provém da junção dos termos *intus* e *legere*, que significam "entre" e "escolher", respectivamente. No entanto, o termo "artificial" vem também do latim *artificiale*, que se refere a algo que não ocorre naturalmente, mas é produzido pelo ser humano. Com essa definição em mente, pode-se entender que a IA é uma criação humana com o propósito de fazer com que máquinas simulem o raciocínio semelhante ao da inteligência humana.

Assim, IA é definida por Barbosa e Portes (2023) como uma tecnologia que possibilita aos dispositivos eletrônicos funcionarem de forma semelhante ao pensamento humano. Desta forma, a IA tem o intuito de criar sistemas que imitem a capacidade humana de raciocinar, inferir e tomar decisões. Já para Teixeira (2019), a criação de programas que imitem a capacidade de raciocínio do ser humano inclui a habilidade de perceber o mundo, identificar objetos e até mesmo compreender a linguagem natural.

Segundo Russell e Norvig (2010), a IA possui duas dimensões: uma dimensão que imita o comportamento humano ao pensar e agir, e a outra dimensão é o comportamento racional de pensar e agir. A IA pode variar desde um sistema que imita o comportamento humano até um que toma decisões racionais baseadas em princípios de racionalidade como silogismo, representação do conhecimento informal e lógica de 1ª ordem

Em suma, a IA possui quatro definições conforme exposto em Mueller e Massaron (2021) e em Russell e Norvig (2010), sendo elas:

- ❑ **Definição 1:** Sistemas que pensam como seres humanos, realizando tarefas que requerem inteligência, em vez de apenas seguirem instruções mecânicas.
- ❑ **Definição 2:** Sistemas que agem como seres humanos, sendo que seu sucesso é avaliado pela dificuldade em diferenciar suas ações das de um ser humano, conforme o Teste de Turing (TURING, 2009).
- ❑ **Definição 3:** Sistemas que pensam de forma racional, seguindo princípios que refletem comportamentos humanos típicos, dentro de níveis aceitáveis de variação.
- ❑ **Definição 4:** Sistemas que agem de forma racional, interagindo com o ambiente com base em ações registradas e considerando fatores e dados previamente estabelecidos.

Por outro lado, o AM, uma subárea da IA, visa identificar padrões por meio de algoritmos que aprendem a partir de dados, utilizando métodos computacionais para realizar previsões ou tomar decisões (BIAMONTE et al., 2017). A principal tarefa do AM é justamente desenvolver esses algoritmos para criar modelos eficientes com base nos dados disponíveis (ZHOU, 2021).

Conforme mencionado por Zhou (2021), o AM baseia-se no aprendizado indutivo, o qual é o processo de transformar informações específicas em conceitos gerais. Esse aprendizado é dividido em duas categorias: aprendizado supervisionado, que utiliza dados rotulados, e aprendizado não supervisionado, que trabalha com dados não rotulados.

No aprendizado supervisionado, os dados de entrada são fornecidos com os correspondentes resultados esperados (rótulos ou dados de saída). O objetivo central do modelo é aprender a relacionar as entradas com as saídas, visando prever corretamente novos resultados para dados desconhecidos. Essa abordagem é amplamente aplicada em tarefas de classificação e regressão, onde o modelo visa prever valores numéricos ou classificar a partir do processo de treinamento.

Por outro lado, no aprendizado não supervisionado, o modelo é treinado com dados que não possuem uma saída correspondente. O principal objetivo dessa técnica é a identificação de padrões. Técnicas comuns de aprendizado não supervisionado incluem *Clustering*, que agrupa dados com base em semelhanças e diferenças e regras de associação, que identifica uma relação entre as variáveis do conjunto de dados.

Embora a IA esteja frequentemente associada ao AM, a IA é um campo extenso e complexo, especialmente quando se trata de definir o que significa “inteligência” (MACEDO, 2023). No qual a IA faz parte de um campo mais amplo que incluem percepção, detecção, raciocínio e representação do conhecimento (JAN et al., 2023). Os algoritmos utilizados neste trabalho são berevementes descritos a seguir.

2.2 Naive Bayes

O modelo de aprendizado de máquina Naive Bayes é amplamente utilizado e reconhecido no campo acadêmico da estatística, sendo fundamentado nos estudos de Thomas Bayes (PANDA; PATRA, 2007). Seu principal objetivo é realizar previsões baseadas em probabilidades, ou seja, na frequência de ocorrência de cada valor presente nos dados de treino, a fim de determinar a probabilidade de um dado pertencer a uma classe específica (OLIVEIRA, 2021).

O modelo Naive Bayes exemplificado na Figura 1 é conhecido por sua simplicidade e eficiência computacional, tornando-se fácil de aplicar em grandes volumes de dados. Esse modelo parte do pressuposto de que as características utilizadas são independentes entre si e possuem o mesmo peso na decisão (MARTINEZ-ARROYO; SUCAR, 2006; ZHOU, 2021).

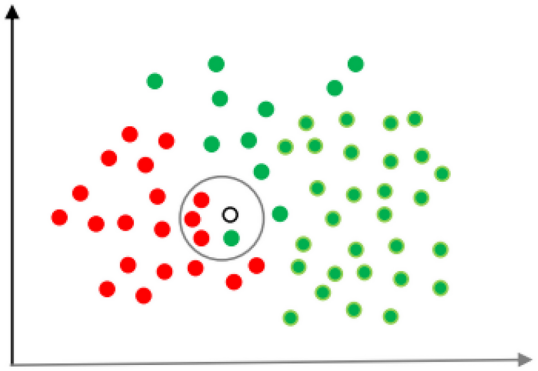


Figura 1 – Ilustração do algoritmo Naive Bayes.
Fonte: (ASGHARI; ARASTEH; KOOCHARI, 2024).

2.3 Máquina de Vetores de Suporte (SVM)

A metodologia da Máquina de Vetores de Suporte (SVM) foi fundamentada nos estudos de Vladimir Vapnik na década de sessenta (CORTES; VAPNIK, 1995). O conceito SVM é a extensão matemática de uma rede neural, podendo classificar dados lineares e não lineares (JAYADEVA; KHEMCHANDANI; CHANDRA, 2007; CHANDRA; BEDI, 2021).

A SVM classifica os dados lineares separando-os em diferentes categorias, como mostrado na Figura 2a, criando um hiperplano que separa geometricamente classes com base nas características, conforme ilustrado na Figura 2b. Os vetores de suporte mostrados na Figura 2c são os pontos mais próximos dessas linhas de separação, sendo fundamentais para o processo de classificação. Quando um novo dado é apresentado ao algoritmo, ele é classificado com base nesses vetores de suporte.

Contudo, existem dados que não podem ser separados linearmente, significando a impossibilidade de utilizar um hiperplano para dividir as classes, como mostrado na Figura 3a. Essa situação gera a necessidade de transformar os pontos da Figura 3a para uma dimensão maior, ou seja, adicionar informações aos dados. Após essa transformação, possibilita-se separar essas duas classes usando um hiperplano, conforme a Figura 3b (NORONHA; FERNANDES, 2016).

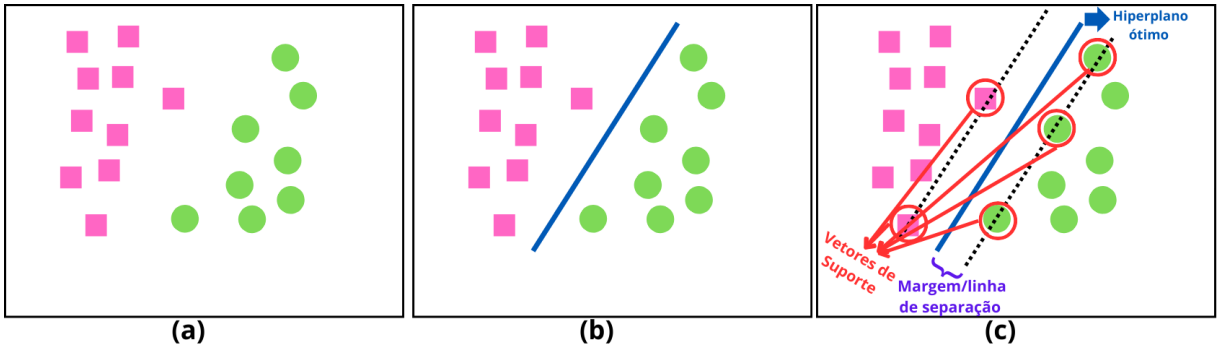


Figura 2 – Conjunto de dados linearmente separável.
Fonte: Autoria própria.

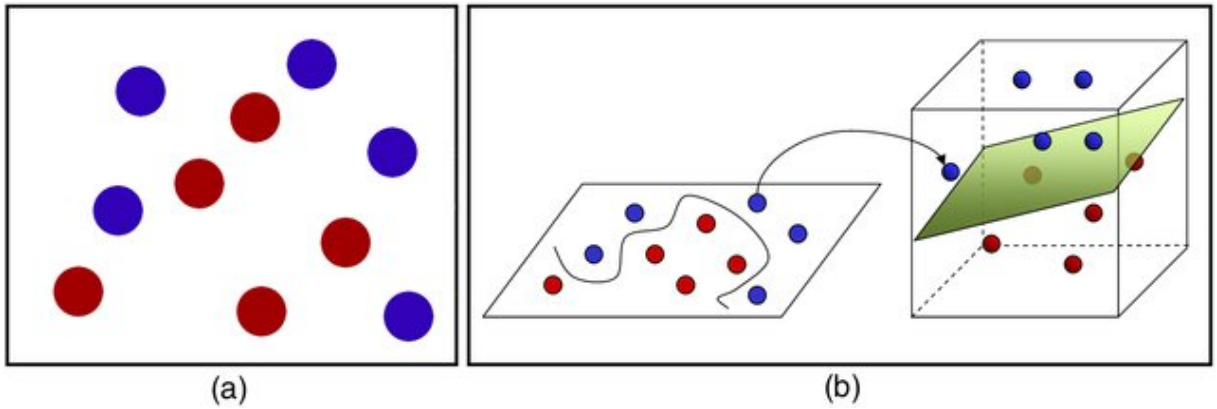


Figura 3 – Conjunto de dados não linearmente separável.
Fonte: (NORONHA; FERNANDES, 2016).

2.4 Regressão Logística

O modelo de aprendizado conhecido como Regressão Logística foi desenvolvido por David Cox (COX, 1958; BITTENCOURT, 2003). Ele é utilizado para prever um resultado que só tem duas possibilidades, como “sucesso ou falha”, ou “0 e 1”. Em outras palavras, o objetivo é estimar a probabilidade de que um determinado evento ocorra.

Para fazer isso, a Regressão Logística utiliza uma fórmula chamada função logística, que tem um formato em "S". Essa fórmula transforma os números em probabilidades que variam entre 0 e 1 (BITTENCOURT, 2003), como mostrado na Figura 4. Durante o treinamento, o modelo aprende a classificar uma amostra em uma das categorias. Se a probabilidade for maior que 0,5, o modelo prevê que o evento acontecerá (classe positiva). Caso contrário, ele prevê que não acontecerá (classe negativa) (GONZALEZ, 2018).

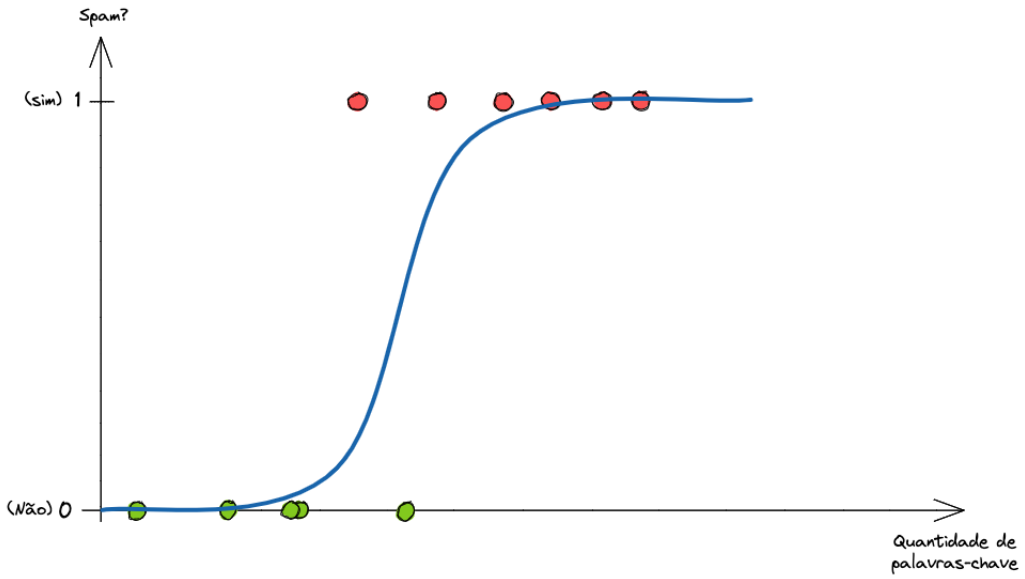


Figura 4 – Modelos de Classificação: Regressão Logística.
Fonte: (Brains.dev, 2023).

2.5 K-ésimo Vizinho mais Próximo

O algoritmo de AM KNN foi inicialmente proposto por Evelyn Fix e Joseph Hodges em 1951 (FIX, 1985). Esse modelo é particularmente útil, pois se inicia com um conjunto de dados de treinamento, que são formados por exemplos classificados em diversas categorias. As categorias são determinadas por meio de uma variável nominal, que atribui rótulos aos dados (LANTZ, 2019).

Um conjunto de dados de teste não rotulados, que possuem características equivalentes aos dados de treinamento, é utilizado para avaliar o modelo. Para cada entrada de dados, o algoritmo identifica "k" registros de treinamento que mais se assemelham aos padrões dos dados de treinamento, semelhante à Figura 5. Por fim, a classe não rotulada recebe o rótulo correspondente à maioria dos "k-vizinhos", conforme ilustrado na Figura 6.

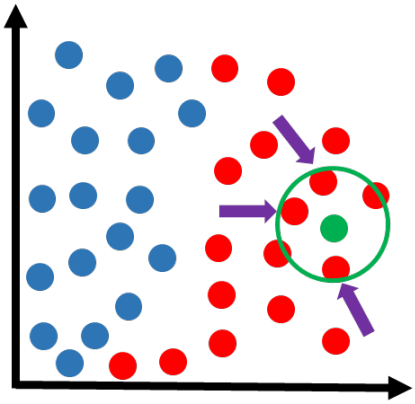


Figura 5 – Exemplo de classificação de uma amostra em um conjunto de dados, dividido em duas classes: azul e vermelho.
Fonte: (TECH, 2023).

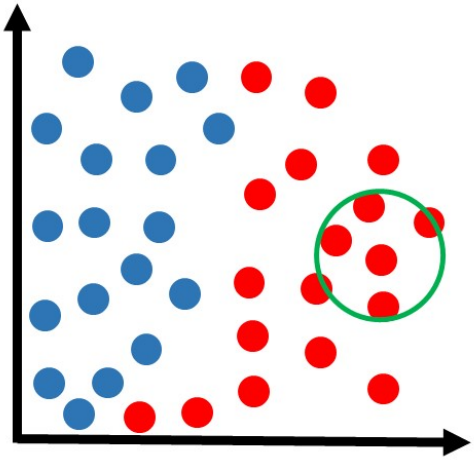


Figura 6 – Exemplo de classificação escolhida em um conjunto de dados, dividido em duas classes: azul e vermelho.
Fonte: (TECH, 2023).

2.6 Árvore de Decisão

A Árvore de Decisão, proposto por Quinlan (1986), recebe esse nome devido à sua estrutura que se assemelha ao formato de uma árvore. Assim sendo, o principal objetivo desse modelo é realizar decisões lógicas, simulando um processo de tomada de decisão sequencial. Em outras palavras, ele avalia se certas condições são atendidas para determinar a execução ou não de uma ação (LANTZ, 2019). Esse processo de tomada de decisão é possibilitado por um algoritmo de construção da árvore que define, em cada ponto de divisão (nó), qual atributo da base de dados será logicamente avaliado.

Para realizar previsões, a construção da árvore de decisão utiliza um método chamado particionamento recursivo. Esse processo é frequentemente descrito pela expressão

"dividir e conquistar", pois consiste em segmentar os dados em grupos com base nas características mais relevantes. A divisão continua de forma sucessiva até que se atinja um ponto final, conforme determinado por critérios específicos (LANTZ, 2019), tais como:

- ❏ A maioria dos dados no nó pertence à mesma classe.
- ❏ Não restam características para distinguir os dados.
- ❏ A árvore atingiu um tamanho máximo.

Uma vez construída, a árvore de decisão permite a classificação de novos dados de teste. Os dados são classificados a partir do nó raiz, e, à medida que são processados por meio de decisões fundamentadas em atributos, a árvore ramifica-se até que os dados alcancem as folhas, que representam as classes a serem previstas (LANTZ, 2019). Como ilustrado na Figura 7, as folhas estão denominadas como "Não vou" e "Eu vou".

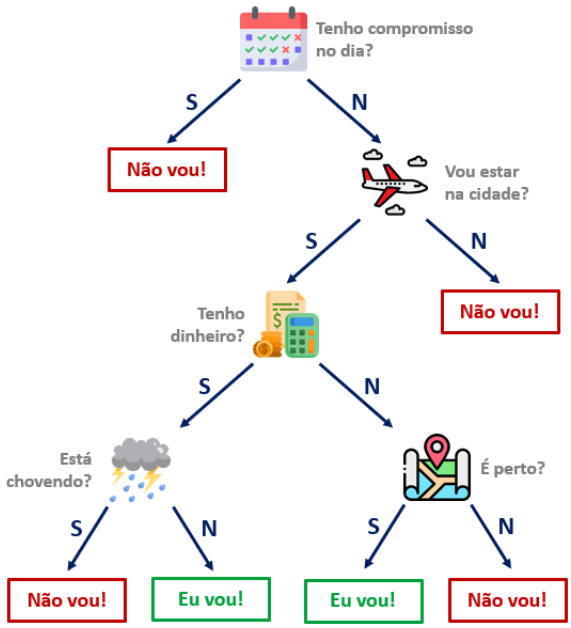


Figura 7 – Representação visual da árvore de decisão.
Fonte: (TREINAMENTOS, 2023).

2.7 Floresta Aleatória

Proposto em 2001 por Leo Breiman e Adele Cutler (BREIMAN, 2001), o modelo Floresta Aleatória utiliza um conjunto de árvores de decisão para realizar previsões de maneira eficiente e robusta (LANTZ, 2019).

Essas árvores de decisão são criadas pelo método conhecido como *bagging*. Nesse processo, várias árvores de decisão são treinadas utilizando subconjuntos aleatórios do conjunto de dados (com reposição), além da seleção aleatória de características, o que gera

diversidade entre as árvores. Para realizar a previsão final de um novo dado, a Floresta Aleatória utiliza a média das previsões no caso de regressão, ou o voto majoritário no caso de classificação (LANTZ, 2019). Esse processo é ilustrado na Figura 8.

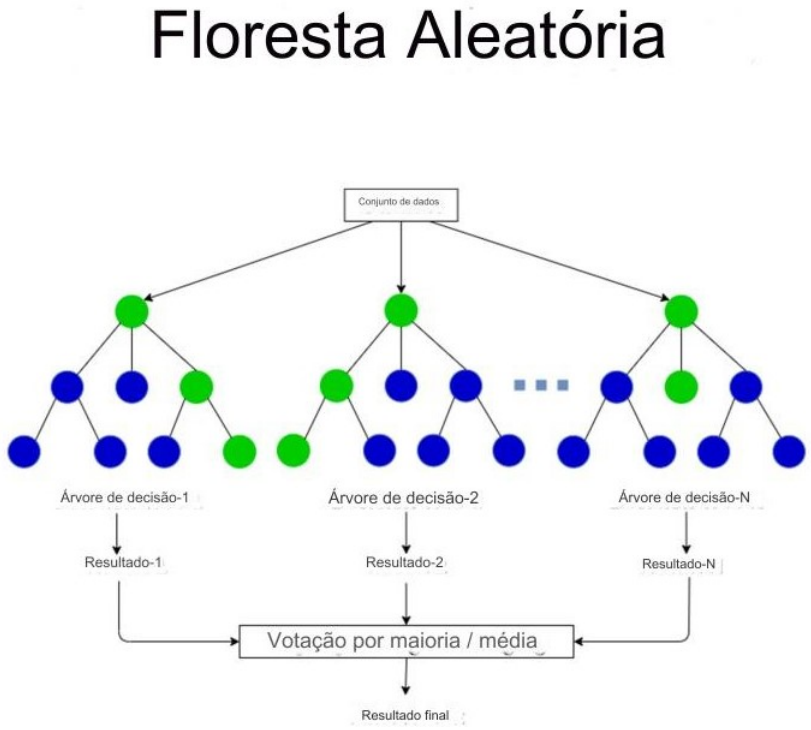


Figura 8 – Representação visual da Floresta Aleatória.
Fonte: Adaptado de (NEPAL et al., 2024).

2.8 XGBoost, LightGBM e CatBoost

Desenvolvido por Tianqi Chen e Carlos Guestrin na Universidade de Washington (CHEN; GUESTRIN, 2016), o XGBoost é um modelo de aprendizado de máquina fundamentado em árvores de decisão, pertencente à categoria de algoritmos de Boosting. Este modelo é projetado para aprimorar seu desempenho ao ajustar as árvores sequencialmente, visando corrigir os erros cometidos pelo modelo anterior (GOHIYA; LOHIYA; PATIDAR, 2018).

Embora o ajuste das árvores ocorra de maneira sequencial, o XGBoost implementa a paralelização da construção de uma árvore ao calcular ramos independentes, o que não interfere no processo de aprendizagem, mas contribui significativamente para a escalabilidade da solução (ZHANG et al., 2022). Essa escalabilidade é ainda favorecida por um dos mecanismos fundamentais do XGBoost: a técnica de "poda". Esse procedimento visa reduzir a complexidade do modelo, limitando tanto o número de nós folhas quanto os valores atribuídos a esses nós na estrutura da árvore (ZHANG et al., 2022).

Por outro lado, o LightGBM, desenvolvido pela Microsoft, é também um algoritmo de Boosting baseado em árvores de decisão, porém se distingue do XGBoost por suas abordagens técnicas inovadoras. Uma das principais inovações é a utilização da técnica de Amostragem unilateral baseada em gradiente (GOSS), que permite a seleção dos exemplos mais relevantes para o treinamento, priorizando as instâncias que apresentam gradientes mais altos. Essa estratégia não apenas acelera o processo de treinamento, mas também contribui para a melhoria da precisão do modelo (KE et al., 2017).

Outra inovação significativa é o Pacote de recursos exclusivos (EFB), que agrupa e combina características exclusivas, resultando na diminuição da dimensionalidade do conjunto de dados e na melhoria da eficiência do treinamento. Essa técnica se mostra especialmente eficaz em conjuntos de dados que contêm muitos recursos dispersos, levando a uma redução significativa no tempo de execução (KE et al., 2017).

Entretanto, o CatBoost, desenvolvido pela Yandex, é um modelo de Boosting baseado em árvores de decisão que se diferencia do XGBoost e do LightGBM por suas inovações específicas. Entre suas principais inovações, destaca-se a capacidade de lidar com variáveis categóricas sem a necessidade de transformá-las em numéricas, além de combinar essas variáveis para capturar dependências complexas. Essa abordagem melhora a capacidade de aprendizado, algo que é necessário em outros modelos, e também evita problemas de *leakage* (vazamento de informações), que podem ocorrer em modelos de aprendizado supervisionado (PROKHORENKOVA et al., 2018).

Para evitar o problema de vazamento, o CatBoost incorpora a técnica de *Ordered Boosting*, uma modificação do gradient boosting clássico. Nessa abordagem, o algoritmo utiliza permutações dos dados de treino para impedir que informações da variável alvo que o modelo busca prever e explicar influenciem indevidamente o processo de aprendizado. O objetivo é reduzir o deslocamento da predição causado pelo uso dos mesmos dados de treino em diferentes estágios do aprendizado (PROKHORENKOVA et al., 2018).

Dessa forma, tanto o XGBoost, quanto o LightGBM e o CatBoost são ferramentas poderosas no contexto de AM. Enquanto o XGBoost se destaca pela sua abordagem de ajuste sequencial e poda, o LightGBM se beneficia da amostragem baseada em gradientes e da combinação de recursos para otimizar a eficiência e escalabilidade. Por sua vez, o CatBoost inova ao lidar com variáveis categóricas sem a necessidade de transformação e ao empregar a técnica de *Ordered Boosting*, que previne o vazamento de informações, melhora a precisão e a robustez dos modelos.

2.9 Regressão: Linear simples, múltipla e polinomial

Os conceitos de regressão linear foi sugerido em 1984 por Francis Galton, os quais foram definidos como um teste matemático que visam prever e quantificar a relação entre diferentes variáveis (MONTGOMERY; PECK; VINING, 2021)(BELSLEY; KUH; WELSCH,

2005)(MAULUD; ABDULAZEEZ, 2020) (AKGÜN; ÖĞÜDÜCÜ, 2015) (DEHGHAN; HAMIDI; SALAJEGHEH, 2015) (MASOOD; ABDULAZEEZ; ZEEBAREE, 2020). Na literatura encontra-se a regressão linear simples, a múltipla e a polinomial que desempenham um papel importante, ao ajudarem a entender como diferentes fatores se relacionam. Esses modelos são a base para muitas abordagens do AM, permitindo que a análise preveja comportamentos de maneira mais eficaz.

A regressão linear simples, analisa a relação entre duas variáveis: uma que influencia e outra que é influenciada (MAULUD; ABDULAZEEZ, 2020). Por exemplo, a Figura 9 a variável que influencia são as horas de estudo, já a variável influenciada é a nota do aluno.

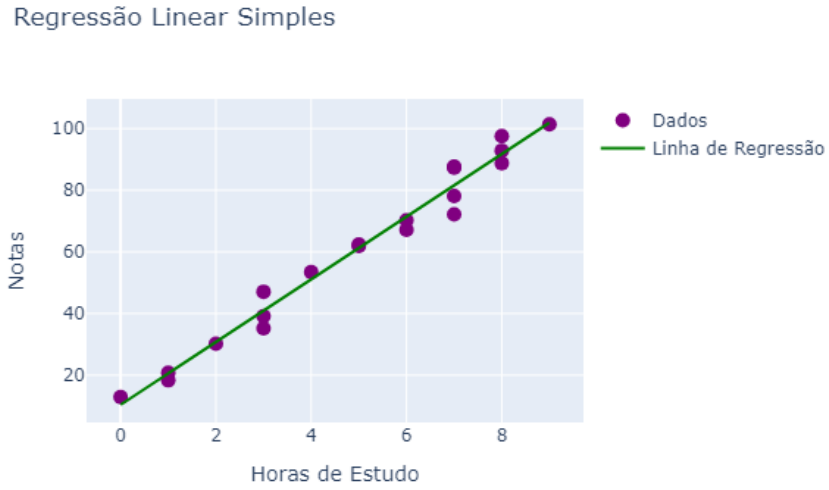


Figura 9 – Relação entre horas de estudo e notas.
Fonte: Autoria própria.

Ademais, a regressão linear múltipla expande a ideia da regressão linear simples, considerando várias variáveis dependentes (ZHANG et al., 2019). Por exemplo, além das horas de estudo, outro fator como frequência às aulas pode influenciar as notas. Esse modelo é útil para entender como diferentes fatores interagem entre si conforme exemplificado na Figura 10.

Por outro lado, a regressão polinomial (ROZIQIN; BASUKI; HARSONO, 2016)(PRASAD et al., 2015) é uma técnica de análise que tem em vista modelar a relação entre variáveis independentes e dependentes usando um polinômio de grau n (MAULUD; ABDULAZEEZ, 2020). Essa abordagem é especialmente útil quando a relação entre os dados é não-linear. Por exemplo, o crescimento de uma planta pode não seguir uma linha reta em diferentes estágios, e a regressão polinomial utiliza curvas para capturar essas interações mais complexas conforme exemplificado na Figura 11.

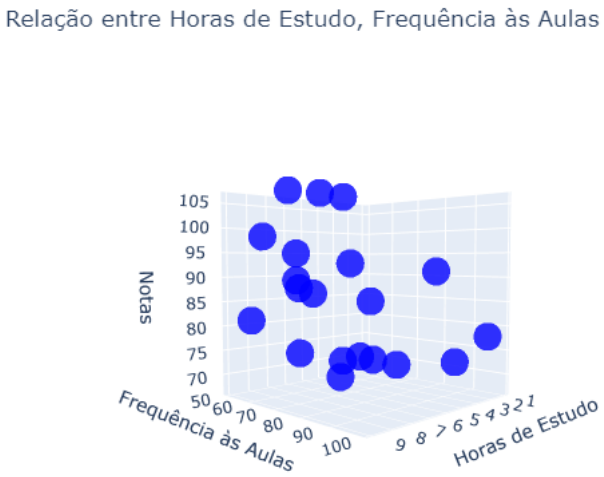


Figura 10 – Relação entre horas de estudo, frequência às aulas e notas.
Fonte: Autoria própria.

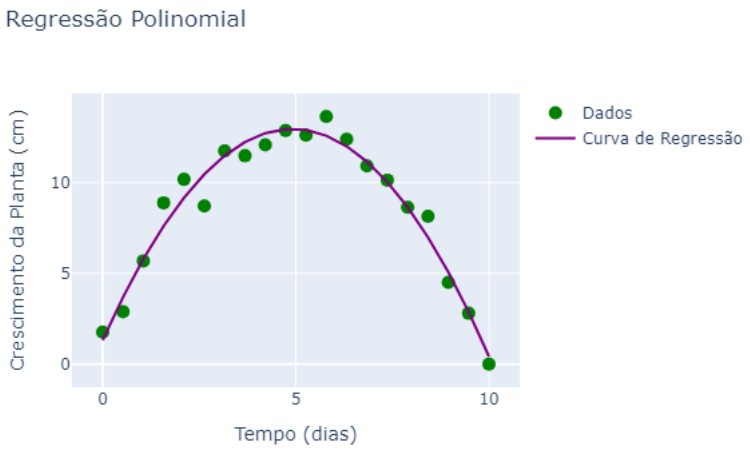


Figura 11 – Relação entre crescimento da planta e tempo.
Fonte: Autoria própria.

2.10 Análise Associativa

Análise de Associação é uma metodologia em AM que tem em vista descobrir relacionamentos importantes escondidos em grandes conjuntos de dados (TAN; STEINBACH; KUMAR, 2009). Esses relacionamentos são representados na forma de regras de associação que envolve uma quantidade aleatória de atributos presentes na base de dados. O algoritmo Apriori define o conjunto de regras para a identificação de padrões comuns em conjuntos de dados. Ele foi proposto em 1994 por Rakesh Agrawal e Ramakrishnan Srikant (AGRAWAL; IMIELIŃSKI; SWAMI, 1993).

Esse algoritmo teve origem ao analisar a venda de produtos em um supermercado que

tendem a ocorrer juntos em transações, buscando identificar regras do tipo “se-então”. Conforme ilustrado na Figura 12, se o cliente compra carvão, então existe uma grande chance percentual de também comprar carne, cerveja e sal grosso.

O algoritmo funciona de maneira interativa, buscando identificar conjuntos de itens frequentes. Inicialmente, ele analisa cada item individualmente, verificando se atende a uma frequência mínima conhecida como suporte, que pode ser expressa em termos percentuais ou fracionais. Em seguida, aplica o mesmo processo a pares, trios e conjuntos maiores, até que não se encontrem mais combinações frequentes. Esse algoritmo possui a propriedade chamada “anti-monotônica”, segundo a qual, se um conjunto de itens não satisfaz o suporte mínimo, nenhum dos seus superconjuntos será frequente. Essa característica permite eliminar combinações irrelevantes, aumentando a eficiência do processo.

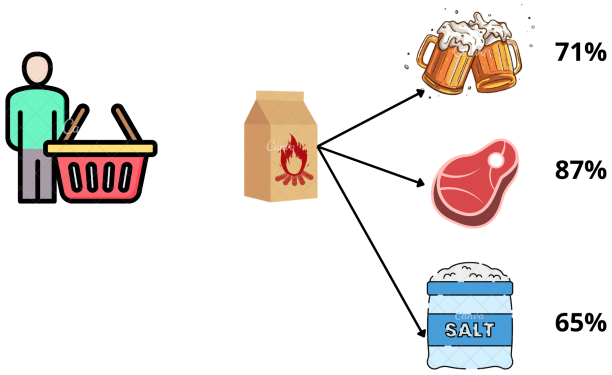


Figura 12 – Exemplo de regras de associação pelo algoritmo Apriori.
Fonte: Autoria própria.

2.11 Trabalhos Correlatos

2.11.1 Metodologias de Ensino para AM

Em seu trabalho, Carney et al. (2020) apresenta a plataforma web Teachable Machine, a qual permite a criação de modelos de AM para classificação de imagens, vídeos e sons. Com uma interface acessível, a plataforma possibilita que tanto crianças quanto adultos desenvolvam seus modelos de AM sem a necessidade de conhecimentos algorítmicos prévio ou especializado.

A ferramenta tem sido amplamente adotada por professores, alunos e designers em todo o mundo, sendo utilizada em mais de 201 países e resultando na criação de mais de 125.000 modelos até o ano de 2020. Um de seus públicos-alvo são os educadores, que utilizam a plataforma para ensinar conceitos de AM nas salas de aula, além de facilitar a

prototipagem, o trabalho de designers e a criação de soluções personalizadas para pessoas com deficiências.

Logo, o estudo de Carney et al. (2020) concluí que é importante destacar que o AM tem se mostrado útil para educadores que desejam apresentar conceitos de forma clara e direta, sem, no entanto, tratar o ensino de maneira superficial. Relatos de administradores, professores e alunos indicam uma escassez de ferramentas que auxiliem no ensino de AM. Nesse contexto, ferramentas web como a Teachable Machine contribuem para preencher essa lacuna e promovem a democratização do AM.

No trabalho de Martins et al. (2023), é descrita uma iniciativa para ensinar conceitos de AM a 158 alunos do ensino fundamental e médio, os quais possuem origens socioeconômicas desfavorecidas no Brasil. O projeto busca democratizar o acesso à IA, oferecendo a capacitação em modelos de classificação de imagens.

Este estudo apresenta o curso Aprendizado de Máquina Para Todos (ML4ALL), projetado para ensinar AM a alunos sem conhecimento prévio em computação. A metodologia do curso inclui o uso de slides interativos e demonstrações com objetos do cotidiano, que ilustram como um algoritmo de AM realiza a classificação. Além disso, são utilizadas ferramentas educacionais e atividades práticas, empregando a ferramenta Teachable Machine para a criação de modelos de classificação de imagens. Essa abordagem visa facilitar a aprendizagem de conceitos fundamentais de AM, como coleta de dados, treinamento de modelos e testes, permitindo uma interpretação eficaz dos resultados.

Os resultados confirmam o sucesso do ensino de AM aos alunos, independentemente do gênero, da etapa escolar ou do turno das aulas. Os estudantes demonstraram compreender as principais etapas da criação de um modelo de classificação de imagem. O destaque do curso é a percepção de que ele é motivador para os alunos, especialmente para as alunas, que manifestaram interesse em seguir carreira na área de Tecnologia da informação (TI). Portanto, o estudo demonstrou que estudantes de baixa renda podem aprender AM, desde que recebam o suporte adequado e utilizem ferramentas intuitivas para o primeiro contato com AM.

2.11.2 Comparativo entre diferentes metodologias para ensino de AM

A utilização de ferramentas com interface gráfica no auxílio do aprendizado de máquina tem sido amplamente discutida na literatura, especialmente no contexto de aprendizado para iniciantes. Conforme avaliado em Çağlayan (2019), foi realizada uma comparação entre o ensino de AM baseado em linhas de código e o baseado em interface gráfica, mais especificamente a ferramenta Orange[3], utilizada para implementar algoritmos de aprendizado de máquina e mineração de dados em bases de dados, visando obter previsões. Essa abordagem facilita a visualização dos algoritmos através da gamificação, que aplica

elementos de jogos para tornar a aprendizagem mais interativa e acessível, eliminando a necessidade de programar.

A hipótese formulada nesse estudo é que ferramentas visuais de ensino aumentam o interesse dos alunos no assunto mais do que o ensino baseado em código. No entanto, o ensino baseado em código proporciona uma melhor percepção de autoaprendizado, pois os alunos sentem que compreendem melhor o algoritmo ao vê-lo implementado no código.

Para testar a hipótese, o trabalho contou com a participação de onze voluntários do primeiro ano dos departamentos de engenharia de computação, software e sistemas de informação, que concluíram o curso de programação na linguagem C. O algoritmo AM implementado pelos participantes foi o KNN, devido à sua facilidade de compreensão e implementação para alunos iniciantes.

Dentre eles, cinco alunos, denominados "Grupo A", participaram do curso no qual o algoritmo KNN foi explicado via código, enquanto seis alunos, denominados "Grupo B", aprenderam através do software Orange[3].

Logo após, foi realizado o desenho experimental, buscando entender o conhecimento dos voluntários sobre o KNN nas etapas *PreTask* e *PostTask*, sendo coletado via questionário. O questionário *PreTask* avalia o conhecimento prévio dos alunos com cinco perguntas em uma escala Likert de cinco pontos, enquanto o *PostTask* mede suas percepções pós-aula e interesse no tema com onze perguntas, das quais as sete primeiras medem o interesse dos alunos também em uma escala Likert (LUNA, 2007) de cinco pontos.

A média dos resultados do nível de interesse foi de 4,08 para o Grupo A e 4,10 para o Grupo B. Portanto, o ensino baseado em ferramentas promove um maior interesse pelo aprendizado, pois os alunos se sentem mais confortáveis e evitam o preconceito em relação à dificuldade dos algoritmos de AM.

2.11.3 O Papel dos Recursos Educacionais Livres na Democratização da Tecnologia

Em seu trabalho, Bothmann et al. (2023) destaca que a comunidade de AM está profundamente alinhada aos princípios do código aberto, ressaltando a importância de disponibilizar Recursos Educacionais Abertos (REA) para todos que desejam aprender sobre AM e Ciência de Dados. Além disso, enfatiza a necessidade de compartilhar o material didático juntamente com o código-fonte, criando, desta forma, os Recursos Educacionais de Código Aberto (OSER).

O compartilhamento do código-fonte permite aprimorar a qualidade dos materiais e reduzir a carga de trabalho individual. Por meio da modularização dos OSER, é possível adaptar e personalizar os recursos de acordo com diferentes cenários, tornando o aprendizado mais acessível e eficaz (BOTHMANN et al., 2023).

No estudo realizado por Colvard, Watson e Park (2018), foram analisados os efeitos

da adoção de REA. O autor destacou que, especialmente em cursos de graduação com inúmeros alunos, essa adoção traz benefícios significativos para os professores e alunos. A pesquisa envolveu mais de 21.000 alunos e utilizou uma abordagem multinível para examinar o desempenho acadêmico dos estudantes antes e depois da implementação dos REA.

Os resultados mostraram que os impactos da adoção de REA vão além da economia financeira para os alunos ou da redução de dívidas estudantis. De maneira geral, os REA levaram a uma melhoria nas notas finais e a uma diminuição nas taxas de reprovação e desistência, especialmente entre os alunos de baixa renda, aqueles que recebem bolsas de estudo Pell Grant subsidiadas pelo estado, e estudantes pertencentes a grupos minoritários sub-representados.

Esses achados indicam que a utilização de REA ajuda a fechar lacunas no desempenho e na falta de conhecimento, proporcionando acesso a materiais gratuitos. Alunos que, anteriormente, poderiam ter optado por não adquirir os materiais necessários apresentaram um desempenho acadêmico melhor ao utilizar os REA.

2.12 Análise Comparativa dos Trabalhos Correlatos

Com base nas pesquisas correlatas, buscou-se compreender o estado da arte em metodologia de ensino de AM e o comparativo entre as diferentes metodologias de AM. A Tabela 1 apresenta informações relacionadas. A partir da Tabela 1, observa-se que todos os estudos (CARNEY et al., 2020; MARTINS et al., 2023; ÇAğLAYAN, 2019) empregam algoritmos de AM em suas pesquisas. A maioria dos trabalhos é voltada para adultos e crianças, com exceção de Çağlayan (2019), que restringe seu público-alvo ao contexto universitário. Com exceção do estudo de Çağlayan (2019), todos os trabalhos abordam o uso de interfaces gráficas web para o ensino de AM, transferindo o processamento de dados para o servidor, e não para a Unidade central de processamento (CPU) do usuário. Em contraste, o trabalho (ÇAğLAYAN, 2019) utiliza um software *stand-alone*, ou seja, pode ser executado de forma independente, sem a necessidade de conexão com outros sistemas ou servidores, sendo geralmente instalado diretamente no dispositivo do usuário. Vale ressaltar que todos os autores utilizam ferramentas de interface gráfica para o ensino de AM.

Trabalhos relacionados a REA (BOTHMANN et al., 2023; COLVARD; WATSON; PARK, 2018) complementam as pesquisas da Tabela 1, oferecendo novas abordagens para o uso de algoritmos de AM no ensino. Ao abordar a adoção de código livre e materiais educativos abertos, facilita o acesso ao conhecimento. Tornando o ensino inclusivo e acessível.

Com base na análise dos trabalhos correlatos, identificam-se lacunas ainda pouco exploradas, especialmente no que diz respeito ao desenvolvimento de uma interface que

Tabela 1 – Análise comparativa dos trabalhos correlatos

Trabalhos correlatos	**Algoritmos**	**Crianças e Adultos**	**Universitários**	**Plataforma Web**	**Plataforma Stand-Alone**	**Lingua Portuguesa**
(CARNEY et al., 2020)	✓	✓	✗	✓	✗	✗
(MARTINS et al., 2023)	✓	✓	✗	✓	✗	✗
(ÇAğLAYAN, 2019)	✓	✗	✓	✗	✓	✗
DreamLearning	✓	✗	✓	✓	✗	✓

implemente algoritmos de AM, voltada para alunos universitários, em uma plataforma web e com foco na língua portuguesa. Diante dessa carência, propõe-se o desenvolvimento de uma interface gráfica web intuitiva, direcionada ao ensino de AM, com o intuito de facilitar o aprendizado e a aplicação de algoritmos supervisionados e não supervisionados. A proposta também incorpora o conceito de REA, ao disponibilizar o código-fonte e vídeo aulas explicativas sobre os algoritmos, promovendo uma compreensão mais aprofundada, tanto teórica quanto prática, das técnicas de AM.

Capítulo 3

Plataforma Educacional Proposta

Neste capítulo, são detalhadas as etapas de desenvolvimento da ferramenta web DreamLearning, projetada para o ensino, a capacitação e o desenvolvimento de habilidades dos usuários em AM. Entre essas etapas, destacam-se a criação de diagramas de caso de uso e de fluxo de dados. Além disso, são apresentados os conceitos fundamentais relacionados à ferramenta, como treino e teste de modelos, hiperparâmetros, métricas de desempenho, validação cruzada, modelo cliente-servidor, além da descrição das ferramentas e tecnologias utilizadas. Por fim, são definidos os requisitos funcionais e não funcionais, com a especificação daqueles aplicados à plataforma desenvolvida.

3.1 Diagrama de Caso de Uso

Para ilustrar as principais funcionalidades do site DreamLearning criou-se um diagrama de caso de uso, o qual representa os usuários do sistema, conhecidos como atores, e suas interações com o sistema computacional proposto. Essa representação visa fornecer uma visão clara das ações e opções disponíveis para os usuários. As interações dos atores com o sistema DreamLearning é ilustrado na Figura 13.

Os atores presentes na Figura 13 são:

- **Usuário:** Um indivíduo que utiliza a plataforma para explorar os algoritmos de AM e realizar experimentos com diferentes bases de dados.
- **Administrador:** O administrador é responsável por gerenciar o sistema, incluindo a manutenção do site e o monitoramento da proteção do histórico de execuções. Apenas o administrador tem acesso ao banco de dados da plataforma DreamLearning e as chaves API, e também somente o administrador pode adicionar novos algoritmos à plataforma original.
- **Firebase:** Um serviço externo utilizado para autenticação e gerenciamento de dados no DreamLearning.

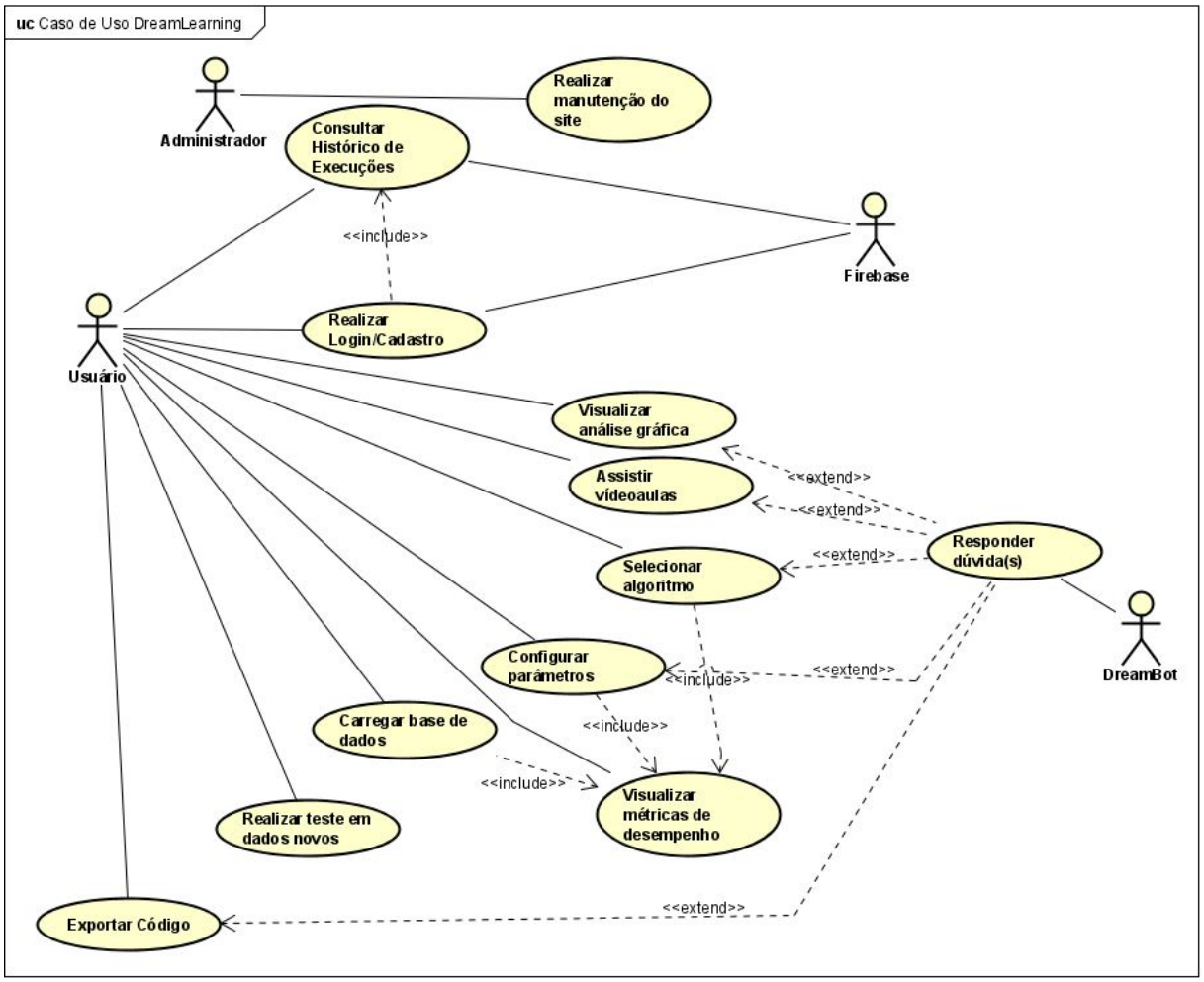


Figura 13 – Diagrama de caso de uso do DreamLearning.
Fonte: Autoria própria.

- **DreamBot:** Um serviço de chatbot foi desenvolvido utilizando a API do Gemini, a inteligência artificial generativa do Google.

Os principais Casos de Uso presentes na Figura 13 são:

- **Realizar Login/Cadastro:** O usuário precisa realizar o login para acessar o histórico de suas execuções na plataforma, utilizando o serviço de autenticação Firebase.
- **Consultar Histórico de Execuções:** O usuário pode acessar o histórico dos algoritmos que utilizou, incluindo informações detalhadas sobre os hiperparâmetros utilizados, métricas de desempenho do algoritmo e o código em Python do algoritmo de AM escolhido. Para visualizar esse histórico, o usuário deve estar autenticado, conforme indicado pela seta "«include»", o qual indica que o caso de uso "Consulta do histórico" está dependente do caso de uso de "Realizar Login/Cadastro".
- **Realizar Manutenção do Site:** O administrador tem por objetivo manter a infraestrutura e resolver eventuais problemas que aconteçam.

Ademais, segundo a Figura 13, as funcionalidades disponíveis no site DreamLearning para uso do usuário são:

- **Carregar base de dados:** O Usuário tem a possibilidade de fazer *upload* de arquivos no formato Valores separados por vírgula (CSV) para aplicar em diferentes algoritmos de AM. Quando o algoritmo selecionado for do tipo supervisionado, é importante que o arquivo esteja organizado de forma que a última coluna contenha as categorias ou respostas corretas (rótulos), que serão usadas pelo sistema para treinar e avaliar os modelos.

- **Selecionar algoritmo:** O usuário pode escolher entre os algoritmos de AM disponíveis na plataforma.

- **Configurar hiperparâmetros:** O usuário pode ajustar os hiperparâmetros do algoritmo para otimizar o desempenho.

- **Visualizar desempenho:** Após o processamento e classificação dos dados pelo algoritmo selecionado pelo usuário, é possível visualizar as métricas de desempenho obtidas do treino e teste. Conforme indicado pela seta "«include»", o qual indica que o caso de uso de "Visualizar resultados" está dependente dos casos "carregar a base de dados", "selecionar o algoritmo" e "configurar os hiperparâmetros".

- **Realizar teste em dados novos:** Após treinar e testar a base de dados, o usuário pode escolher testar novos dados, também no formato CSV, para verificar o desempenho do modelo. Quando o algoritmo selecionado for do tipo supervisionado, esse novo arquivo deve estar organizado de forma que não contenha os rótulos, permitindo ao sistema realizar previsões com base no modelo treinado e apresentar os resultados ao usuário.

- **Exportar código:** Após a execução do algoritmo selecionado, o usuário pode exportar o código do algoritmo de AM gerado para utilizar externamente.

- **Visualizar análise gráfica:** O usuário pode visualizar os dados do arquivo CSV de forma gráfica, facilitando a interpretação dos dados.

- **Assistir videoaulas:** Os usuários podem acessar, na ferramenta DreamLearning, o conteúdo educacional disponível, focado em AM, por meio de videoaulas, para auxiliar na compreensão dos conceitos de cada algoritmo de AM.

- **Responder dúvida(s):** O usuário tem a possibilidade de esclarecer dúvidas relacionadas à análise gráfica da base de dados selecionada, obter informações sobre o funcionamento de cada algoritmo de Aprendizado de AM da plataforma DreamLearning, compreender a finalidade de cada hiperparâmetro dos algoritmos e receber

orientações sobre quais hiperparâmetros utilizar. Além disso, por meio do DreamBot, o usuário pode adquirir um conhecimento prévio ou complementar antes ou após a videoaula referente ao algoritmo em estudo. Também é possível solicitar, ao exportar o código, uma explicação detalhada linha a linha sobre as operações realizadas no programa.

3.2 Diagrama de Fluxo de Dados

Para ilustrar o funcionamento do site DreamLearning, foi criado um diagrama de fluxo de dados, cujo objetivo é representar as principais etapas e processos do sistema computacional proposto. Essa representação visa fornecer uma visão clara do fluxo de ações e das opções disponíveis para os usuários, destacando a sequência lógica das interações no sistema. O diagrama de fluxo do sistema DreamLearning é apresentado na Figura 14.

- **Usuário acessa o site:**
  - Inicialmente, o usuário acessa o site DreamLearning.
- **Selecionar a página?:**
  - O usuário escolhe a página do site que deseja acessar. As opções incluem "Classificação", "Regressão", "Análise Associativa", "Análise Gráfica", "Vídeo Aulas" e "Histórico de Execuções".
- **Classificação:**
  - Caso o usuário opte pela página "Classificação", ele é direcionado para a devida página onde selecionará um arquivo CSV. É importante que esse arquivo esteja organizado de forma que a coluna de rótulo, ou seja, a que contém as respostas corretas seja a última coluna do arquivo.
  - Em seguida, o usuário escolherá o o algoritimo de classificação.
  - O usuário tem a opção de consultar, por meio do DreamBot, a funcionalidade de cada hiperparâmetro do algoritmo selecionado, bem como as opções disponíveis para esse algoritmo.
  - Em seguida, o usuário escolherá os hiperparâmetros para configurar o modelo de classificação ao qual será treinado e testado com base nos dados do arquivo CSV.
  - Após o treino e teste do modelo, as métricas de desempenho são exibidas.
  - O usuário tem a opção de realizar a validação cruzada e verificar a capacidade de generalização do modelo.

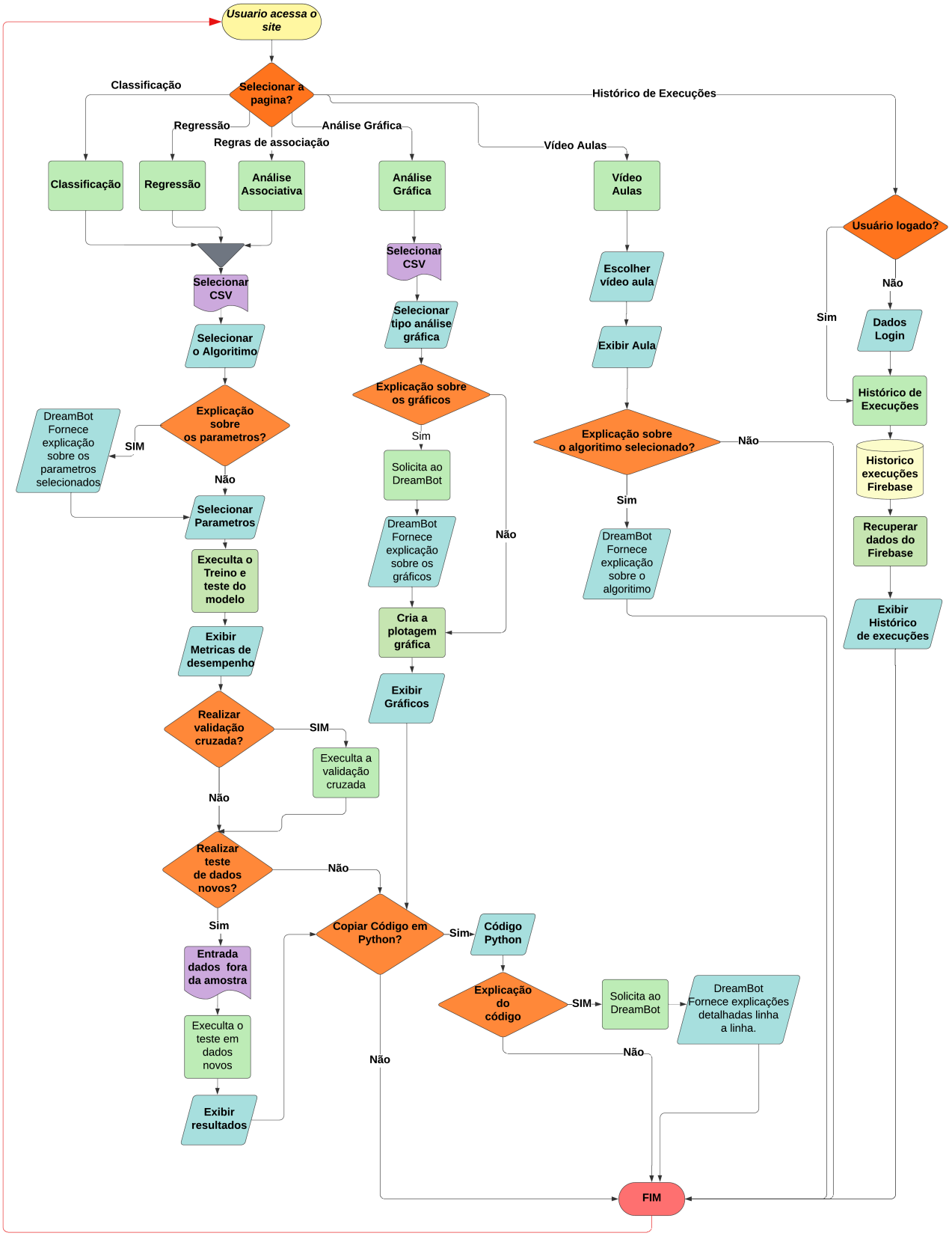


Figura 14 – Diagrama de fluxos do DreamLearning.
Fonte: Autoria própria.

- O usuário pode testar dados novos, que não foram usados no treinamento e teste, enviando um arquivo no formato CSV e obtendo os resultados.
- O usuário tem a opção de copiar o código Python para reproduzir o modelo e, também, solicitar ao DreamBot uma explicação detalhada linha a linha do código, para esclarecer dúvidas.

❑ **Regressão:**

- Similar à "Classificação", o usuário escolhe um arquivo CSV o algoritimo e define os hiperparâmetros para o modelo de regressão escolhido.
- Após o treinamento e teste, as métricas de desempenho são apresentadas, e o usuário pode optar por utilizar a validação cruzada, testar novos dados, copiar o código Python gerado e esclarecer dúvidas sobre o código com o DreamBot.

❑ **Análise Associativa:**

- Neste caso o usuário também escolhe um arquivo CSV. Os passos subsequentes serão semelhantes à "Classificação".
- Assim como nos modelos anteriores, o usuário pode optar por realizar a validação cruzada, testar novos dados, copiar o código gerado e esclarecer dúvidas sobre o código.

❑ **Análise Gráfica:**

- Neste caso, o usuário também seleciona um arquivo CSV e define entre a análise gráfica de todas as variáveis ou a correlação entre duas variáveis, além de escolher o tipo de gráfico a ser utilizado.
- Em seguida, o usuário pode solicitar ao DreamBot uma explicação sobre o funcionamento de cada tipo de gráfico.
- A ferramenta cria automaticamente a plotagem gráfica, exibindo os gráficos.
- Assim como nos passos anteriores, o usuário pode optar por copiar o código gerado e esclarecer dúvidas sobre o código.

❑ **Vídeo Aulas:**

- Nesta página o usuário escolhe uma vídeo aula para assistir, e o vídeo é apresentado no próprio site.
- Em seguida, o usuário pode solicitar ao DreamBot uma explicação sobre os conhecimentos prévios necessários para aprender o algoritmo ou esclarecer dúvidas sobre o algoritmo apresentado na videoaula.

❑ **Histórico de Execuções:**

- O usuário deve estar logado no sistema, caso contrário será necessário realizar o login para acessar a página "Histórico de Execuções".
- Com o usuário autenticado é possível acessar o histórico de execuções, armazenados no banco de dados Firebase.
- Nessa etapa, a ferramenta recupera os dados de execuções anteriores e os exibe, permitindo ao usuário observar a base de dados utilizada, o algoritmo empregado, os hiperparâmetros, as métricas de desempenho, os resultados e o código-fonte gerado.

❑ **Fim:**

- Essa etapa finaliza o fluxo e decreta o fim das interações possíveis dos usuários.

3.3 Metodologia e Funções do Site DreamLearning

Dentre os diagramas de caso de uso e o de fluxo de dados descritos nas Figuras 13 e 14, respectivamente, citou-se algumas funções importantes para o funcionamento do site. Logo, nesta seção apresenta-se conceitos que deixarão mais compreensível e completo a descrição das funcionalidades do site.

3.3.1 Treino e Teste do Modelo

No contexto de AM, o conjunto de treino é uma amostra da base de dados destinada ao treinamento de um modelo, o qual irá aprender a identificar padrões ou relações entre variáveis, adquirindo uma capacidade preditiva fundamentada em tais padrões.

Semelhante ao treino, a fase de teste é uma porção de dados, que não foi utilizada durante o treinamento, e será exposta ao modelo. O principal objetivo desta fase é verificar a capacidade adquirida pelo modelo em realizar as previsões precisas com novos dados. Nesta fase, preocupa-se em identificar o *overfitting*, que ocorre quando o modelo "memoriza" detalhes específicos dos dados de treino, tornando-se "viciado" em pontos que não são úteis para novos dados. Assim, embora o modelo tenha um bom desempenho durante o treinamento, ele enfrenta dificuldades em classificar novos dados, pois se concentra em detalhes específicos em vez de generalizar as informações. Outro problema que pode ocorrer é o *underfitting*, quando o modelo não aprende o "suficiente", resultando em uma baixa precisão em classificar novos dados.

3.3.2 Hiperparâmetros

Os hiperparâmetros de um modelo são variáveis ajustáveis que visam maximizar a precisão durante o treinamento. O ajuste dos hiperparâmetros influencia diretamente

no desempenho do modelo, pois quando está otimizado, gera um equilíbrio entre precisão e generalização dos dados. Por exemplo, em um modelo de classificação, a taxa de aprendizado é um hiperparâmetro que determina quão rapidamente o modelo se ajusta durante o treinamento. Se a taxa de aprendizado for muito alta, o modelo pode perder a capacidade de aprender, enquanto uma taxa muito baixa pode levar a um treinamento excessivamente longo e ineficiente.

3.3.3 Métricas de Desempenho

Para avaliar o desempenho do algoritmo de AM escolhido, são utilizadas métricas aplicadas aos resultados obtidos a partir da base de dados de teste. As métricas a serem calculadas e analisadas são:

- **Acurácia:** É uma métrica de avaliação que mede o desempenho de um modelo em relação à proporção de previsões corretas em relação ao total de previsões realizadas. No entanto, em cenários onde uma classe pode dominar a distribuição dos dados, é essencial considerar outras métricas de avaliação para obter uma visão mais abrangente do desempenho do modelo (STRAUSS; JÚNIOR; FERREIRA, 2022).
- **Precisão:** Avalia o percentual de previsões corretas entre todas as previsões que foram positivas. Essa métrica tem seu destaque em contextos onde o custo de falsos positivos é elevado (STRAUSS; JÚNIOR; FERREIRA, 2022).
- ***Recall:*** Avalia o percentual de verdadeiros positivos identificados em relação ao total de casos reais positivos. Essa métrica tem seu destaque em identificar o máximo possível de instâncias positivas, minimizando o custo de falsos negativos (STRAUSS; JÚNIOR; FERREIRA, 2022).
- ***F1-Score:*** É a média harmônica entre precisão e *recall*, proporcionando um equilíbrio entre as duas métricas (STRAUSS; JÚNIOR; FERREIRA, 2022).

3.3.4 Validação Cruzada

A validação cruzada é uma metodologia que segmenta o conjunto de dados em múltiplos subconjuntos, permitindo que o modelo de AM seja avaliado em diferentes amostras de treino e teste (KOHAVI et al., 1995), em uma quantidade de ciclos pré-definidos.

Dessa forma, a validação cruzada possibilita a realização de múltiplos testes em distintas partes dos dados, calculando a média aritmética da acurácia dos diversos modelos treinados. Esse procedimento permite avaliar a capacidade de generalização do modelo em relação à base de dados (KOHAVI et al., 1995).

Caso a acurácia média obtida pela validação cruzada apresente uma discrepância significativa em relação à acurácia do modelo treinado anteriormente, isso pode indicar que

o modelo está apenas memorizando os exemplos de treino, em vez de aprender padrões gerais.

3.3.5 Modelo Cliente-Servidor

Neste trabalho, foi adotada a arquitetura cliente-servidor, na qual o servidor tem a função de hospedar a aplicação desenvolvida. O cliente, representado pelo dispositivo do usuário, acessa o site por meio de um navegador de internet, como o Google Chrome ou o Firefox. Essa abordagem possibilita a interação do usuário com a aplicação por meio de uma interface gráfica on-line, permitindo a realização de diversas operações em tempo real.

Na referida arquitetura, ilustrada na Figura 15, o cliente envia solicitações ao servidor e aguarda uma resposta. Para essa comunicação é utilizado o Protocolo de Transferência de Hipertexto (HTTP), o qual estabelece regras de troca de informações entre cliente e servidor. O servidor, por sua vez, processa a solicitação recebida da rede, executa-a e retorna o resultado ao solicitante (LIZAMA et al., 2016).

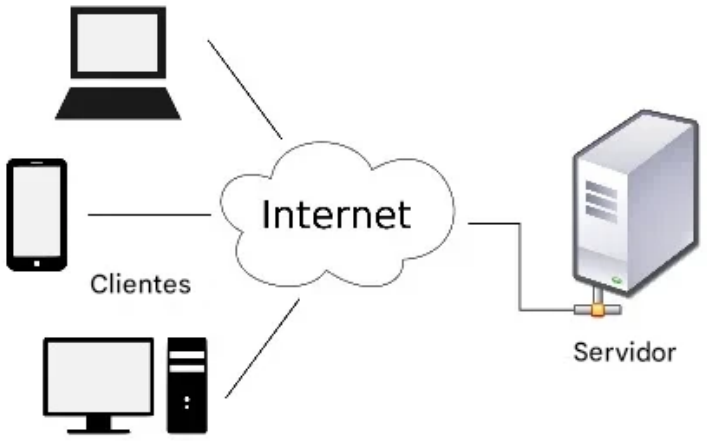


Figura 15 – Representação arquitetura cliente-servidor.
Fonte: Adaptado de (SAMPAIO, 2024).

3.3.6 Ferramentas e Tecnologias Utilizadas

As ferramentas tecnológicas de *front-end*, as quais os usuários interagem diretamente no site, foram utilizadas para a construção de uma interface, *layout* e design amigável para a interação com os algoritmos de AM. Por outro lado, para o *back-end*, que gerencia as requisições do lado do servidor, foram empregadas soluções que garantem a eficiência no processamento dos dados.

3.3.6.1 Front-end

- ❑ **Framework Web Bootstrap:** Possui o foco no *front-end*, para facilitar o desenvolvimento de interfaces web responsivas e móveis.
- ❑ **Framework Web NES.css:** Este framework é utilizado para a estilização do site, com ênfase na estética inspirada em jogos 8-bit.
- ❑ **JavaScript:** Linguagem de programação utilizada para enviar requisições de processamento ao servidor.
- ❑ **Linguagem de Marcação de HiperTexto (HTML):** Seu objetivo é gerar uma estrutura para as páginas Web.
- ❑ **Linguagem de estilização Folhas de Estilo em Cascata (CSS):** Tem como objetivo definir a aparência visual das páginas web, determinando características como cor, largura, altura e outros aspectos exibidos nos elementos da página.

3.3.6.2 Back-end

- ❑ **Python:** Linguagem de programação de alto nível na versão 3.12.4, sendo a linguagem mais recomendada para a ciência de dados (Awari, 2022), com uma sintaxe simplificada e de fácil legibilidade.
- ❑ **Framework Web Python Flask:** Permite a criação de aplicações web de forma simples e rápida. Gerenciando de maneira eficiente as requisições do cliente e facilitando a comunicação entre o *front-end* e o *back-end*.
- ❑ **Algoritmos de AM:** Foram utilizados os seguintes algoritmos: Naive Bayes, SVM, Regressão Logística, KNN, Árvore de Decisão, Floresta Aleatória, XGBoost, LightGBM, CatBoost, Regressão Linear Simples, Regressão Múltipla, Regressão Polinomial e as Regras de Associação Apriori.

3.3.7 Requisitos do Sistema

Para a criação da ferramenta DreamLearning, foi estabelecido um conjunto de requisitos que asseguram sua funcionalidade, usabilidade e eficiência. O objetivo é criar uma plataforma intuitiva que facilite o uso do AM, promovendo o aprendizado contínuo e a prática dessas técnicas.

3.3.7.1 Funcionais

Os requisitos funcionais descrevem as funcionalidades do sistema, como o sistema deve reagir a entradas específicas, seu comportamento em determinadas situações e o que

não deve ser feito por esse sistema (FIGUEIREDO, 2011). Os requisitos funcionais da ferramenta são:

- ❑ **Autenticação:**
  - O sistema deve permitir o login/cadastro dos usuários que desejam acessar o histórico de treinamentos.
- ❑ **Acesso ao Histórico de Execuções:**
  - O sistema deve registrar todas as execuções feitas pelo usuário, possibilitando o acesso a um histórico detalhado das operações realizadas em diferentes bases de dados, incluindo o nome da base de dados, o algoritmo utilizado, os hiperparâmetros configurados, as métricas de desempenho e o código-fonte do algoritmo em Python gerado.
- ❑ **Exportação da base de dados:**
  - Deve ser permitido o usuário exportar a base de dados no formato CSV.
- ❑ **Interface de Seleção e Aplicação de Algoritmos:**
  - O sistema deve permitir que o usuário selecione uma base de dados a sua escolha e aplique algoritmos, como Random Forest, SVM e Regressão Linear Simples.
- ❑ **Configuração de Hiperparâmetros:**
  - O sistema deve permitir a configuração de hiperparâmetros específicos de cada algoritmo, como número de árvores para Random Forest ou taxa de aprendizado para o XGBoost.
- ❑ **Exibição de Resultados:**
  - Deve-se exibir as métricas de desempenho após o treino e teste dos algoritmos de AM, como: acurácia, *recall*, precisão e *F1-score*. E também o resultado da validação cruzada.
- ❑ **Exportação do Código-fonte:**
  - O sistema deve permitir que o usuário copie o código-fonte após o treino e teste dos algoritmos de AM, incluindo os hiperparâmetros previamente configurados, para execução local em seu ambiente Python.
- ❑ **Análise Gráfica das Bases de Dados:**

- O sistema deve disponibilizar uma ferramenta de visualização gráfica que permita aos usuários analisar todas as variáveis ou a análise comparativa entre duas variáveis, facilitando assim a interpretação das variáveis e dos resultados obtidos pelos algoritmos.

❑ **Acesso a Videoaulas:**

- O sistema deve possibilitar o acesso dos usuários a videoaulas sobre os algoritmos de AM disponíveis na ferramenta.

❑ **Acesso ao DreamBot:**

- O sistema deve possibilitar o acesso dos usuários ao chatbot denominado "DreamBot".

3.3.7.2 Não Funcionais

Os requisitos não funcionais podem ser mais críticos que os requisitos funcionais, definindo as propriedades e restrições do sistema, como segurança, armazenamento e desempenho (FIGUEIREDO, 2011). Os requisitos não funcionais da ferramenta são:

❑ **Desempenho:**

- O sistema deve conseguir processar a aplicação dos algoritmos em diferentes bases de dados em até 20 segundos, independente do tamanho da base de dados.
- A operação de autenticação deve ser realizado em menos de 5 segundos.

❑ **Segurança:**

- O sistema deve proteger os dados do usuário e o histórico de execuções.

❑ **Escalabilidade:**

- O sistema deve ser escalável e elástico para suportar execuções simultâneas em diferentes bases de dados e para diferentes usuários.

❑ **Portabilidade:**

- O sistema deve ser acessível em diferentes dispositivos, como computadores, smartphones e tablets.

❑ **Manutenibilidade:**

- O código deve seguir boas práticas de modularidade, permitindo a fácil manutenção do código e adição de novas funcionalidades.

3.3.8 Telas do Sistema Propostos

A seguir, são apresentadas as telas do site DreamLearning, incluindo a tela inicial, as telas de algoritmos de classificação e regressão, a de análise gráfica, a de análise associativa e a de histórico de execuções.

3.3.8.1 Inicial

A tela inicial do DreamLearning, apresentada na Figura 16, representa o primeiro contato visual do usuário com a plataforma e foi projetada para oferecer uma navegação prática e direta. Essa tela serve como ponto de partida para acessar as principais funcionalidades do site, especialmente o histórico de execuções. No entanto, por questões de segurança e privacidade, essa interface inicial só estará disponível após o usuário realizar o login ou criar uma conta na plataforma. Somente após esse processo de autenticação será possível visualizar as informações do histórico, permitindo ao usuário acompanhar as execuções já realizadas.

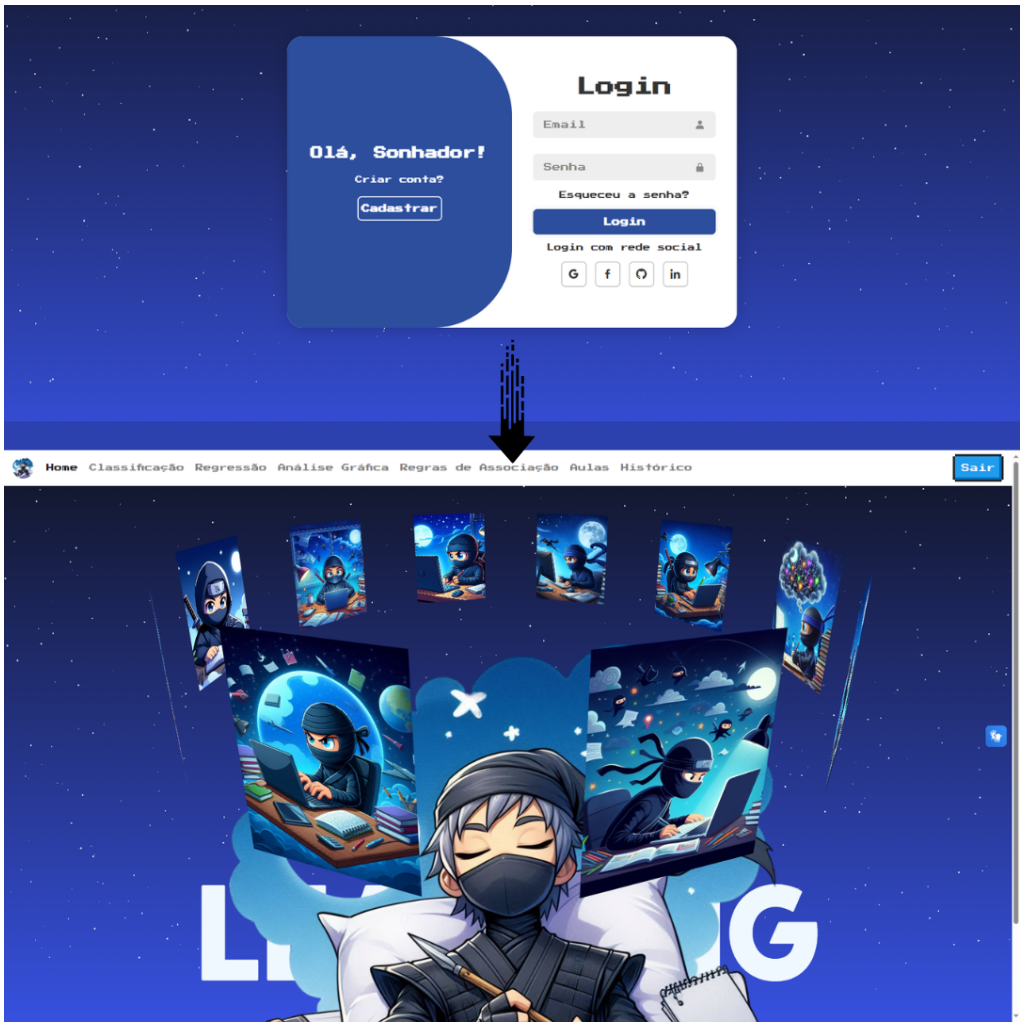


Figura 16 – Tela login do DreamLearning.
Fonte: Autoria própria.

3.3.8.2 Classificação

A tela de Classificação disponibiliza os seguintes algoritmos de classificação: Árvore de Decisão, Floresta Aleatória, SVM, Regressão Logística, KNN, XGBoost, LightGBM e CatBoost. A seguir, são apresentados os passos para classificar os dados.

- ❑ **1 passo:** No primeiro passo o usuário deve escolher o algoritmo de classificação a sua escolha, como ilustrado na Figura 17.

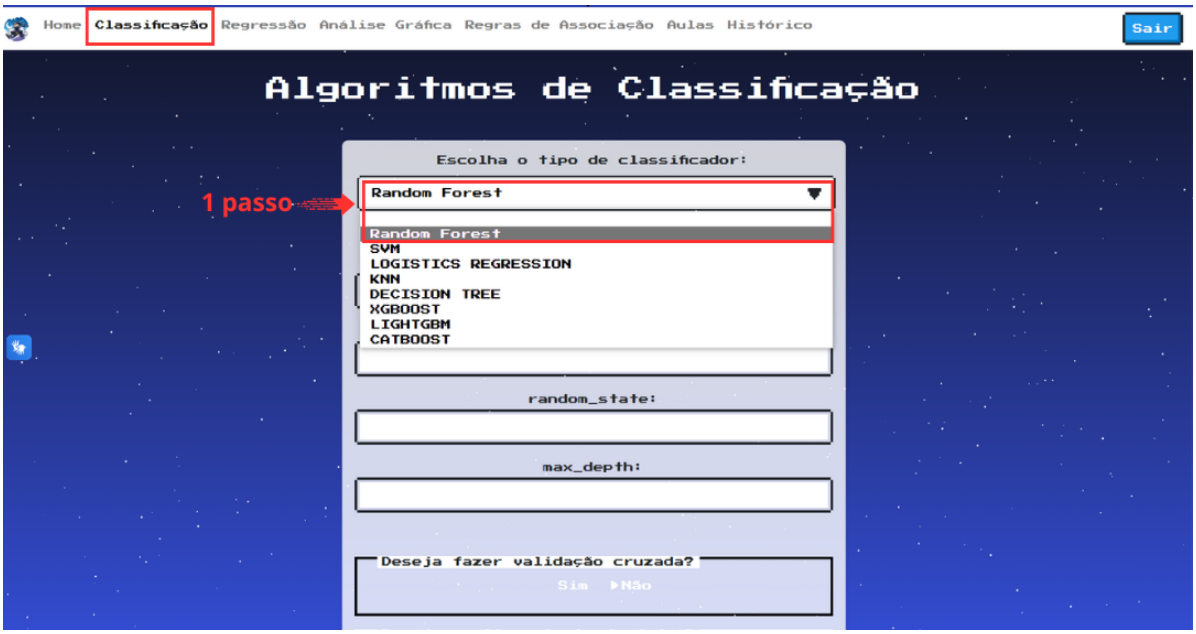


Figura 17 – Tela classificação: Etapa de seleção do algoritmo.
Fonte: Autoria própria.

- ❑ **2 passo:** No segundo passo, o usuário deve clicar no botão "Selecionar o arquivo" para escolher um arquivo CSV a ser utilizado na classificação, como ilustrado na Figura 18.

- ❑ **3 passo:** No terceiro passo, o usuário deve selecionar o arquivo CSV desejado em seu gerenciador de arquivos, como ilustrado na Figura 18.

- ❑ **4 passo:** No quarto passo, o usuário deve clicar no botão "Abrir" para enviar o arquivo CSV para o site, conforme ilustrado na Figura 18.

- ❑ **5 passo:** No quinto passo, o usuário deve definir os valores que serão utilizados nos hiperparâmetros, conforme ilustrado na Figura 19.

- ❑ **6 passo:** No sexto passo, o usuário pode escolher se deseja realizar a validação cruzada dos dados, selecionando a opção "Sim" ou "Não", conforme ilustrado na Figura 19.

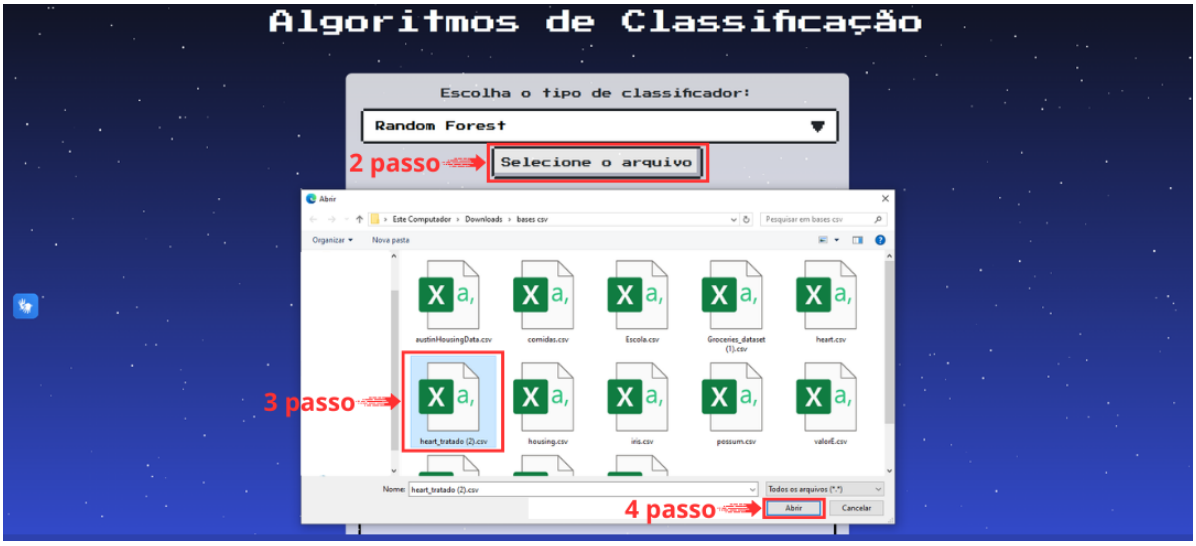


Figura 18 – Tela classificação: Etapa de seleção do arquivo CSV.
Fonte: Autoria própria.

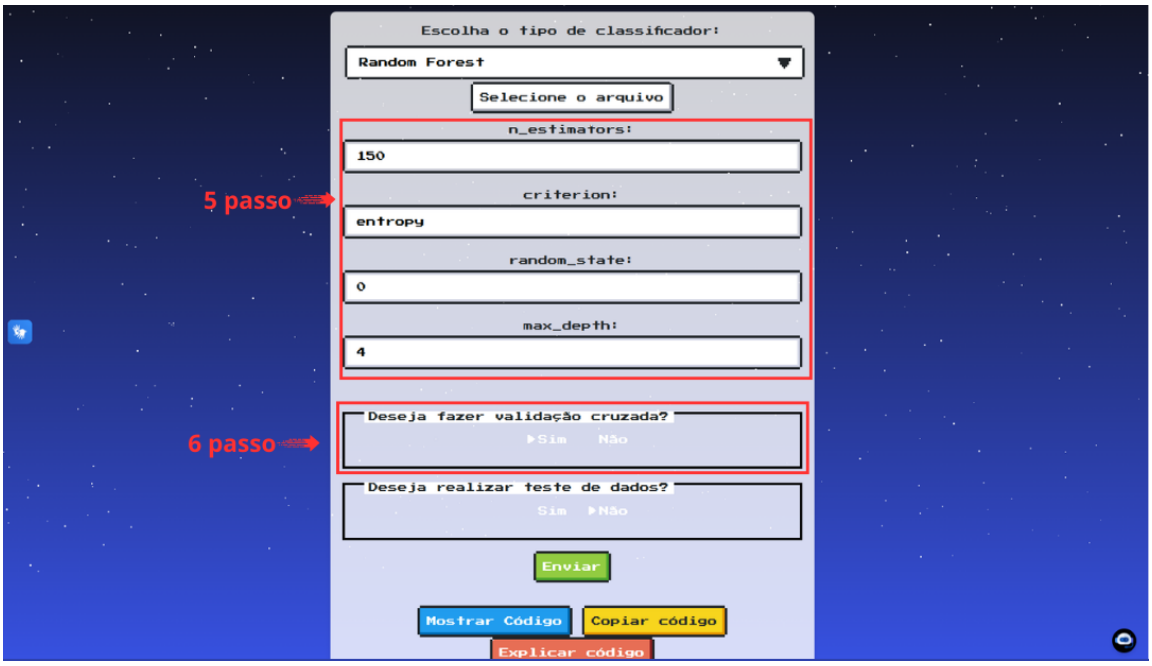


Figura 19 – Tela classificação: Etapas de seleção dos hiperparâmetros e validação Cruzada.
Fonte: Autoria própria.

- **7 passo:** No sétimo passo, o usuário pode escolher se deseja realizar o teste em dados diferentes dos dados de treino e teste, selecionando a opção "Sim" ou "Não", conforme ilustrado na Figura 20.

- **8 passo:** No oitavo passo, caso o usuário tenha escolhido "Realizar teste de dados" no sétimo passo, ele deve clicar no botão "Selecionar o arquivo para teste", conforme ilustrado na Figura 20.

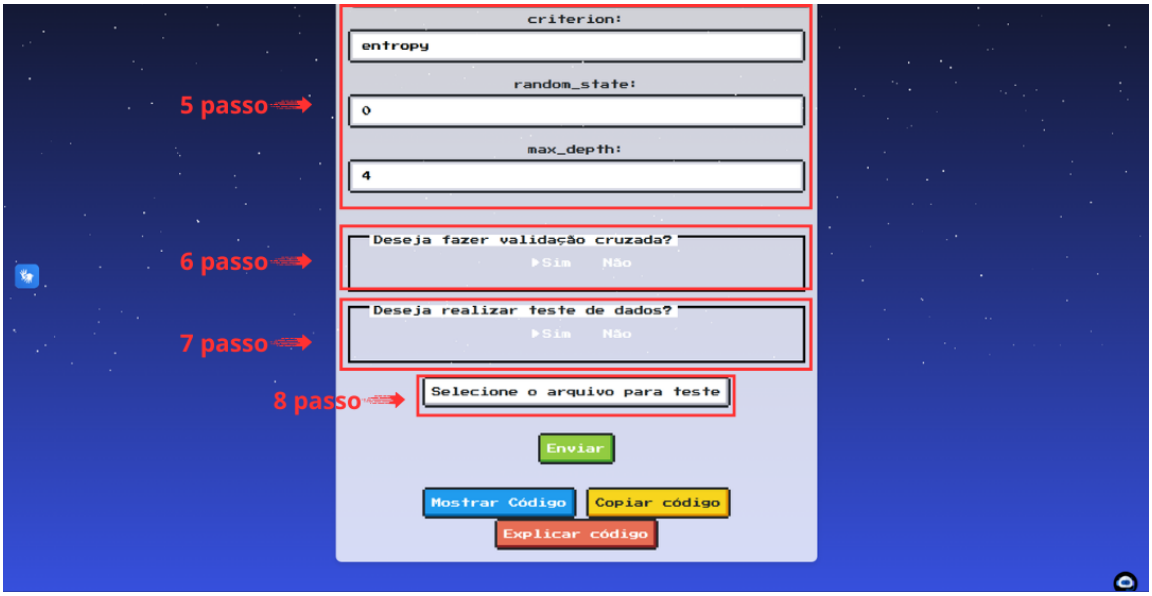


Figura 20 – Tela Classificação: Etapa de teste de dados.
Fonte: Autoria própria.

- **9 passo:** No nono passo, o usuário deve selecionar o arquivo CSV desejado para realizar teste de dados em seu gerenciador de arquivos, conforme ilustrado na Figura 21.

- **10 passo:** No décimo passo, o usuário deve clicar no botão "Abrir" para enviar o arquivo CSV para o site, conforme ilustrado na Figura 21.

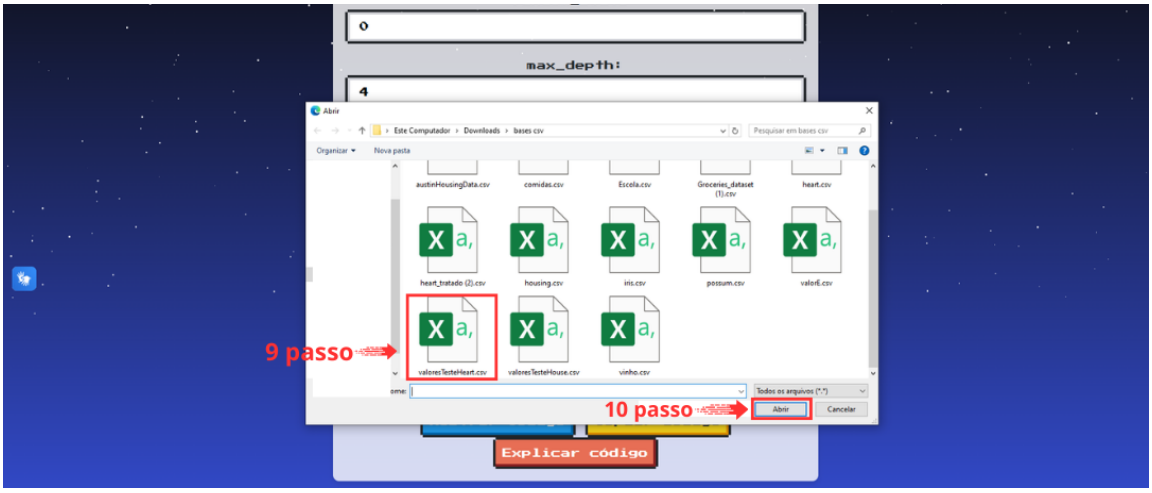


Figura 21 – Tela classificação: Etapa de seleção do arquivo CSV para teste dos dados.
Fonte: Autoria própria.

- **11 passo:** No décimo primeiro passo, o usuário deve selecionar o botão "Enviar" e observará os valores relacionados ao teste de novos dados, a acurácia do treino e do teste, a acurácia da validação cruzada e as matrizes de confusão do treino e teste, conforme ilustrado na Figuras 22 e 23.

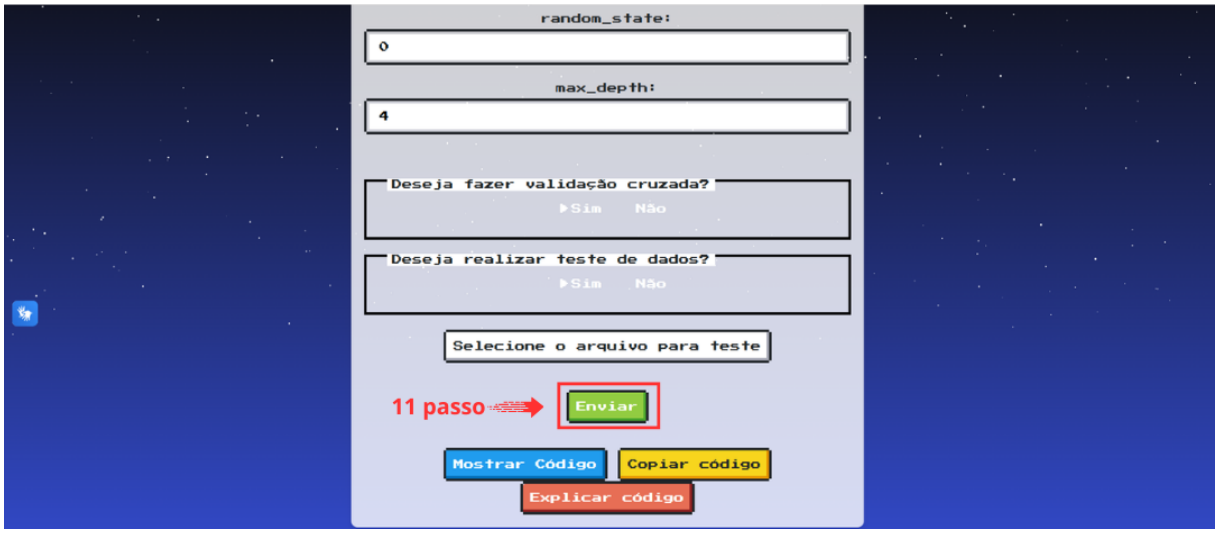


Figura 22 – Tela classificação: Etapa treinamento do modelo.
Fonte: Autoria própria.

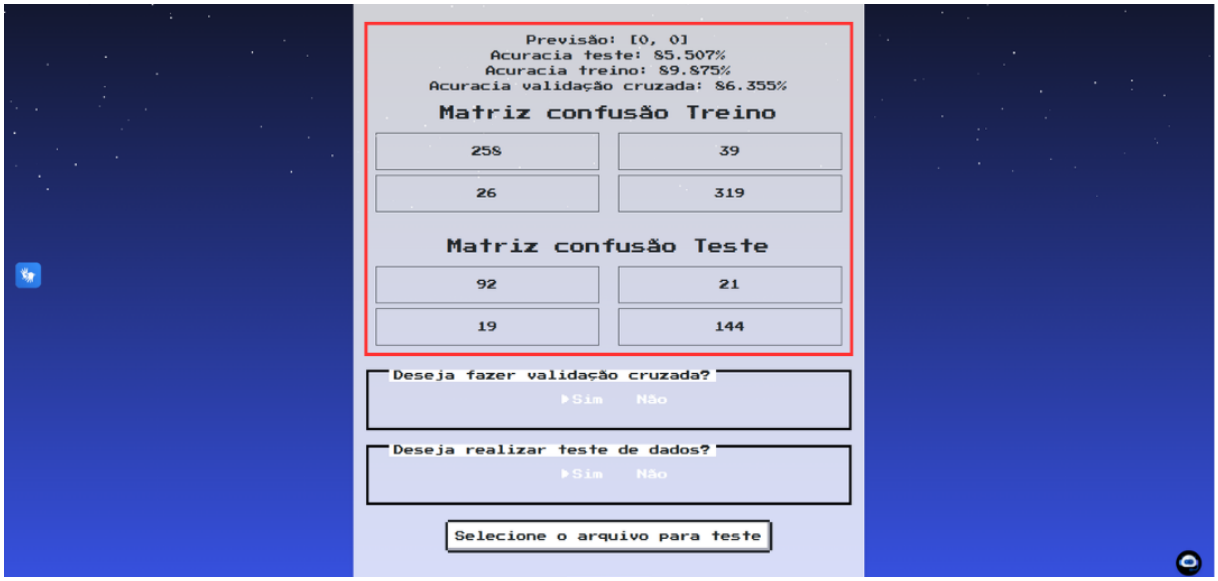


Figura 23 – Tela classificação: Métricas de desempenho do modelo.
Fonte: Autoria própria.

- **12 e 13 passo :** No décimo segundo e décimo terceiro passo, o usuário pode selecionar os botões "Mostrar código" e "Copiar Código" para visualizar e copiar o código do algoritmo de AM escolhido para a classificação, conforme ilustrado nas Figuras 24 e 25.

- **14 passo:** No décimo quarto passo, o usuário pode selecionar o botão "Explicar código" para obter uma explicação linha a linha de como funciona o código, conforme ilustrado nas Figuras 26 e 27 .

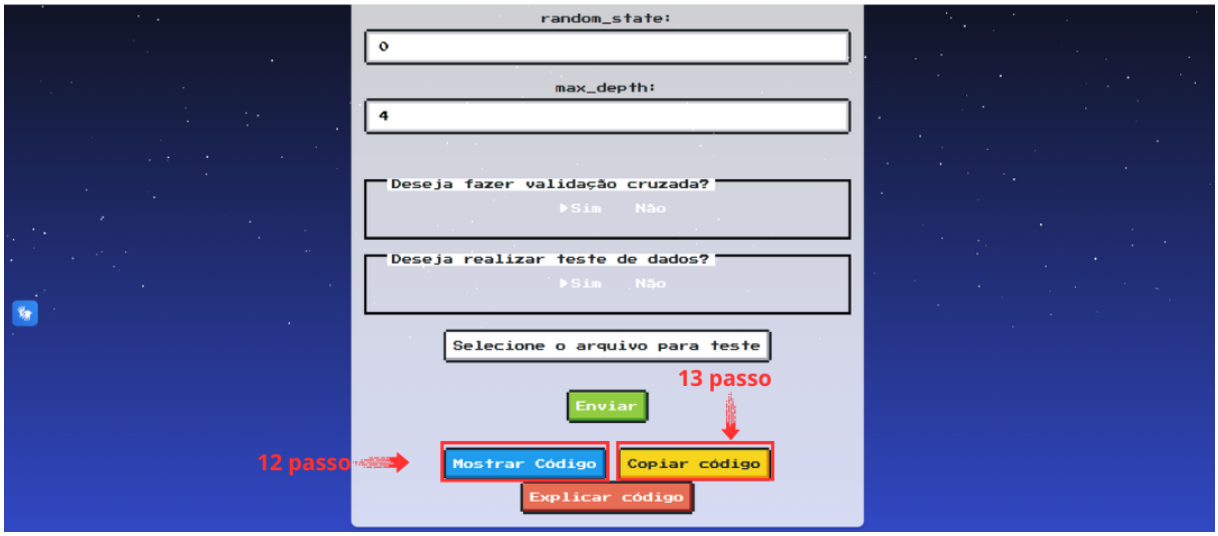


Figura 24 – Tela classificação: Etapas de exibição e cópia do código-fonte
Fonte: Autoria própria.

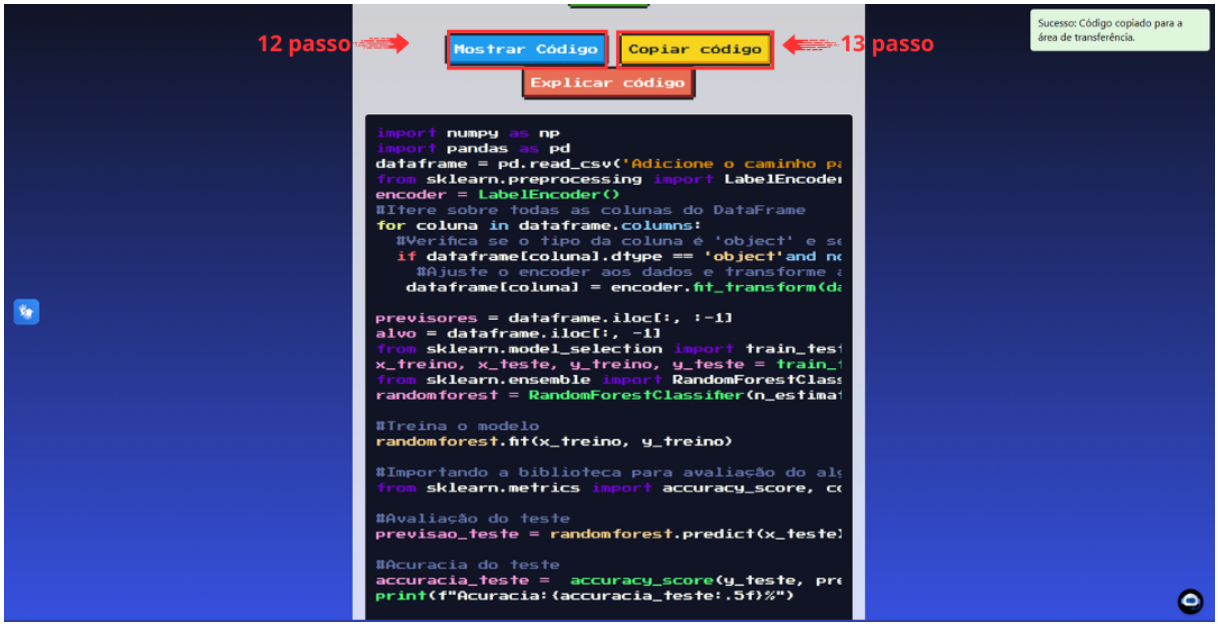


Figura 25 – Tela classificação: Código-fonte.
Fonte: Autoria própria.

- **Passo opcional:** No passo opcional, o usuário tem a possibilidade de consultar o DreamBot para obter informações detalhadas sobre o algoritmo de classificação de interesse e os hiperparâmetros associados. Ademais, o usuário possui a liberdade de realizar perguntas adicionais sobre os algoritmos, conforme ilustrado na Figura 28.

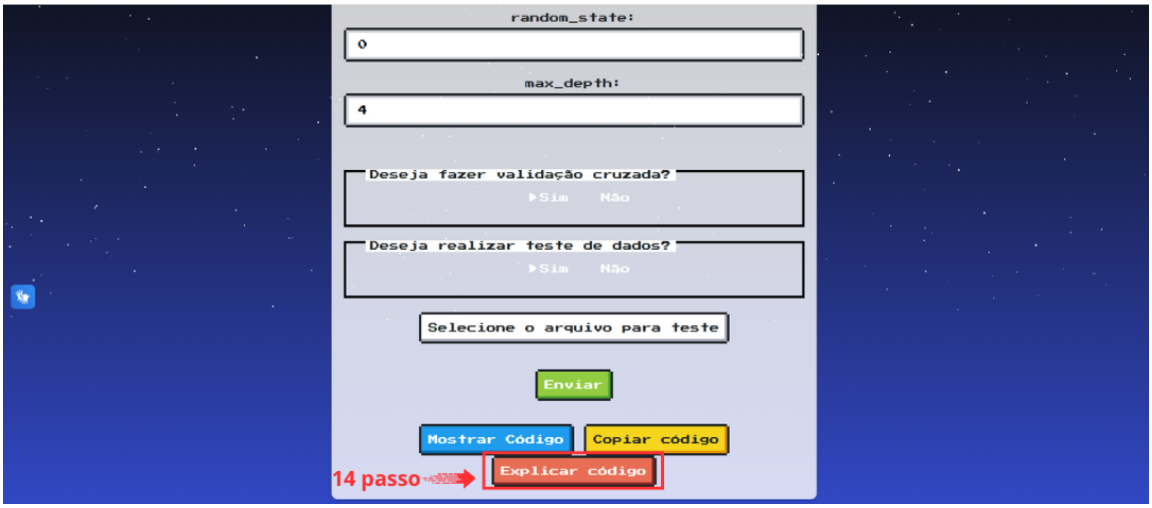


Figura 26 – Tela classificação: Etapa de explicação do código-fonte.
Fonte: Autoria própria.

Figura 27 – Tela classificação: Explicação do código-fonte.
Fonte: Autoria própria.

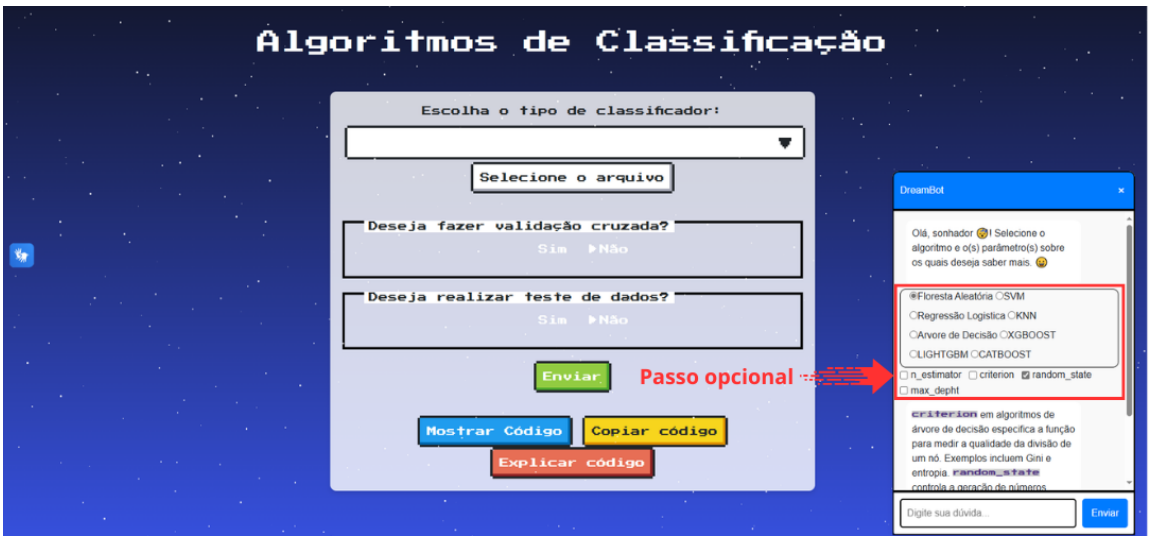


Figura 28 – Tela Classificação: Etapa Opcional para Esclarecimento de Dúvidas com o DreamBot.

3.3.8.3 Regressão

A tela de Regressão disponibiliza os seguintes algoritmos de regressão: Regressão Simples, Regressão Múltipla, Regressão Polinomial, Árvore de Decisão, Floresta Aleatória, SVM, XGBoost, LightGBM e CatBoost. A seguir, são descritos os passos que o usuário deve seguir para realizar a regressão.

- ❑ **1 passo:** No primeiro passo o usuário deve escolher o algoritmo de regressão a sua escolha, como ilustrado na Figura 29.

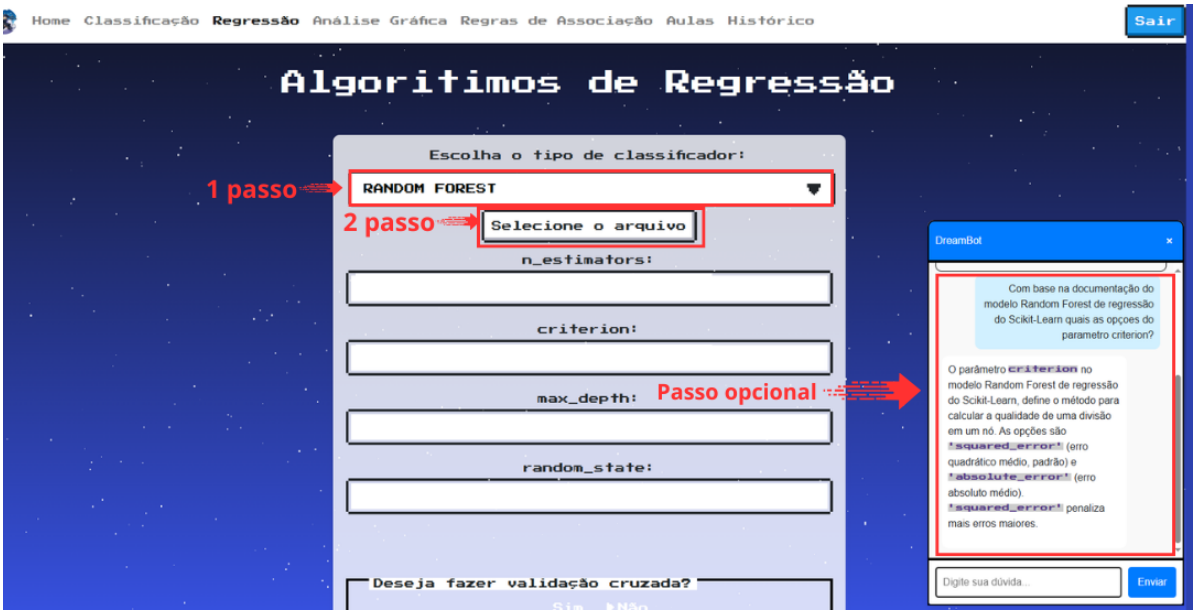


Figura 29 – Tela regressão: Etapa de seleção do algoritmo.
Fonte: Autoria própria.

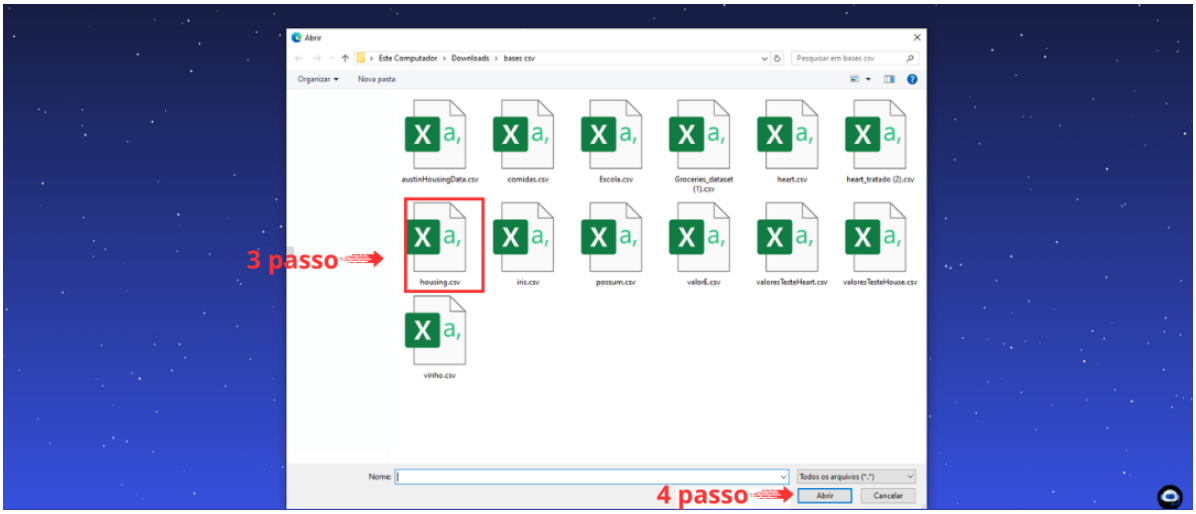


Figura 30 – Tela regressão: Etapa de seleção do arquivo CSV.
Fonte: Autoria própria.

- **2 passo:** No segundo passo, o usuário deve clicar no botão "Selecionar o arquivo" para escolher um arquivo CSV a ser utilizado na regressão, como ilustrado na Figura 29.

- **3 passo:** No terceiro passo, o usuário deve selecionar o arquivo CSV desejado em seu gerenciador de arquivos, como ilustrado na Figura 30.

- **4 passo:** No quarto passo, o usuário deve clicar no botão "Abrir" para enviar o arquivo CSV para o site, conforme ilustrado na Figura 30.

- **5 passo:** No quinto passo, o usuário deve definir os valores que serão utilizados nos hiperparâmetros, conforme ilustrado na Figura 31.

Figura 31 – Tela regressão: Etapa de seleção dos hiperparâmetros.
Fonte: Autoria própria.

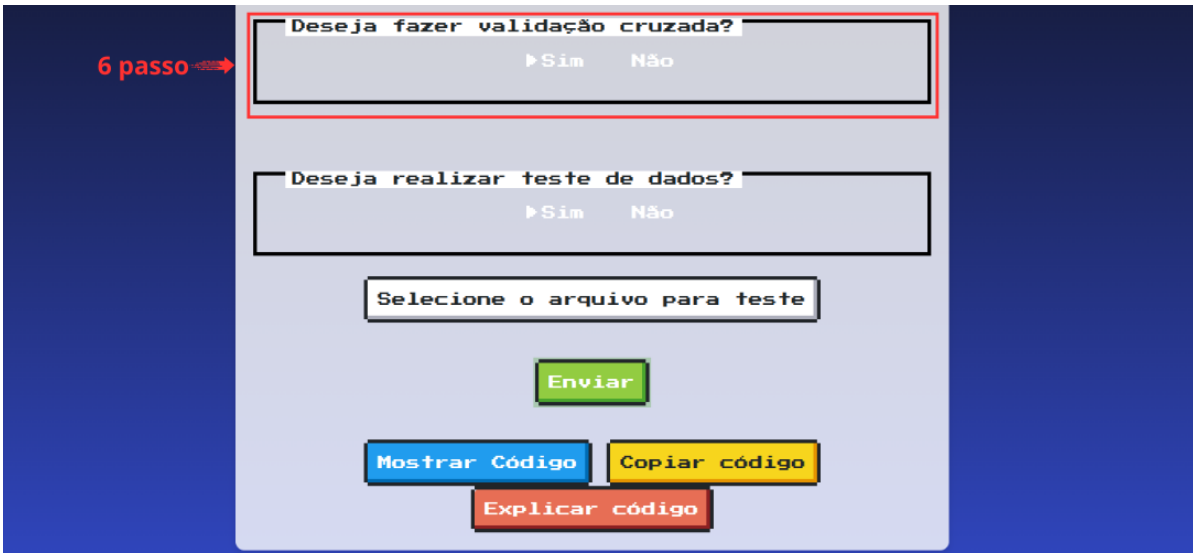


Figura 32 – Tela regressão: Etapa de seleção da validação cruzada.
Fonte: Autoria própria.

- **6 passo:** No sexto passo, o usuário pode escolher se deseja realizar a validação cruzada dos dados, selecionando a opção "Sim" ou "Não", conforme ilustrado na Figura 32.
- **7 passo:** No sétimo passo, o usuário pode escolher se deseja realizar o teste em dados diferentes dos dados de treino e teste, selecionando a opção "Sim" ou "Não", conforme ilustrado na Figura 33.
- **8 passo:** No oitavo passo, caso o usuário tenha escolhido "Realizar teste de dados" no sétimo passo, ele deve clicar no botão "Selecionar o arquivo para teste", conforme ilustrado na Figura 33.

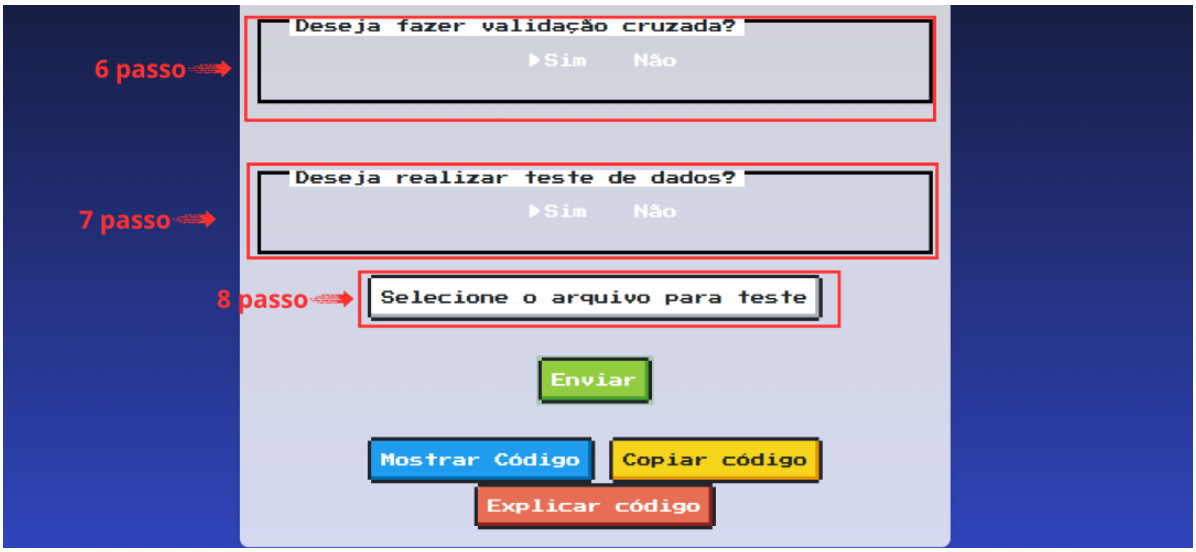


Figura 33 – Tela regressão: Etapa de teste de dados.
Fonte: Autoria própria.

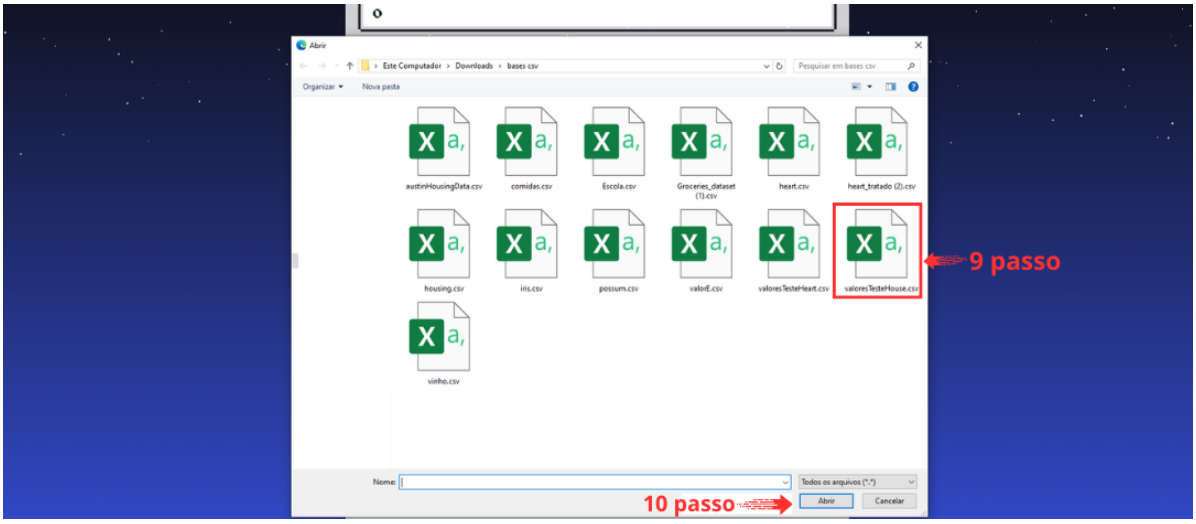


Figura 34 – Tela regressão: Etapa de seleção do arquivo CSV para teste dos dados.
Fonte: Autoria própria.

- **9 passo:** No nono passo, o usuário deve selecionar o arquivo CSV desejado para realizar teste de dados em seu gerenciador de arquivos, conforme ilustrado na Figura 34.
- **10 passo:** No décimo passo, o usuário deve clicar no botão "Abrir" para enviar o arquivo CSV para o site, conforme ilustrado na Figura 34.
- **11 passo:** No décimo primeiro passo, o usuário deve selecionar o botão "Enviar" e observará os valores relacionados ao teste de novos dados, o coeficiente de determinação do treinamento e do teste, o erro médio absoluto, a raiz do erro quadrático médio e o resultado da validação cruzada, conforme ilustrado nas Figuras 35 e 36.

Figura 35 – Tela regressão: Etapa treinamento do modelo.
Fonte: Autoria própria.

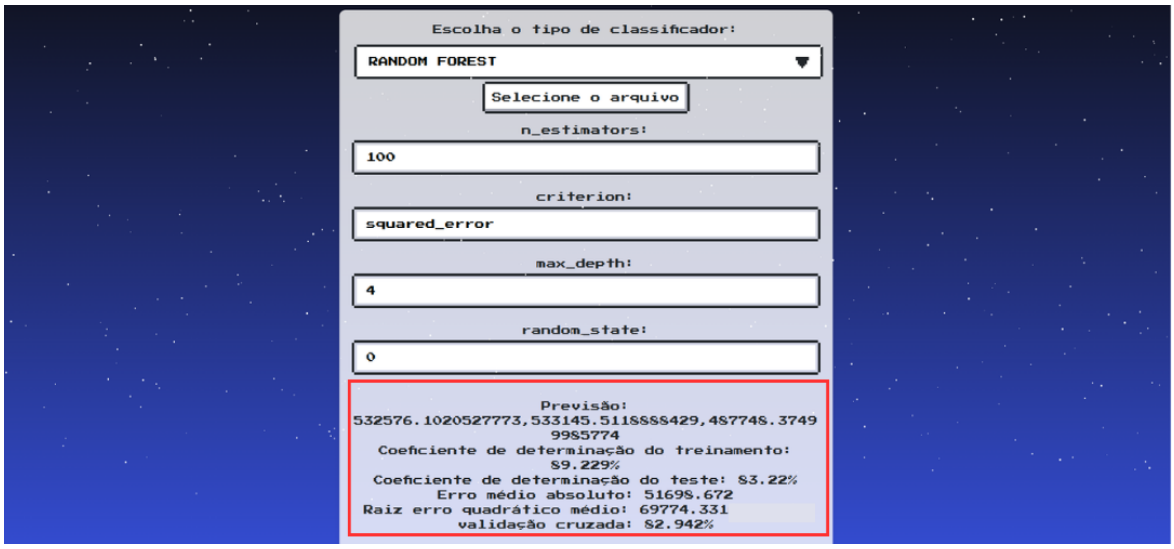


Figura 36 – Tela regressão: Métricas de desempenho do modelo.
Fonte: Autoria própria.

- **12 e 13 passo :** No décimo segundo e décimo terceiro passo, o usuário pode selecionar os botões "Mostrar código" e "Copiar Código" para visualizar e copiar o código do algoritmo de AM escolhido para a regressão, conforme ilustrado na Figuras 37 e 38.

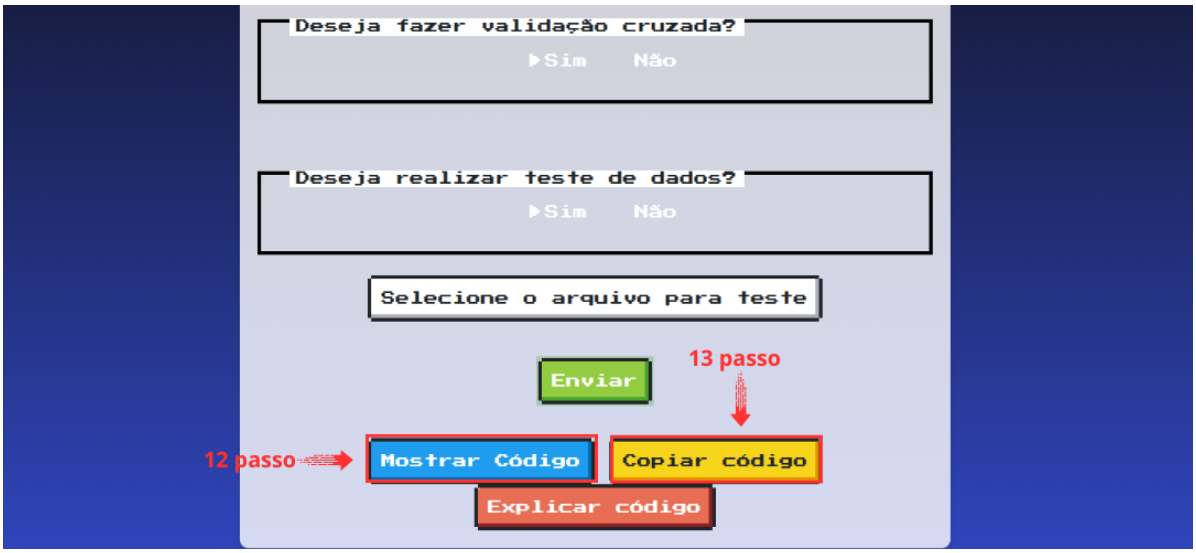


Figura 37 – Tela regressão: Etapas de exibição e cópia do código-fonte.
Fonte: Autoria própria.

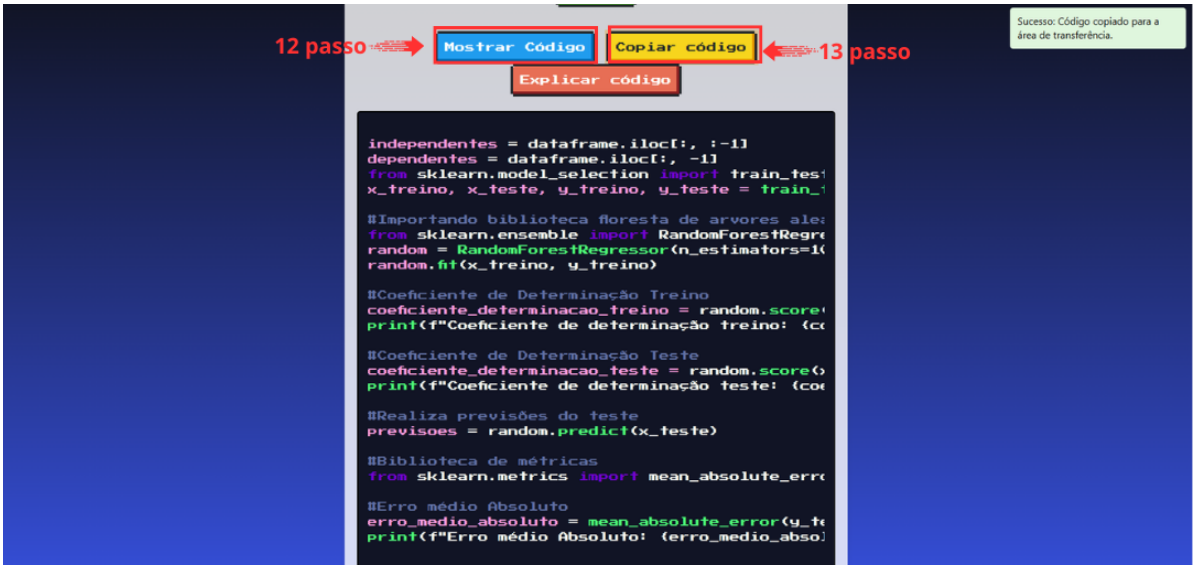


Figura 38 – Tela regressão: código-fonte.
Fonte: Autoria própria.

- **14 passo:** No décimo quarto passo, o usuário pode selecionar o botão "Explicar código" para obter uma explicação linha a linha de como funciona o código, conforme ilustrado nas Imagens 39 e 40.

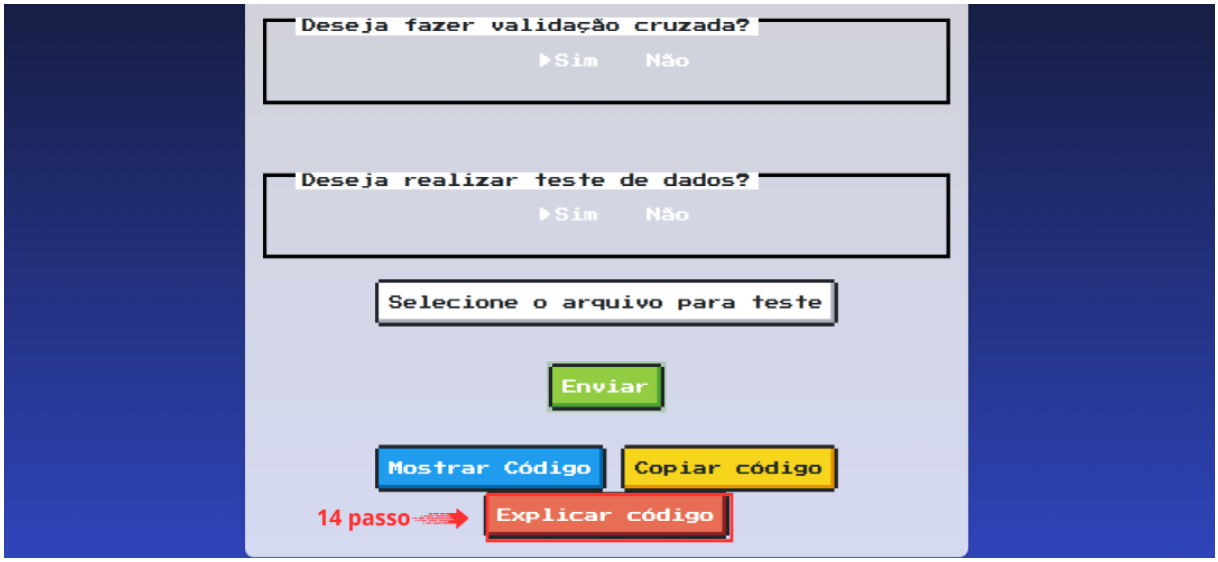


Figura 39 – Tela regressão: Etapa de explicação do código-fonte.
Fonte: Autoria própria.

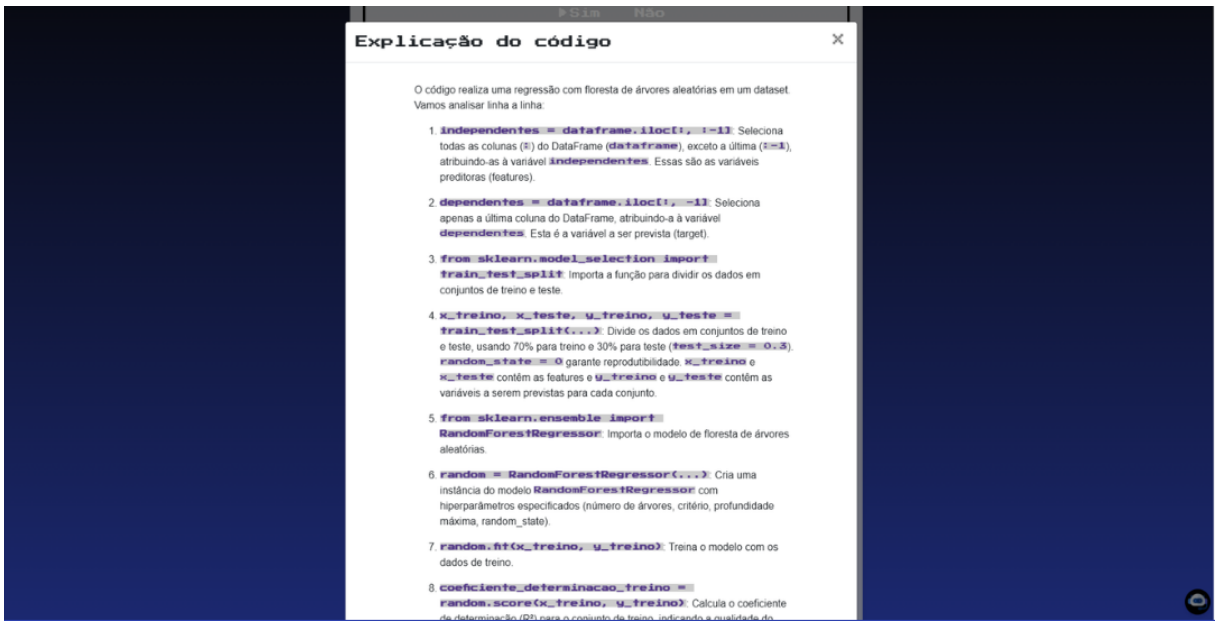


Figura 40 – Tela regressão: Explicação do código-fonte.
Fonte: Autoria própria.

- **Passo opcional:** No passo opcional, o usuário tem a possibilidade de consultar o DreamBot para obter informações detalhadas sobre o algoritmo de regressão de interesse e os hiperparâmetros associados. Ademais, o usuário possui a liberdade de realizar perguntas adicionais sobre os algoritmos, conforme ilustrado na Figura 29.

3.3.8.4 Análise Associativa

A tela de Análise Associativa disponibiliza o algoritmo Apriori para identificar regras de associação, permitindo a descoberta de padrões e relações entre diferentes itens nos dados. A seguir, são descritos os passos que o usuário deve seguir para realizar a identificação de regras de associação.

- ❑ **1 passo:** No primeiro passo, o usuário deve clicar no botão "Selecionar o arquivo" para escolher um arquivo CSV a ser utilizado na análise associativa, como ilustrado na Figura 41.

Figura 41 – Análise associativa: Etapa de seleção do arquivo CSV.
Fonte: Autoria própria.

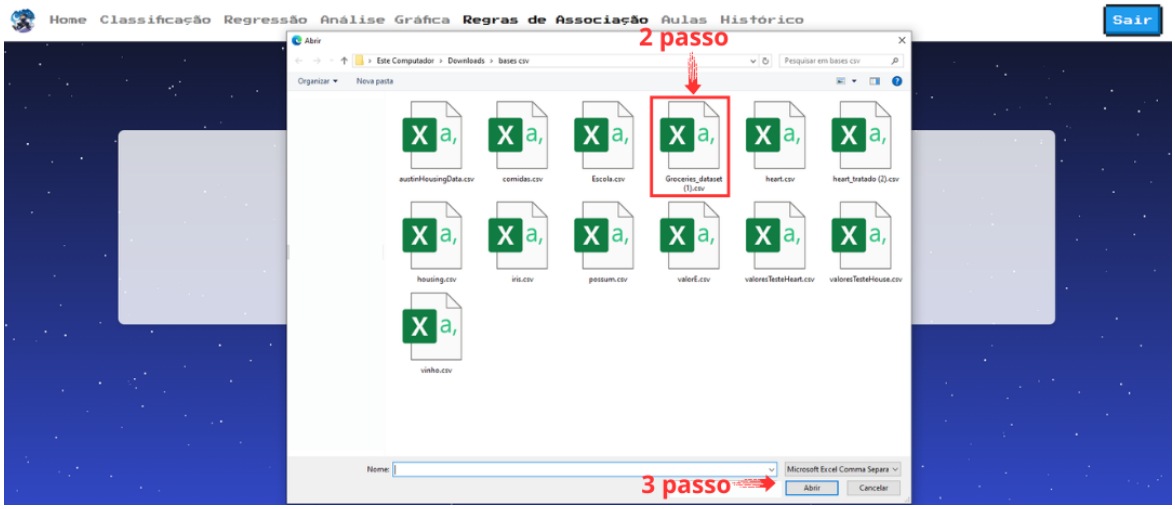


Figura 42 – Análise associativa: Etapa de seleção do arquivo CSV.
Fonte: Autoria própria.

- ❑ **2 passo:** No segundo passo, o usuário deve selecionar o arquivo CSV desejado em seu gerenciador de arquivos, como ilustrado na Figura 42.

- ❑ **3 passo:** No terceiro passo, o usuário deve clicar no botão "Abrir" para enviar o arquivo CSV para o site, conforme ilustrado na Figura 42.

❑ **4, 5 e 6 passo:** No quarto, quinto e sexto passo, o usuário deve selecionar a métrica de associação, o identificador e a coluna que agrupará os itens com base no identificador, conforme ilustrado na Figura 43.

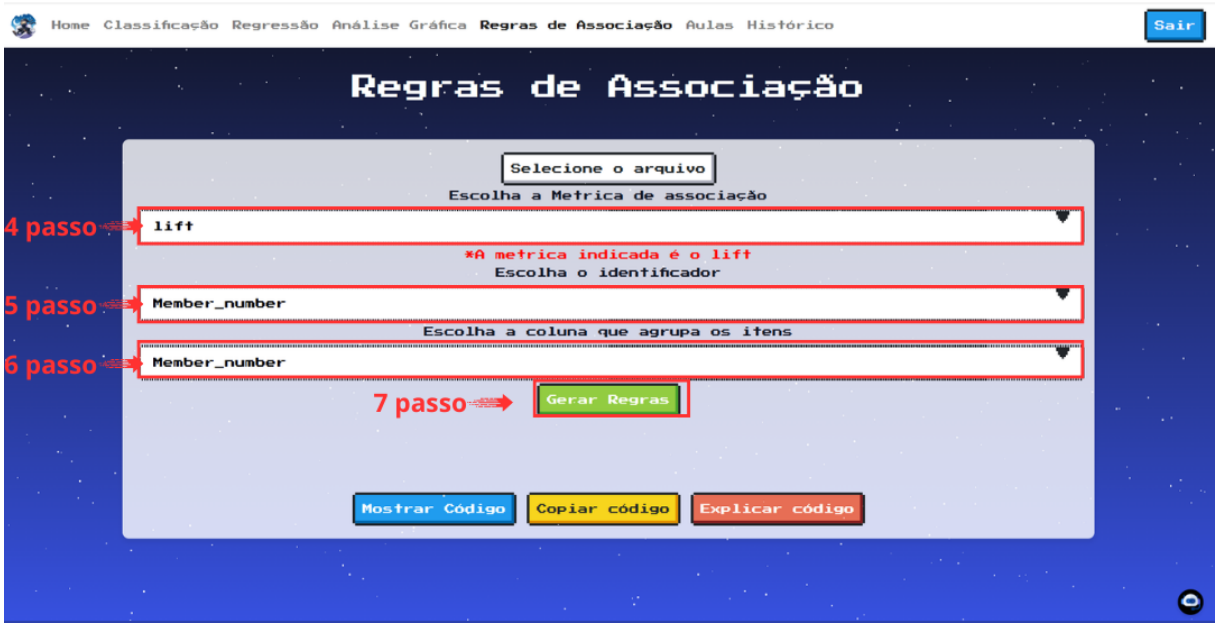
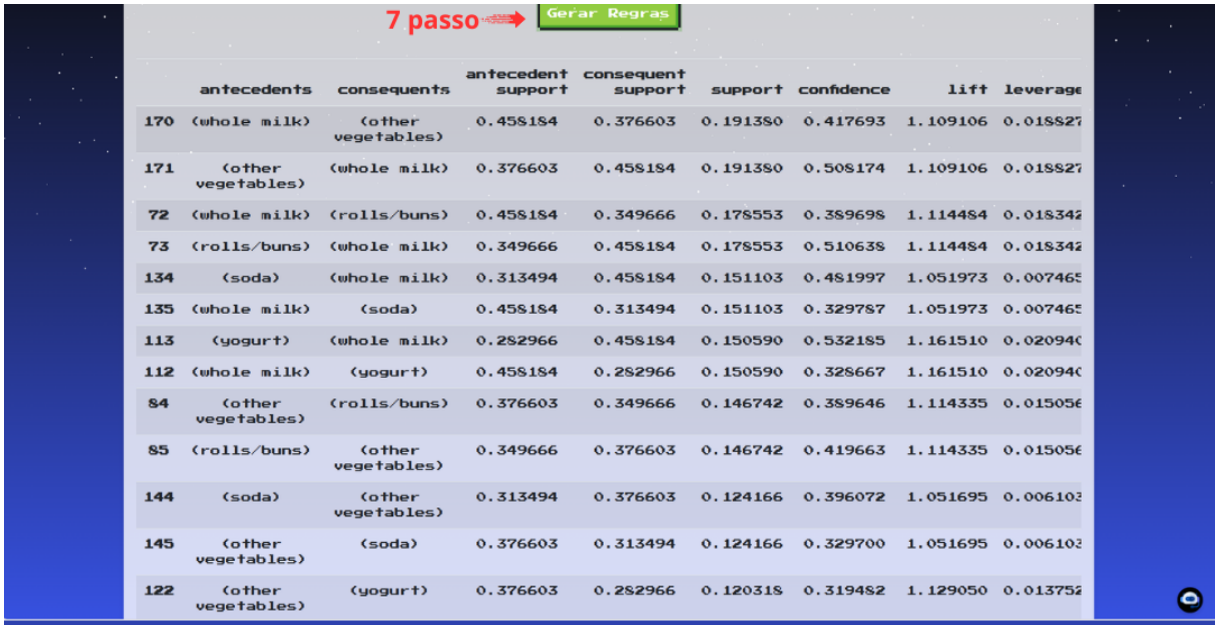


Figura 43 – Análise associativa: Etapa de seleção da métrica de associação, identificador e coluna agrupadora
Fonte: Autoria própria.



	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage
170	(whole milk)	(other vegetables)	0.458184	0.376603	0.191380	0.417693	1.109106	0.018827
171	(other vegetables)	(whole milk)	0.376603	0.458184	0.191380	0.508174	1.109106	0.018827
72	(whole milk)	(rolls/buns)	0.458184	0.349666	0.178553	0.389698	1.114484	0.018342
73	(rolls/buns)	(whole milk)	0.349666	0.458184	0.178553	0.510638	1.114484	0.018342
134	(soda)	(whole milk)	0.313494	0.458184	0.151103	0.481997	1.051973	0.007465
135	(whole milk)	(soda)	0.458184	0.313494	0.151103	0.329787	1.051973	0.007465
113	(yogurt)	(whole milk)	0.282966	0.458184	0.150590	0.532185	1.161510	0.020940
112	(whole milk)	(yogurt)	0.458184	0.282966	0.150590	0.328667	1.161510	0.020940
84	(other vegetables)	(rolls/buns)	0.376603	0.349666	0.146742	0.389646	1.114335	0.015056
85	(rolls/buns)	(other vegetables)	0.349666	0.376603	0.146742	0.419663	1.114335	0.015056
144	(soda)	(other vegetables)	0.313494	0.376603	0.124166	0.396072	1.051695	0.006103
145	(other vegetables)	(soda)	0.376603	0.313494	0.124166	0.329700	1.051695	0.006103
122	(other vegetables)	(yogurt)	0.376603	0.282966	0.120318	0.319482	1.129050	0.013752

Figura 44 – Análise associativa: Regras de associação.
Fonte: Autoria própria.

- **7 passo:** No sétimo passo, o usuário deve selecionar o botão "Gerar Regras" e observará os valores relacionados as regras de associação geradas, conforme ilustrado nas Figura 44.

- **8 e 9 passo :** No oitavo e nono passo, o usuário pode selecionar os botões "Mostrar código" e "Copiar Código" para visualizar e copiar o código do algoritmo Apriori, conforme ilustrado nas Figuras 45 e 46.

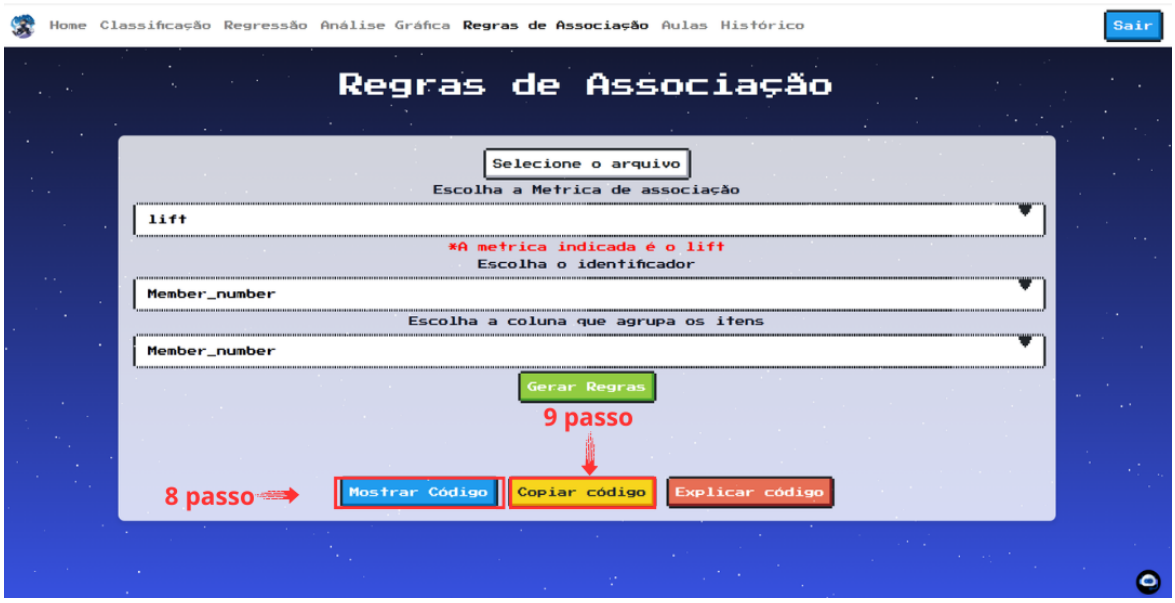


Figura 45 – Análise associativa: Etapas de exibição e cópia do código-fonte.
Fonte: Autoria própria.

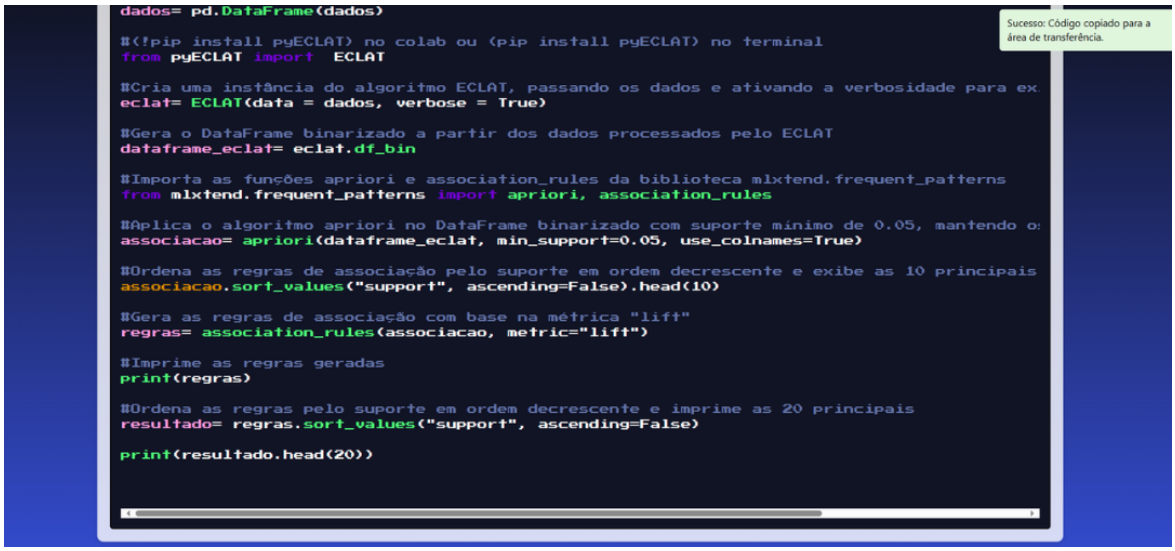


Figura 46 – Análise associativa: Código-fonte.
Fonte: Autoria própria.

❑ **10 passo:** No décimo quarto passo, o usuário pode selecionar o botão "Explicar código" para obter uma explicação linha a linha de como funciona o código, conforme ilustrado nas Figuras 47 e 48.

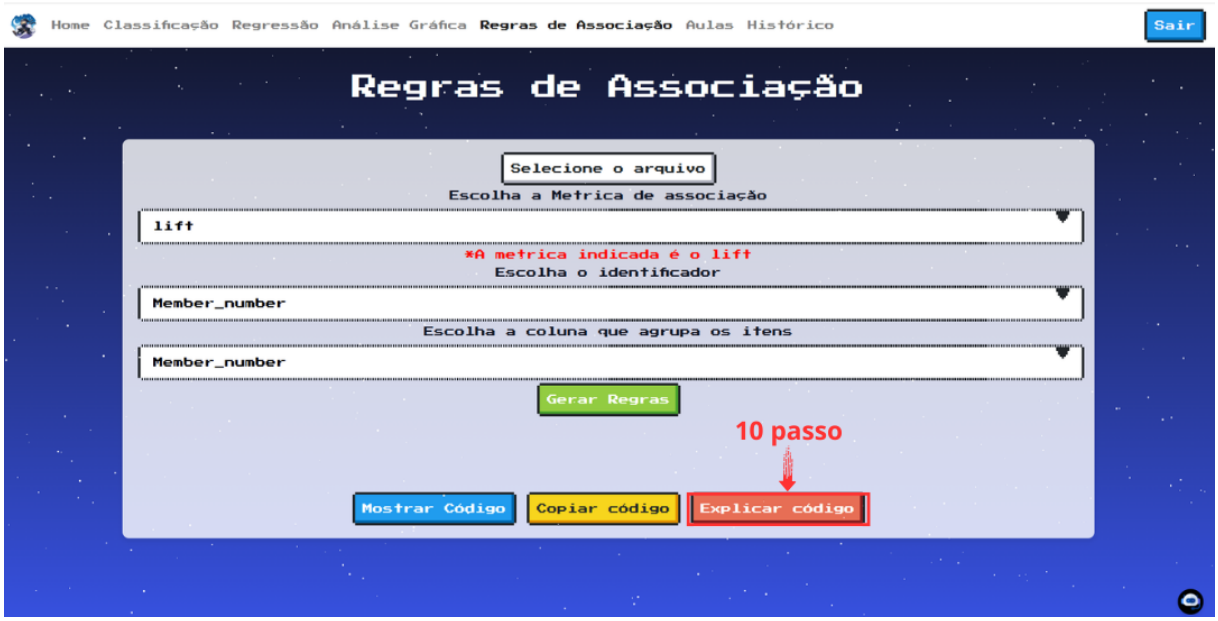


Figura 47 – Análise associativa: Etapa de explicação do código-fonte.
Fonte: Autoria própria.

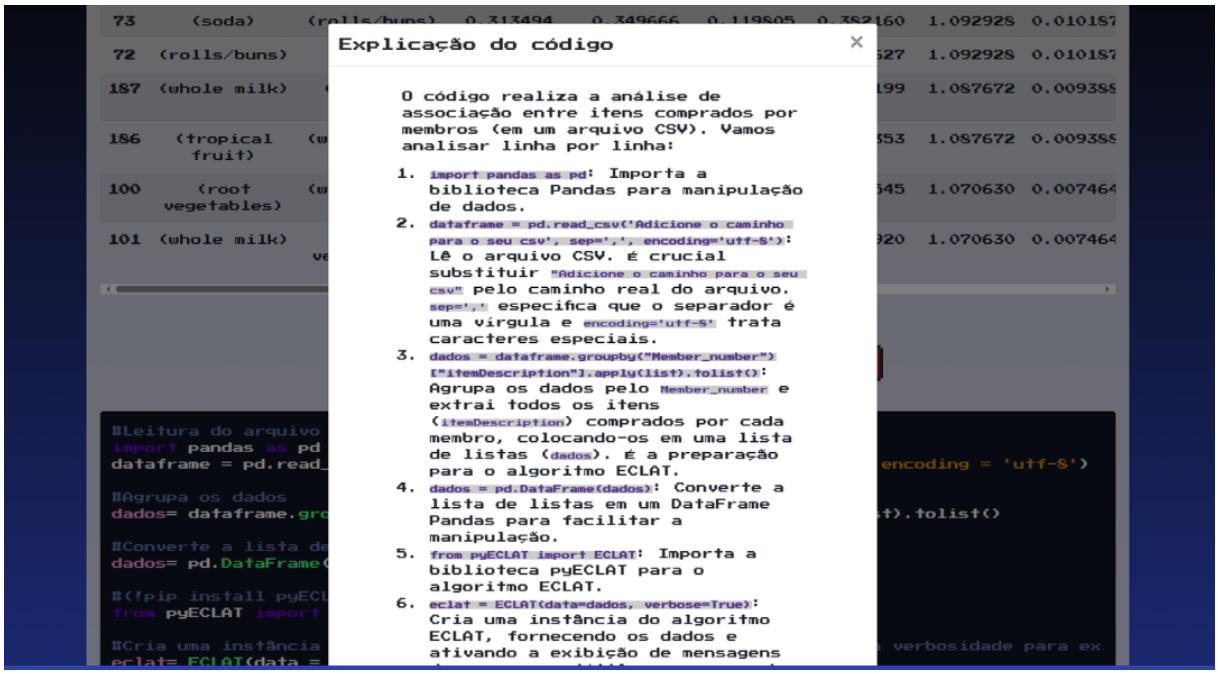


Figura 48 – Análise associativa: Explicação do código-fonte.
Fonte: Autoria própria.

❑ **Passo opcional:** No passo opcional, o usuário tem a possibilidade de consultar o DreamBot para tirar dúvidas sobre a análise associativa, conforme ilustrado na Figura 49.

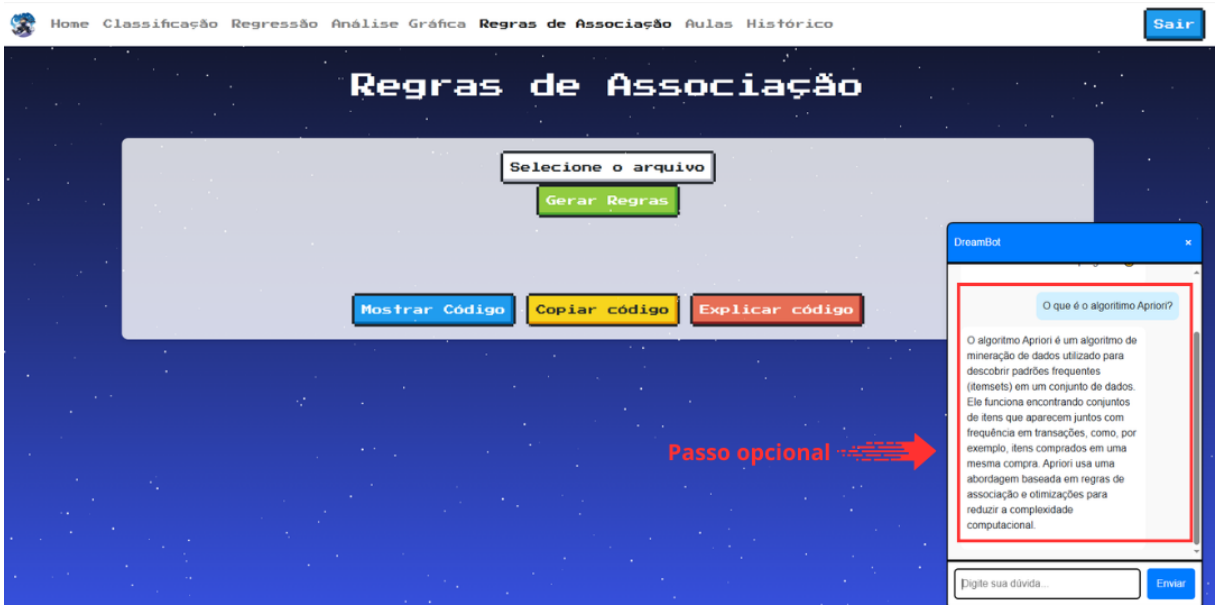


Figura 49 – Análise Associativa: Etapa Opcional para Esclarecimento de Dúvidas com o DreamBot.
Fonte: Autoria Própria.

3.3.8.5 Análise Gráfica

A tela de Análise Gráfica permite visualizar as variáveis da base CSV selecionada, possibilitando tanto a análise individual de cada variável quanto a análise associativa entre duas delas. A seguir, são apresentados os passos que o usuário deve seguir para realizar essa análise.

Figura 50 – Tela Análise Gráfica: Etapa de Seleção do Tipo de Análise Gráfica.
Fonte: Autoria Própria.

- **1 passo:** No primeiro passo, o usuário deve selecionar uma análise gráfica para todas as variáveis ou optar por realizar uma análise de correlação entre duas variáveis, conforme ilustrado na Figura 50.
- **2 passo:** No segundo passo, o usuário deve clicar no botão "Selecionar o arquivo" para escolher um arquivo CSV a ser utilizado na análise gráfica, como ilustrado na Figura 50.
- **3 passo:** No terceiro passo, o usuário deve selecionar o arquivo CSV desejado em seu gerenciador de arquivos, como ilustrado na Figura 51.

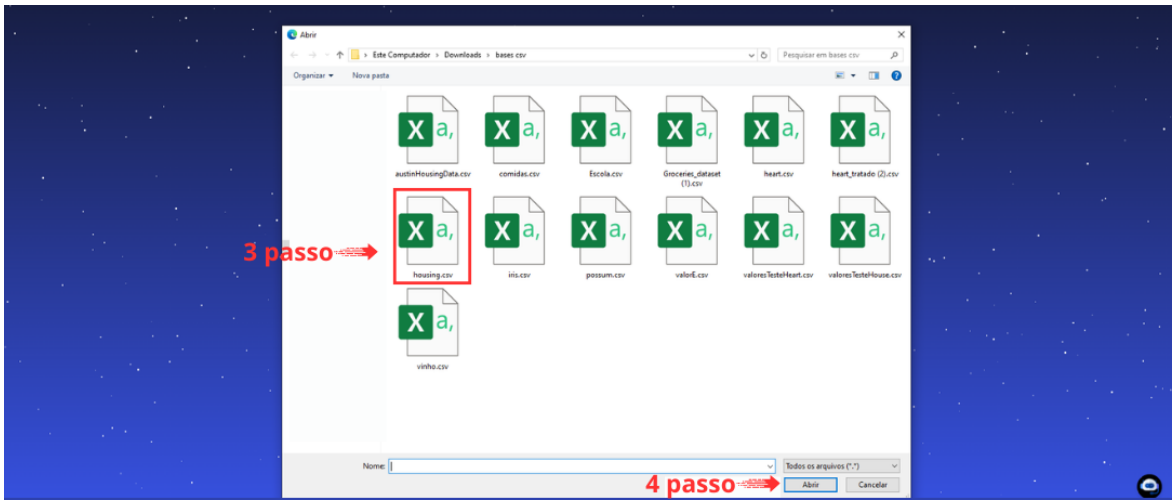


Figura 51 – Tela Análise Gráfica: Etapa de Seleção do Arquivo CSV.
Fonte: Autoria Própria.

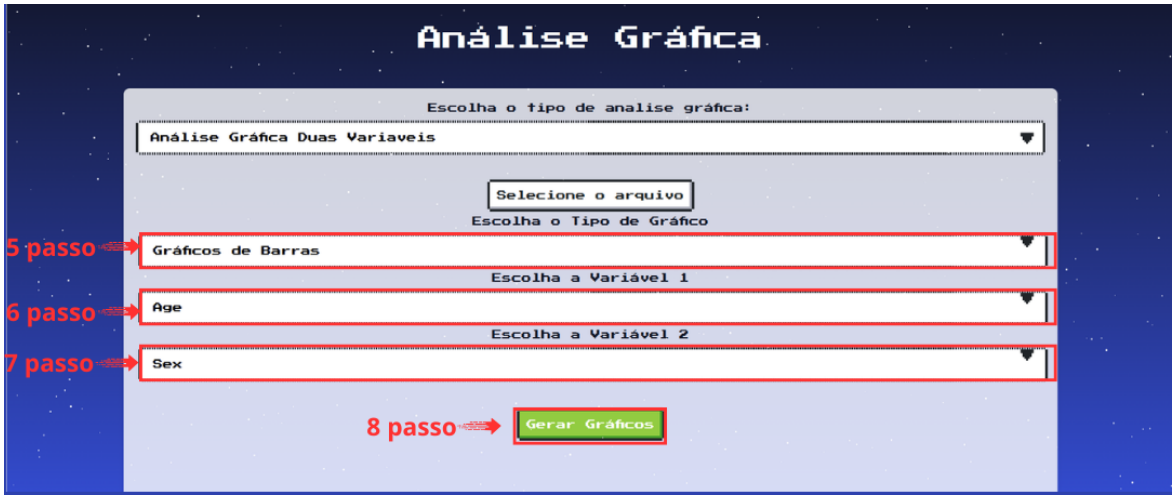


Figura 52 – Tela Análise Gráfica: Etapas de Seleção do Tipo de Gráfico, das Variáveis e Geração do Gráfico.
Fonte: Autoria Própria.

- **4 passo:** No quarto passo, o usuário deve clicar no botão "Abrir" para enviar o arquivo CSV para o site, conforme ilustrado na Figura 51.

- **5 passo:** No quinto passo, o usuário deve escolher o tipo de gráfico que deseja utilizar (barra, linha, dispersão, histograma ou boxplot), conforme ilustrado na Figura 52.

- **6 e 7 passo:** No sexto e sétimo passo, caso o usuário tenha escolhido realizar uma análise entre duas variáveis, ele deve selecionar a primeira e a segunda variável, conforme ilustrado na Figura 52.

- **8 passo:** No oitavo passo, o usuário deve selecionar o botão "Gerar Gráficos" e observará o gráfico gerado, conforme ilustrado nas Figuras 52 e 53.

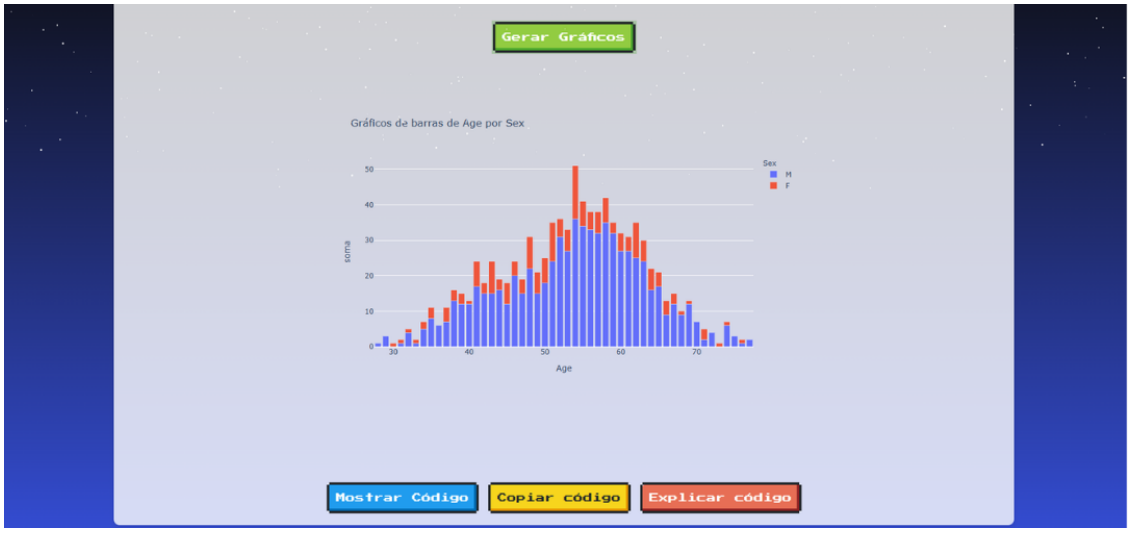


Figura 53 – Tela Análise Gráfica: Exibição do Gráfico.
Fonte: Autoria Própria.

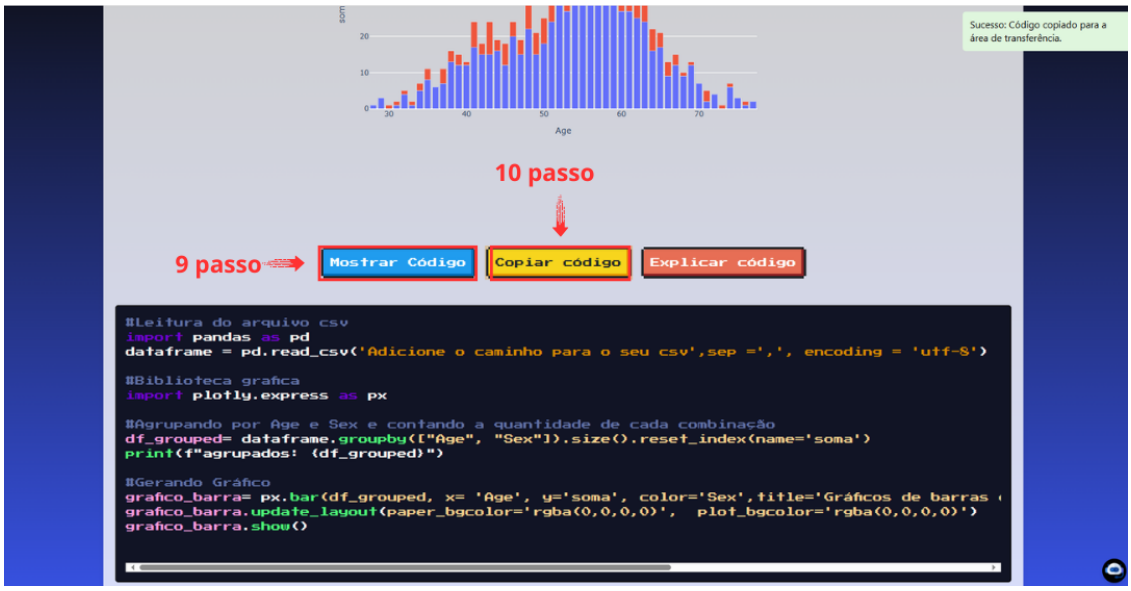


Figura 54 – Tela Análise Gráfica: Etapas de Exibição e Cópia do Código-Fonte.
Fonte: Autoria Própria.

- **9 e 10 passo :** No nono e décimo passo, o usuário pode selecionar os botões "Mostrar código" e "Copiar Código" para visualizar e copiar o código do(s) gráfico(s), conforme ilustrado na Figura 54.

- **11 passo:** No décimo primeiro passo, o usuário pode selecionar o botão "Explicar código" para obter uma explicação linha a linha de como funciona o código, conforme ilustrado nas Figuras 55 e 56

Figura 55 – Tela Análise Gráfica: Etapa de Explicação do Código-Fonte.
Fonte: Autoria Própria.

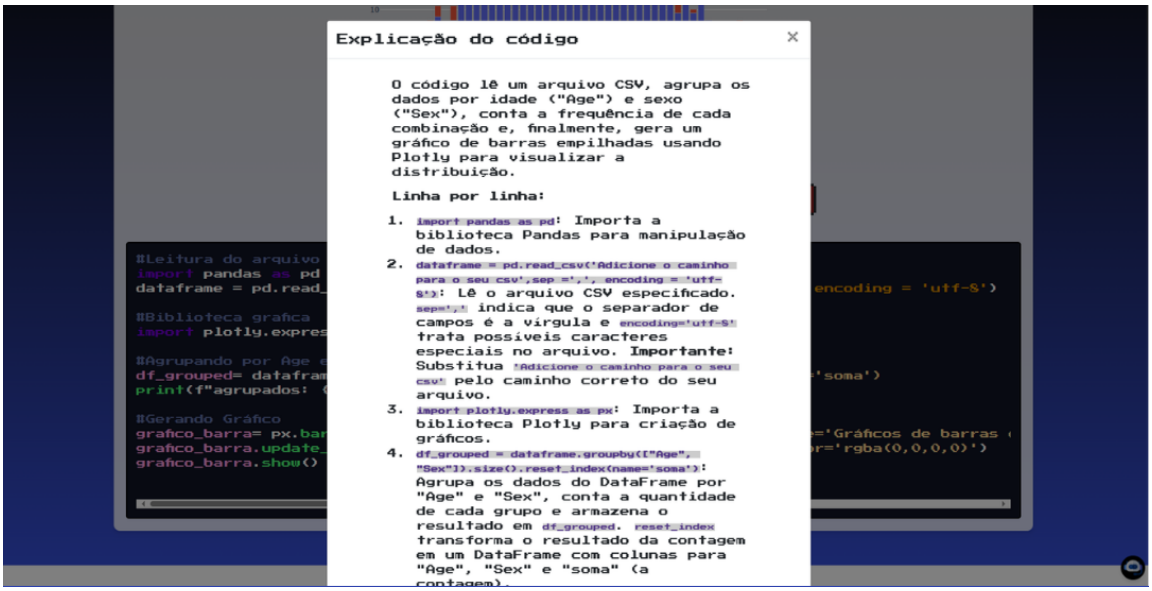


Figura 56 – Tela Análise Gráfica: Explicação do Código-Fonte.
Fonte: Autoria Própria.

- **Passo opcional:** No passo opcional, o usuário tem a possibilidade de consultar o DreamBot para tirar dúvidas sobre a análise gráfica, conforme ilustrado na Figura

57.

Figura 57 – Tela Análise Gráfica: Etapa Opcional para Esclarecimento de Dúvidas com o DreamBot.
Fonte: Autoria Própria.

3.3.8.6 Aulas

A tela de Aulas permite que o usuário estude os algoritmos de AM. A seguir, são descritos os passos que o usuário deve seguir para acessar as vídeo aulas.

- **1 passo:** No primeiro passo, o usuário pode buscar ou selecionar uma videoaula popular para estudar, conforme ilustrado na Figura 58.

Figura 58 – Tela Aulas: Etapa de Selecionar a Vídeo Aula.
Fonte: Autoria Própria.

- **2 passo:** No segundo passo, o usuário deve clicar no botão de play para iniciar a videoaula escolhida, conforme ilustrado na Figura 59.

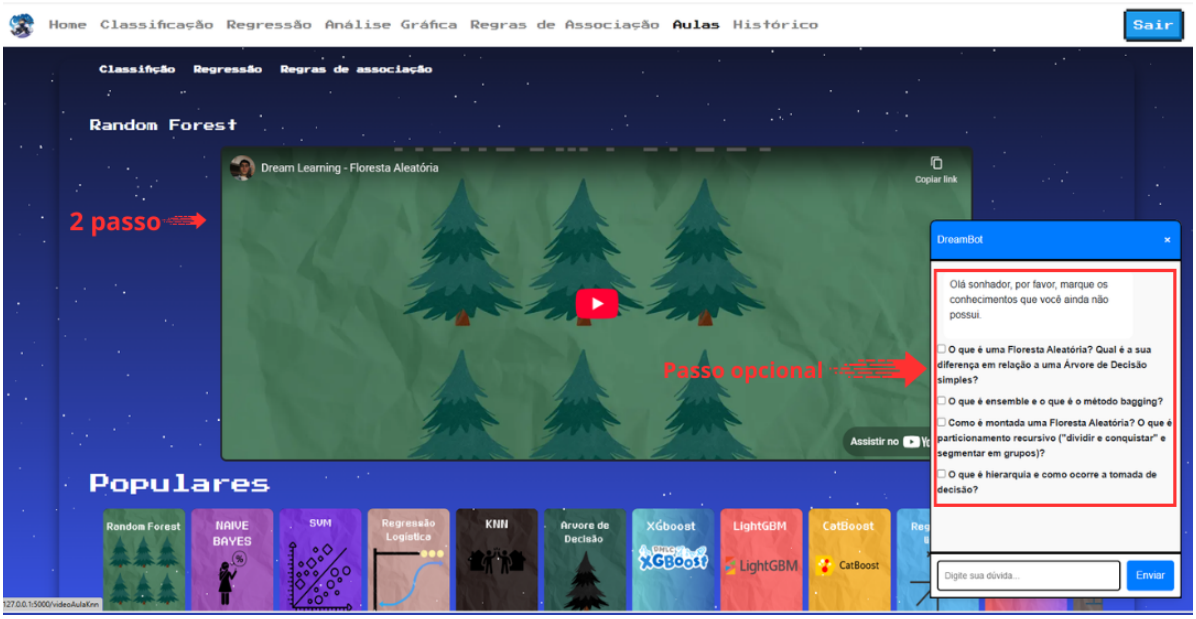


Figura 59 – Tela Aulas: Exibição da Vídeo Aula.
Fonte: Autoria Própria.

- **Passo opcional:** No passo opcional, o DreamBot solicita ao usuário que indique os conhecimentos que não possui sobre o algoritmo. Essa etapa tem como objetivo identificar os conhecimentos prévios do usuário e, se necessário, complementá-los para garantir a compreensão completa do algoritmo. Além disso, o usuário tem a liberdade de fazer perguntas adicionais sobre o algoritmo durante a vídeo aula, conforme ilustrado na Figura 59.

3.3.8.7 Histórico

A tela de Histórico permite a visualização dos algoritmos utilizados em ordem cronológica, bem como o resultado da validação cruzada e as métricas de avaliação correspondentes a cada modelo treinado. Para cada entrada registrada, o usuário tem a opção de realizar o download do arquivo CSV e de copiar o código-fonte gerado, ambos específicos para a respectiva execução. A Figura 60 ilustra as funcionalidades disponíveis na interface de histórico.

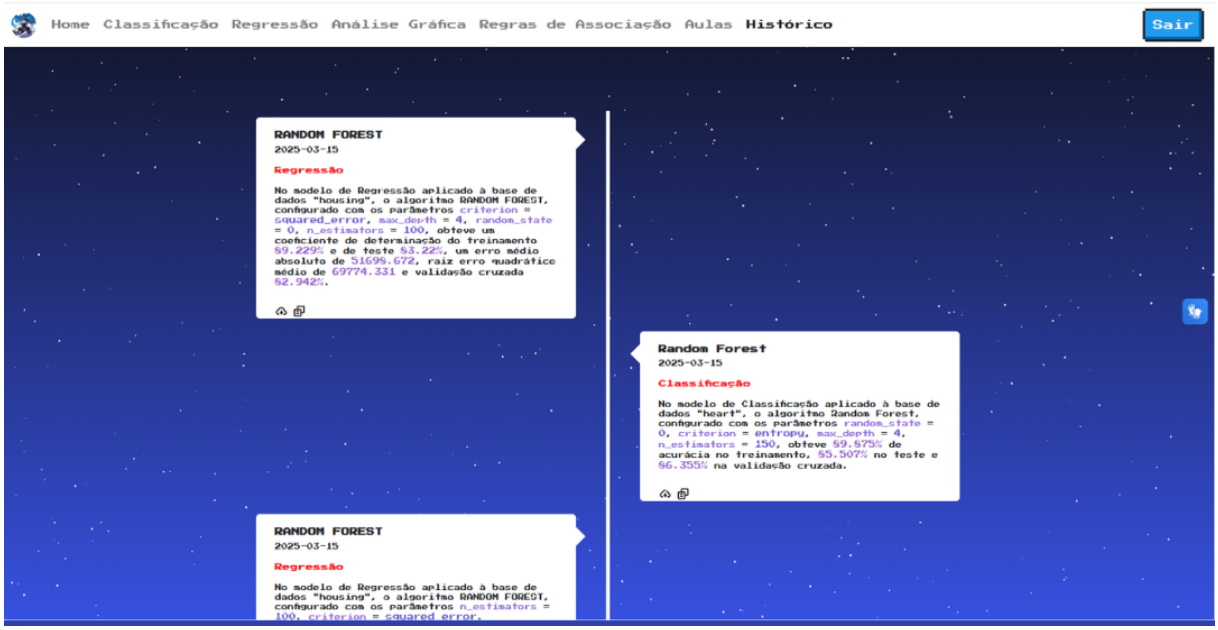


Figura 60 – Tela Histórico.
Fonte: Autoria Própria.

Capítulo 4

Considerações Finais e Trabalhos Futuros

Diante da crescente necessidade de tornar o ensino dos algoritmos de AM livre e mais acessível aos estudantes de graduação, este trabalho propôs o desenvolvimento de uma plataforma web educacional pioneira na área, denominada DreamLearning. A proposta consistiu em integrar algoritmos de AM supervisionados e não supervisionados em um ambiente web de fácil execução e compreensão, complementado por videoaulas e materiais explicativos, com conteúdo apresentado em língua portuguesa e fundamentado nos princípios dos recursos educacionais abertos. O desenvolvimento da plataforma permitiu alcançar esse objetivo, oferecendo aos usuários uma ferramenta prática e acessível para explorar os fundamentos do AM de maneira visual, intuitiva e com foco no processo de aprendizagem.

Nesse contexto, ao longo do desenvolvimento da plataforma DreamLearning, foi possível analisar diversos algoritmos de AM e selecionar aqueles que melhor se adequavam ao contexto educacional, priorizando modelos de fácil compreensão como, KNN, SVM, Floresta Aleatória, Regressão Logística, Regressão Linear e o algoritmo Apriori. Além da escolha dos modelos, após uma análise empírica também foram incorporadas funcionalidades complementares, como a visualização gráfica das variáveis e o histórico de execuções, com o intuito de enriquecer a experiência de aprendizagem do usuário e possibilitar uma melhor assimilação dos conceitos abordados.

Complementarmente, o sistema foi documentado seguindo as boas práticas de desenvolvimento de software, contemplando diagramas de caso de uso e de fluxo de dados elaborados em UML, bem como a definição clara dos requisitos funcionais e não funcionais da aplicação. Esses elementos documentais corroboram o objetivo inicial de oferecer uma solução funcional e acessível para estudantes interessados em aprender e aplicar os conceitos básicos de AM, além de indicarem oportunidades de aprimoramento para versões futuras da plataforma. Além disso, o código-fonte da plataforma DreamLearning [13]

[13] <https://github.com/EduardodosSantosRocha2/DreamLearning>

foi disponibilizado na plataforma GitHub.

Desta forma, conclui-se que o DreamLearning alcançou os objetivos propostos, servindo como um recurso educacional pioneiro em sua proposta, auxiliando a formação acadêmica de futuros profissionais da área de AM. As principais contribuições da plataforma DreamLearning concentram-se na democratização do ensino e da prática de AM, ao proporcionar uma experiência de aprendizagem simples, intuitiva e livre da necessidade de programação, promovendo, assim, a inclusão e o acesso ao conhecimento técnico por meio de uma abordagem prática, aberta e centrada no processo de aprendizagem do estudante.

Visando avaliar a usabilidade da plataforma, um experimento com 10 alunos do curso de Sistemas de Informação foi realizado utilizando a metodologia SUS, que avalia a usabilidade do site com uma média de referência de 68 pontos, conforme estabelecido por Brooke et al. (1996), em uma escala de 0 a 100. O resultado obtido foi de 66,5 pontos, valor próximo da média e que, dada a amostra reduzida, já indica uma recepção positiva da interface. Dessa forma, para trabalhos futuros, pretende-se refatorar o código do Front-End, visando implementar novas técnicas de Interface do Usuário (UI) e de Experiência do Usuário (UX), além de incorporar novas funcionalidades de AM, como a classificação de imagens por meio de Redes Neurais Convolucionais (CNN).

Nesse sentido, também está prevista a introdução de elementos de gamificação, incluindo um sistema de pontuação mensal e premiações para incentivar o aprendizado de novos usuários. Adicionalmente, busca-se aplicar a técnica de conectar "bloquinhos", conhecida como *Building Workflows*, na qual o usuário poderá selecionar diferentes nós e interconectá-los para criar um fluxo de processamento. Por exemplo, o usuário poderá escolher um nó para carregar um arquivo CSV e conectá-lo a um nó de classificação. Em seguida, poderá selecionar um nó de hiperparâmetros, definir os valores conforme o algoritmo escolhido e integrá-lo ao fluxo. O usuário também terá a opção de adicionar um nó de validação cruzada e um nó para testar novos dados.

Por fim, poderá ser incluído um nó de treinamento que, com base no fluxo criado, será responsável por gerar uma classificação, uma regressão ou regras de associação.

Referências

AGRAWAL, R.; IMIELIŃSKI, T.; SWAMI, A. Mining association rules between sets of items in large databases. In: **Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data**. Washington, D.C., USA: ACM Press, 1993. p. 207–216. Disponível em: <https://doi.org/10.1145/170035.170072>. Citado na página 30.

AKGÜN, B.; ÖĞÜDÜCÜ, Ş. G. Streaming linear regression on spark mllib and moa. In: **Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015**. New York, NY., USA: IEEE/ACM, 2015. p. 1244–1247. Disponível em: <https://doi.org/10.1145/2808797.2809374>. Citado na página 29.

ASGHARI, Z.; ARASTEH, B.; KOOCHARI, A. Effective software mutation-test using program instructions classification. **Springer Science+Business Media, LLC, part of Springer Nature**, January 2024. Published online: 9 January 2024. Disponível em: <https://doi.org/10.1007/s10836-023-06089-0>. Citado 2 vezes nas páginas 7 e 22.

Awari. **Conheça as principais linguagens de programação para Ciência de Dados**. 2022. Accessed: 2024-07-30. Disponível em: <https://awari.com.br/linguagens-de-programacao-para-ciencia-de-dados/?utm_source=blog>. Citado na página 45.

BARBOSA, L. M.; PORTES, L. A. F. A inteligência artificial. **Revista Tecnologia Educacional [on line], Rio de Janeiro**, n. 236, p. 16–27, 2023. Disponível em: <https://abt-br.org.br/wp-content/uploads/2023/03/RTE_236.pdf#page=16>. Citado na página 20.

BELSLEY, D. A.; KUH, E.; WELSCH, R. E. **Regression diagnostics: Identifying influential data and sources of collinearity**. USA: John Wiley & Sons, 2005. Disponível em: <https://doi.org/10.1002/0471725153>. Citado na página 29.

BIAMONTE, J. et al. Quantum machine learning. **Nature**, Nature Publishing Group UK London, v. 549, n. 7671, p. 195–202, 2017. Disponível em: <https://doi.org/10.1038/nature23474>. Citado na página 21.

BITTENCOURT, H. R. Regressão logística politômica: revisão teórica e aplicações. **Acta Scientiae**, v. 5, n. 1, p. 77–86, 2003. Disponível em: <http://www.periodicos.ulbra.br/index.php/acta/article/view/146>. Citado 2 vezes nas páginas 23 e 24.

BOTHMANN, L. et al. Developing open source educational resources for machine learning and data science. In: KINNAIRD, K. M.; STEINBACH, P.; GUHR, O. (Ed.). **Proceedings of the Third Teaching Machine Learning and Artificial Intelligence Workshop**. PMLR, 2023. (Proceedings of Machine Learning Research, v. 207), p. 1–6. Disponível em: <https://proceedings.mlr.press/v207/bothmann23a.html>. Citado 2 vezes nas páginas 33 e 34.

Brains.dev. **Modelos de Classificação: Regressão Logística**. 2023. Acesso em: 30 out. 2024. Disponível em: <https://brains.dev/2023/modelos-de-classificacao-regressao-logistica/>. Citado 2 vezes nas páginas 7 e 24.

BREIMAN, L. Random forests. **Machine learning**, Springer, v. 45, p. 5–32, 2001. Disponível em: <https://doi.org/10.1023/A:1010933404324>. Citado na página 26.

BROOKE, J. et al. Sus-a quick and dirty usability scale. **Usability evaluation in industry**, London, England., v. 189, n. 194, p. 4–7, 1996. Disponível em: <https://doi.org/10.1201/9781498710411-35>. Citado na página 73.

CARNEY, M. et al. Teachable machine: Approachable web-based tool for exploring machine learning classification. In: **Extended abstracts of the 2020 CHI conference on human factors in computing systems**. Honolulu, Havai, EUA: ACM, 2020. p. 1–8. Disponível em: <https://doi.org/10.1145/3334480.3382839>. Citado 4 vezes nas páginas 31, 32, 34 e 35.

CHANDRA, M. A.; BEDI, S. Survey on svm and their application in image classification. **International Journal of Information Technology**, Springer, v. 13, n. 5, p. 1–11, 2021. Disponível em: <https://doi.org/10.1007/s41870-017-0080-1>. Citado na página 22.

CHEN, T.; GUESTRIN, C. Xgboost: A scalable tree boosting system. In: **Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining**. San Francisco, CA, USA: ACM, 2016. p. 785–794. Disponível em: <https://doi.org/10.1145/2939672.2939785>. Citado na página 27.

COLVARD, N. B.; WATSON, C. E.; PARK, H. The impact of open educational resources on various student success metrics. **International Journal of Teaching and Learning in Higher Education**, ERIC, v. 30, n. 2, p. 262–276, 2018. Disponível em: <https://eric.ed.gov/?id=EJ1184998>. Citado 2 vezes nas páginas 33 e 34.

CORTES, C.; VAPNIK, V. N. Support-vector networks. **Machine Learning**, Springer, v. 20, n. 3, p. 273–297, 1995. Disponível em: <https://doi.org/10.1007/BF00994018>. Citado na página 22.

COX, D. R. The regression analysis of binary sequences. **Journal of the Royal Statistical Society Series B: Statistical Methodology**, Oxford University Press, v. 20, n. 2, p. 215–232, 1958. Disponível em: <https://doi.org/10.1111/j.2517-6161.1958.tb00292.x>. Citado na página 23.

DEHGHAN, M. H.; HAMIDI, F.; SALAJEGHEH, M. Study of linear regression based on least squares and fuzzy least absolutes deviations and its application in geography. In: IEEE. **2015 4th Iranian Joint Congress on Fuzzy and Intelligent Systems**

(CFIS). 2015. p. 1–6. Disponível em: <https://doi.org/10.1109/CFIS.2015.7391667>. Citado na página 29.

FERNANDES, A. M. d. R. **Inteligência Artificial: Noções gerais**. Rio de Janeiro, RJ: Visual Books, 2004. 146 p. ISBN 978-8575021149. Disponível em: <https://books.google.com.br/books?hl=pt-BR&lr=&id=79q5DAAAQBAJ&oi=fnd&pg=PT3&dq=Intelig%C3%AAncia+Artificial:+No%C3%A7%C3%B5es+gerais&ots=u8cz-59gY0&sig=4jfjyliy9NIog_svRZWp5Kphylo&redir_esc=y#v=onepage&q=Intelig%C3%AAncia%20Artificial%3A%20No%C3%A7%C3%B5es%20gerais&f=false>. Citado na página 20.

FIGUEIREDO, E. Requisitos funcionais e requisitos não funcionais. **Icex, Dcc/Ufmg**, 2011. Disponível em: <https://homepages.dcc.ufmg.br/~figueiredo/disciplinas/aulas/req-funcional-rnf_v01.pdf>. Citado 2 vezes nas páginas 46 e 47.

FIX, E. **Discriminatory analysis: nonparametric discrimination, consistency properties**. USAF school of Aviation Medicine, 1985. v. 1. Disponível em: <https://doi.org/10.2307/1403797>. Citado na página 24.

GOHIYA, H.; LOHIYA, H.; PATIDAR, K. A survey of xgboost system. **Int. J. Adv. Technol. Eng. Res**, v. 8, p. 25–30, 2018. Disponível em: <http://www.ijater.com/Files/aa09b180-add4-4a6d-b234-bc122eb305d4_IJATER_39_07.pdf>. Citado na página 27.

GONZALEZ, L. d. A. Regressão logística e suas aplicações. Universidade Federal do Maranhão, 2018. Disponível em: <https://monografias.ufma.br/jspui/bitstream/123456789/3572/1/LEANDRO-GONZALEZ.pdf>. Citado na página 24.

JAN, Z. et al. Artificial intelligence for industry 4.0: Systematic review of applications, challenges, and opportunities. **Expert Systems with Applications**, v. 216, p. 119456, 2023. ISSN 0957-4174. Disponível em: <https://doi.org/10.1016/j.eswa.2022.119456>. Citado na página 21.

JAYADEVA; KHEMCHANDANI, R.; CHANDRA, S. Twin support vector machines for pattern classification. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 29, n. 5, p. 905–910, 2007. Disponível em: <https://doi.org/10.1109/TPAMI.2007.1068>. Citado na página 22.

KE, G. et al. Lightgbm: A highly efficient gradient boosting decision tree. **Advances in neural information processing systems**, v. 30, 2017. Disponível em: <https://proceedings.neurips.cc/paper/2017/hash/6449f44a102fde848669bdd9eb6b76fa-Abstract.html>. Citado na página 28.

KOHAVI, R. et al. A study of cross-validation and bootstrap for accuracy estimation and model selection. In: MONTREAL, CANADA. **Ijcai**. 1995. v. 14, n. 2, p. 1137–1145. Disponível em: <https://www.researchgate.net/profile/Ron-Kohavi/publication/2352264_A_Study_of_Cross-Validation_and_Bootstrap_for_Accuracy_Estimation_and_Model_Selection/links/02e7e51bcc14c5e91c000000/A-Study-of-Cross-Validation-and-Bootstrap-for-Accuracy-Estimation-and-Model-Selection.pdf>. Citado na página 43.

KUČAK, D.; JURIČIĆ, V.; ĐAMBIĆ, G. Machine learning in education-a survey of current research trends. **Annals of DAAAM & Proceedings**, v. 29, 2018. Disponível em: <https://doi.org/10.2507/29th.daaam.proceedings.059>. Citado na página 17.

LANTZ, B. **Machine learning with R: expert techniques for predictive modeling**. Packt publishing ltd, 2019. ISBN 1782162143. Disponível em: <https://books.google.com.br/books?hl=pt-BR&lr=&id=iNuSDwAAQBAJ&oi=fnd&pg=PP1&dq=Machine+Learning+with+R&ots=O95Wpi6GO_&sig=fCen-L9TLFgrBvCgNR7w0eqDoX0&redir_esc=y#v=onepage&q=Machine%20Learning%20with%20R&f=false>. Citado 4 vezes nas páginas 24, 25, 26 e 27.

LIZAMA, O. et al. Redes de computadores arquitectura cliente-servidor. **Universidad Tecnica Federico Santa Maria**, p. 1–8, 2016. Disponível em: <http://profesores.elo.utfsm.cl/~agv/elo322/1s16/projects/reports/Proyecto%20Cliente%20-%20Servidor.pdf>. Citado na página 44.

LUNA, S. M. M. Manual práctico para el diseño de la escala likert. **Xihmai**, Universidad La Salle Pachuca, v. 2, n. 4, p. 14, 2007. Disponível em: <https://doi.org/10.37646/xihmai.v2i4.101>. Citado na página 33.

MACEDO, L. do N. **Comparação de algoritmos de aprendizado de máquina para predizer futuras cepas do vírus da influenza**. Trabalho de conclusão de curso — Universidade Federal de Uberlândia, Uberlândia, Brasil, 2023. Disponível em: <https://repositorio.ufu.br/handle/123456789/38329>. Citado na página 21.

MARTINEZ-ARROYO, M.; SUCAR, L. Learning an optimal naive bayes classifier. In: **18th International Conference on Pattern Recognition (ICPR'06)**. IEEE, 2006. v. 3, p. 1236–1239. Disponível em: <https://doi.org/10.1109/ICPR.2006.748>. Citado na página 22.

MARTINS, R. M. et al. Teaching machine learning to middle and high school students from a low socio-economic status background. **Informatics in Education**, Vilnius University Institute of Data Science and Digital Technologies, 2023. Disponível em: <https://doi.org/10.15388/infedu.2024.13>. Citado 3 vezes nas páginas 32, 34 e 35.

MASOOD, A. D.; ABDULAZEEZ, A. M.; ZEEBAREE, D. Q. Machine learning supervised algorithms of gene selection: A review. **Machine Learning**, v. 62, p. 03, 2020. Disponível em: <https://www.researchgate.net/publication/341119469_Machine_Learning_Supervised_Algorithms_of_Gene_Selection_A_Review>. Citado na página 29.

MAULUD, D.; ABDULAZEEZ, A. M. A review on linear regression comprehensive in machine learning. **Journal of Applied Science and Technology Trends**, v. 1, n. 2, p. 140–147, 2020. Disponível em: <https://doi.org/10.38094/jastt1457>. Citado na página 29.

MONTGOMERY, D. C.; PECK, E. A.; VINING, G. G. **Introduction to linear regression analysis**. New York, NY, USA: John Wiley & Sons, 2021. ISBN 978-1-119-57875-8. Disponível em: <https://books.google.com.br/books?hl=pt-BR&lr=&id=tCIgEAAAQBAJ&oi=fnd&pg=PR13&dq=introduction+to+linear+regression+analysis&ots=lgveRxf1Mn&sig=jynAzT0oFABQv4IVUFXOwo1C64M&redir_esc=y#v=onepage&q=introduction%20to%20linear%20regression%20analysis&f=false>. Citado na página 28.

MUELLER, J. P.; MASSARON, L. **Artificial intelligence for dummies**. Hoboken, NJ, USA: John Wiley & Sons, 2021. ISBN 9781119796763. Disponível em: <https://books.google.com.br/books?hl=pt-BR&lr=&id=s9RKEAAAQBAJ&oi=fnd&pg=PA1&dq=Artificial+intelligence+for+dummies&ots=c_f3ljXkJt&sig=aVpWOUbZx9vykPHLrF5smy0VAZI&redir_esc=y#v=onepage&q=Artificial%20intelligence%20for%20dummies&f=false>. Citado na página 20.

NAIK, A.; SAMANT, L. Correlation review of classification algorithm using data mining tool: Weka, rapidminer, tanagra, orange and knime. **Procedia Computer Science**, v. 85, p. 662–668, 2016. ISSN 1877-0509. International Conference on Computational Modelling and Security (CMS 2016). Disponível em: <https://doi.org/10.1016/j.procs.2016.05.251>. Citado na página 17.

NEPAL, R. et al. **Solar Power Prediction Using Satellite Data in Different Parts of Nepal**. 2024. Disponível em: <https://doi.org/10.48550/arXiv.2406.11877>. Citado 2 vezes nas páginas 7 e 27.

NORONHA, D. H.; FERNANDES, M. A. **Implementaçao em fpga de máquina de vetores de suporte (svm) para classificaçao e regressao**. 2016. Documento PDF. Disponível em: <https://www.researchgate.net/publication/309477255_Implementacao_em_FPGA_de_Maquina_de_Vetores_de_Suporte_SVM_para_Classificacao_e_Regressao>. Citado 2 vezes nas páginas 7 e 23.

OLIVEIRA, T. A. S. D. **Comparação de Algoritmos de Aprendizado de Máquina na Classificação de Neoplasias Mamárias**. 2021. 33 p. Documento PDF. Disponível em: <https://repositorio.ufu.br/bitstream/123456789/32251/1/Compara%c3%a7%c3%a3oAlgoritmosAprendizado.pdf>. Citado na página 22.

PANDA, M.; PATRA, M. R. Network intrusion detection using naive bayes. **International journal of computer science and network security**, v. 7, n. 12, p. 258–263, 2007. Documento PDF. Disponível em: <https://www.researchgate.net/publication/241397131_Network_intrusion_detection_using_naive_bayes>. Citado na página 22.

PRASAD, A. K. et al. Accurate polynomial chaos expansion for variability analysis using optimal design of experiments. In: IEEE. **2015 IEEE MTT-S International Conference on Numerical Electromagnetic and Multiphysics Modeling and Optimization (NEMO)**. IEEE, 2015. p. 1–4. Disponível em: <https://doi.org/10.1109/NEMO.2015.7415055>. Citado na página 29.

PROKHORENKOVA, L. et al. Catboost: unbiased boosting with categorical features. **Advances in neural information processing systems**, v. 31, 2018. Disponível em: <https://doi.org/10.48550/arXiv.1706.09516>. Citado na página 28.

QUINLAN, J. R. Induction of decision trees. **Machine Learning**, v. 1, n. 1, p. 81–106, Mar 1986. ISSN 1573-0565. Disponível em: <https://doi.org/10.1007/BF00116251>. Citado na página 25.

ROZIQIN, M. C.; BASUKI, A.; HARSONO, T. A comparison of montecarlo linear and dynamic polynomial regression in predicting dengue fever case. In: **2016 International Conference on Knowledge Creation and Intelligent Computing (KCIC)**. Manado, Indonesia: IEEE, 2016. p. 213–218. Disponível em: <https://doi.org/10.1109/KCIC.2016.7883649>. Citado na página 29.

RUSSELL, S. J.; NORVIG, P. **Artificial intelligence: a modern approach**. Boston, MA, USA: Pearson, 2010. ISBN 978-0134610993. Citado na página 20.

SAMPAIO, G. Introdução à arquitetura cliente-servidor. **Medium**, 2024. Acesso em: 05 nov. 2024. Disponível em: <https://medium.com/@kaisergui258/introdu%C3%A7%C3%A3o-a-arquitetura-cliente-servidor-dfae4c0218bd>. Citado 2 vezes nas páginas 8 e 44.

STRAUSS, E.; JÚNIOR, M. V. B.; FERREIRA, W. L. L. A importância de utilizar métricas adequadas de avaliação de performance em modelos preditivos de machine learning. **Projectus**, v. 7, n. 2, p. 52–62, 2022. Disponível em: <https://doi.org/10.15202/25254146.2022v7n2p52>. Citado na página 43.

TAN, P.-N.; STEINBACH, M.; KUMAR, V. **Introdução ao Data Mining Mineração de Dados**. Rio de Janeiro, Brasil: Editora Ciência Moderna Ltda, 2009. ISBN 978-85-7393-761-9. Citado na página 30.

TECH, D. **O que é e como funciona o algoritmo KNN**. 2023. Accessed: 2024-10-30. Disponível em: <https://didatica.tech/o-que-e-e-como-funciona-o-algoritmo-knn/>. Citado 2 vezes nas páginas 7 e 25.

TEIXEIRA, J. **O que é inteligência artificial**. São Paulo, Brasil: E-galáxia, 2019. ISBN 9788584742615. Citado na página 20.

TREINAMENTOS, H. **Árvore de Decisão: Algoritmo para Classificação e Regressão**. 2023. Accessed: 2024-10-30. Disponível em: <https://www.hashtagtreinamentos.com/arvore-decisao-ciencia-dados>. Citado 2 vezes nas páginas 7 e 26.

TURING, A. On computable numbers, with an application to the entscheidungs problem. **Proceedings of the London Mathematical Society Series/2 (42)**, p. 230–42, 1936. Disponível em: <https://doi.org/10.1112/plms/s2-42.1.230>. Citado na página 16.

TURING, A. M. **Computing machinery and intelligence**. Springer, 2009. Disponível em: <https://doi.org/10.1007/978-1-4020-6710-5_3>. Citado na página 21.

ZHANG, Y. et al. Telecom customer chum prediction based on half termination dynamic label and xgboost. In: **2022 IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)**. Wuhan, China: IEEE, 2022. p. 1563–1568. ISSN 2324-9013. Disponível em: <https://doi.org/10.1109/TrustCom56396.2022.00224>. Citado na página 27.

ZHANG, Z. et al. Multiple linear regression for high efficiency video intra coding. In: **ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)**. Brighton, UK: IEEE, 2019. p. 1832–1836. Disponível em: <https://doi.org/10.1109/ICASSP.2019.8682358>. Citado na página 29.

ZHOU, Z.-H. **Machine learning**. Berlin, Germany: Springer nature, 2021. ISBN 9789811519666. Citado 2 vezes nas páginas 21 e 22.

ÇAğLAYAN, C. Comparison of the code-based or tool-based teaching of the machine learning algorithm for the first-time learners. In: **2019 1st International Informatics and Software Engineering Conference (UBMYK)**. Ankara, Turkey: IEEE, 2019. p. 1–3. Disponível em: <https://doi.org/10.1109/UBMYK48245.2019.8965519>. Citado 4 vezes nas páginas 16, 32, 34 e 35.