UNIVERSIDADE FEDERAL DE SANTA CATARINA
CAMPUS REITOR JOÃO DAVID FERREIRA LIMA
PROGRAMA DE GRADUAÇÃO EM CIÊNCIAS DA COMPUTAÇÃO

Marina Pereira das Neves Guidolin

Uso de técnicas de mineração de dados e aprendizado de máquina para identificação de casos suspeitos de transtorno do humor bipolar

Florianópolis
2023

Marina Pereira das Neves Guidolin

Uso de técnicas de mineração de dados e aprendizado de máquina para identificação de casos suspeitos de transtorno do humor bipolar

Trabalho de Conclusão de Curso do PROGRAMA DE GRADUAÇÃO EM CIÊNCIAS DA COMPUTAÇÃO do CAMPUS REITOR JOÃO DAVID FERREIRA LIMA da Universidade Federal de Santa Catarina para a obtenção do título de bacharel em Ciências da Computação.
Orientador: Prof. Dr. Mateus Grellert da Silva
Coorientador: Prof. Dr. Jônata Tyska Carvalho

Florianópolis
2023

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Guidolin, Marina Pereira das Neves
Uso de técnicas de mineração de dados e aprendizado de máquina para identificação de casos suspeitos de transtorno do humor bipolar / Marina Pereira das Neves Guidolin ; orientador, Mateus Grellert da Silva, coorientador, Jônata Tyska Carvalho, 2023.
73 p.

Trabalho de Conclusão de Curso (graduação) - Universidade Federal de Santa Catarina, Centro Tecnológico, Graduação em Ciências da Computação, Florianópolis, 2023.

Inclui referências.

1. Ciências da Computação. 2. transtorno do humor bipolar. 3. aprendizado de máquina. 5. mineração de dados. I. da Silva, Mateus Grellert. II. Carvalho, Jônata Tyska. III. Universidade Federal de Santa Catarina. Graduação em Ciências da Computação. IV. Título.

Marina Pereira das Neves Guidolin

Uso de técnicas de mineração de dados e aprendizado de máquina para identificação de casos suspeitos de transtorno do humor bipolar

Florianópolis, 14 de Junho de 2023.

Banca Examinadora:

Prof. Dr. Mateus Grellert da Silva
Orientador

Prof. Dr. Jônata Tyska Carvalho
Coorientador
Instituição UFSC

Prof. Dr. Patricia Della Méa Plentz
Avaliadora
Instituição UFSC

Prof. Dr Manuella Kaster
Avaliadora
Instituição UFSC

A todos que torcem por mim, em especial Larissa e Kiara,
que me apoiaram durante todos os dias dessa trajetória

RESUMO

O Transtorno do Humor Bipolar (THB) é uma condição na qual o paciente apresenta oscilações de humor e alterações comportamentais repentinas. O diagnóstico é de extrema importância, pois possibilita que pacientes que sofrem com este transtorno possam buscar o tratamento adequado para uma qualidade de vida maior. O diagnóstico é feito por um profissional a partir da análise do perfil do paciente. Todavia, este processo envolve analisar diversos sintomas e marcadores biológicos relacionados ao transtorno. Diagnosticar um paciente com THB depende de uma análise minuciosa de interações entre diversas características, o que torna este processo bastante complexo mesmo para especialistas. A literatura tem apontado o uso de ferramentas de inteligência artificial (IA) para auxiliar no diagnóstico de doenças e transtornos, atingindo resultados satisfatórios para inúmeras aplicações. O objetivo do presente trabalho é desenvolver uma solução para identificação de THB e seus principais determinantes utilizando técnicas de aprendizado de máquina e mineração de dados. Inicialmente, os dados foram analisados utilizando diversas técnicas de visualização e sumarização, permitindo um entendimento melhor do tipo de informação presente nestes dados e sua relação com o diagnóstico de THB. O principal resultado desta etapa aponta que dentre os dados numéricos o atributo raiva é o que tem maior correlação com o THB, já entre os categóricos, o questionário *I4E* foi o que apresentou um maior percentual de ocorrências do THB para a resposta *TOC atual*. Em seguida, diferentes modelos foram treinados a partir de um banco de dados contendo pacientes já diagnosticados com THB e pacientes controle (não diagnosticados). Técnicas de ajuste de hiperparâmetros, regularização, balanceamento de dados e engenharia de atributos são adotadas com a finalidade gerar modelos com maior eficiência de predição. Os modelos foram testados com dados distintos daqueles utilizados no treino, utilizando a métrica *F1-score* como critério de desempenho. O melhor modelo foi obtido utilizando o *GradientBoostingClassifier* com técnicas de pré processamento, engenharia de atributos e balanceamento atingindo f1-score de 47,05% e acurácia de 83,12%. Este trabalho busca contribuir com profissionais de saúde mental, auxiliando estes especialistas a gerar um diagnóstico mais preciso.

Palavras-chave: Aprendizado de máquina. Mineração de dados. Transtorno do humor bipolar.

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

A Inteligência Artificial (IA) é uma área da computação definida como o estudo de agentes que recebem percepções do ambiente e tomam decisões ou realizam ações. Para isso, cada um desses agentes implementa uma função que mapeia sequências de percepção para estas decisões (RUSSEL, 2010).

A IA tem se desenvolvido muito nos últimos anos, em parte devido ao crescimento de outra área de pesquisa conhecida como Mineração de Dados (do inglês *Data Mining* - DM). Esta área estuda técnicas e ferramentas para coleta, tratamento, análise e extração de informação presente em dados de diversos tipos (textos, valores formatados, imagens etc) (AGGARWAL *et al.*, 2015).

As técnicas de DM são particularmente eficientes quando combinadas com algoritmos de Aprendizado de Máquina (do inglês *Machine Learning* - ML). De acordo com MITCHELL (1997), soluções baseadas em ML aprendem através de experiência (ou dados) a desenvolver certas tarefas, com base em uma medida de desempenho, se esse desempenho melhora a partir da experiência. Essas duas áreas de estudo são tão próximas que por vezes são utilizadas como sinônimos.

A combinação entre DM e ML possibilita que sejam construídos diversos tipos de modelos de aprendizado. Dentre eles, destacam-se os modelos preditivos, os quais conseguem definir tendências a partir dos padrões identificados e prever o valor de uma variável alvo. Tais modelos possuem diversas aplicações, sobretudo na área da saúde, na qual auxiliam os profissionais no diagnóstico de doenças como COVID-19 (ZOABI; DERI-ROZOV; SHOMRON, 2021), assim como doenças e transtornos mentais (AZAR *et al.*, 2015).

Dentre os transtornos mentais, podemos destacar o Transtorno do Humor Bipolar (THB), uma condição que afeta de 1% a 2% da população adulta (FEARS; REUS, 2015). O diagnóstico do THB ainda é feito de forma clínica e é tópico intensa de pesquisa, pois ele envolve muitas variáveis e inúmeras interações entre elas. Com isso, alguns trabalhos publicados na literatura já propõem o uso de ML para auxiliar no diagnóstico de THB (PEREZ ARRIBAS *et al.*, 2018; JADHAV *et al.*, 2019).

O presente trabalho busca trazer contribuições nesta direção, propondo uma solução que combina mineração de dados e aprendizado de máquina para auxiliar no diagnóstico do Transtorno do Humor Bipolar.

1.1 OBJETIVOS

O objetivo geral deste trabalho é verificar, a partir de dados clínicos e de escalas de saúde mental, se é possível prever um perfil que possua tendência a desenvolver o Transtorno do Humor Bipolar a partir de técnicas de *Machine Learning*.

Como objetivos específicos, são identificados os seguintes:

a) Estudo da literatura sobre transtornos de humor, possíveis bases de dados e aprendizado de máquina aplicado à saúde.

b) Pré-processamento e visualização dos dados em busca de informações úteis para as etapas posteriores.

c) Estudo de possíveis atributos derivados para auxiliar na etapa de treinamento (engenharia de atributos).

d) Treinamento de modelos preditivos utilizando uma metodologia robusta com validação cruzada e separação treino/teste.

e) Análise e validação dos resultados.

1.2 ESTRUTURA

Este trabalho foi organizado da seguinte forma no que diz respeito à sua estrutura de capítulos:

- Capítulo 2: apresentação de conceitos que serão utilizados nesse trabalho para fundamentação teórica seguido de um estudo sobre os trabalhos relacionados ao tema com finalidade de comparação de metodologias utilizadas.
- Capítulo 3: mostra a proposta de solução preparada para este trabalho juntamente com a metodologia que será seguida e as etapas necessárias para o desenvolvimento.
- Capítulo 4: trata da etapa de mineração de dados mencionada na solução pro posta, contendo as etapas de pré processamento, analise dos dados utilizados, e engenharia de atributos para serem utilizados no modelo preditivo.
- Capítulo 5: explica como se deu o treinamento dos modelos e analisa os resultados encontrados de forma gradativa a fim de comparar o impacto de cada uma das etapas.
- Capítulo 6: conclusão do trabalho mostrando pontos positivos e pontos de melhoria, além de trabalhos futuros

2 FUNDAMENTOS E TRABALHOS RELACIONADOS

Neste capítulo, será apresentado o referencial teórico do trabalho, contendo definições fundamentais para o entendimento da solução proposta. Em seguida terá uma análise sobre os trabalhos relacionados, descrevendo processos utilizados em trabalhos similares a este. Por fim, o capítulo será fechado com a sumarização dos trabalhos relacionados em uma tabela para uma comparação mais explícita entre seus objetivos, dados, algoritmos e resultados.

2.1 CONCEITOS DE INTELIGÊNCIA ARTIFICIAL

2.1.1 Aprendizado de Máquina

Segundo (MITCHELL, 1997), um processo de aprendizado se dá a partir da experiência adquirida a respeito de uma classe de tarefas, fazendo que, com o decorrer do tempo, sua performance de execução aumente conforme mais experiências são adquiridas. Quando esse processo de aprendizado é estendido para um algoritmo, tem-se o conceito de Aprendizado de Máquina (do inglês *Machine Learning* - ML).

Com isso, pode-se levantar o questionamento sobre como devem ser construídos programas de computador que possuam essa capacidade de aprender. Essa construção deve levar em consideração que uma abordagem de aprendizado de máquina contempla decisões envolvendo padrões para modelagem do projeto, treinamento para chegar ao aprendizado esperado, definição de qual será a tarefa a ser aprendida, busca de um exemplo sobre como é essa tarefa e um algoritmo para que a partir do exemplo da tarefa seja possível realizar o processo de aprendizado juntamente aos treinamentos escolhidos. (MITCHELL, 1997)

As técnicas de aprendizado de máquina podem ser categorizadas de acordo com três tipos (STUART RUSSELL, 1995), o primeiro deles é o **aprendizado não supervisionado**, no qual ao aprender a executar o exemplo de uma tarefa não se sabe qual o resultado esperado para ela. Já o segundo chama-se **aprendizado por reforço**, quando existe a necessidade de avaliar o contexto de uma tarefa e gerar um reforço positivo ou negativo para que o agente executando tome uma decisão. Por último, a técnica de **aprendizado supervisionado** diz respeito à situações em que os exemplos de tarefas a serem realizadas também já possuem os resultados esperados.

2.1.2 Árvore de Decisão

O aprendizado de máquina a partir de árvores de decisão é um dos métodos mais utilizados e práticos para inferência indutiva (MITCHELL, 1997), no qual os dados são analisados com o objetivo de reconhecimento de padrões. Seu aprendizado se dá a partir do uso de dados de entrada, com os quais são feitos testes lógicos para verificar os padrões existentes e construir um caminho a partir dos padrões considerados como verdade até

chegar em uma tomada de decisão (STUART RUSSELL, 1995). Essa sequência de passos é construída de maneira hierárquica pela importância de cada atributo, sendo que para cada dado de entrada é feita uma validação com seus possíveis valores com o objetivo de encontrar uma comparação verdadeira. Até encontrar comparações verdadeiras a árvore de decisão vai criando sua estrutura a partir de nós internos que representam as condições que foram avaliadas. Quando uma comparação é verdadeira, é criado um 'nó folha', que representa o desfecho.

A Fig.1 representa a estrutura de decisão montada por uma árvore. Dentro de cada um dos nós é possível identificar o nome do atributo ao qual ele se refere, seguido pelo *gini*, que mede a qualidade da bifurcação para aquele nó a partir do cálculo de impureza ou incerteza dos dados. Dessa maneira, o passo a passo para uma tomada de decisão se dá da seguinte forma:

- Identifica o atributo de maior relevância observando o valor de *gini*;
- Bifurca o atributo criando dois novos nós;
- Repete o processo até encontrar um nó folha ou atingir a profundidade máxima.

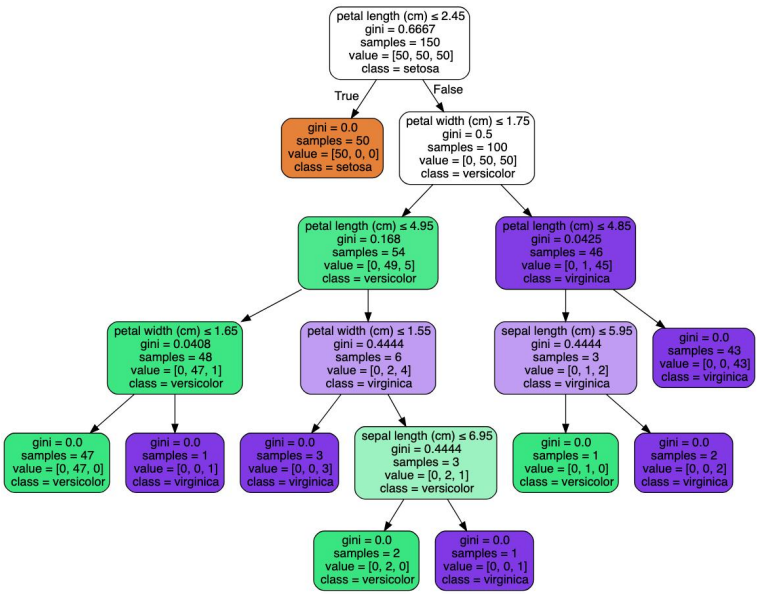

Figura 1 – Exemplo de árvore de decisão

Fonte: ScikitLearn

2.1.3 Regressão Logística

O modelo de Regressão Logística é uma ferramenta de aprendizado de máquina que é implementada a partir de um modelo linear de regressão. Ele surge a partir da necessidade de modelar a posteriori probabilidades de K classes via funções lineares em x,

enquanto ao mesmo tempo garantem que essas probabilidades somam um e permanecem no intervalo [0, 1] (HASTIE; TIBSHIRANI; FRIEDMAN, 2001). Dessa forma, ao receber os dados de entrada o algoritmo as combina de maneira linear para prever o desfecho. Ele pode ser utilizado com três propósitos:

- Prever a probabilidade de o desfecho ser igual a 1;
- Categorizar o desfecho ou as predições;
- Verificar as probabilidades ou riscos associados aos preditores do modelo

Ele é bastante utilizado para quando o desfecho tem valores binários (0 ou 1) e segue uma distribuição de probabilidade de Bernoulli. Como a distribuição de Bernoulli é um subconjunto da distribuição binomial mais geral, a regressão logística é reconhecida como um membro da família binomial de modelos de regressão (HILBE, 2015).

2.1.4 XGBoost

Outro modelo de aprendizado de máquina é o XGBoost, também conhecido como *Gradient Boosting Machine* e implementado pelo *ScikitLearn* (PEDREGOSA *et al.*, 2011). Ele é um modelo robusto que implementa técnicas que objetivam fornecer alta performance e escalabilidade, que é o principal fator responsável pelo sucesso deste modelo. Essa escalabilidade se deve aos vários sistemas e otimizações algorítmicas utilizadas por ele, que incluem otimizações ao utilizar a árvore de decisão, criando um novo algoritmo de aprendizado para lidar com dados esparsos e a inclusão de um procedimento de aprendizado ponderado, permitindo lidar com pesos ao utilizar árvores de decisão. Além disso, também implementa técnicas de regularização L1 e L2, podas de árvore, que consiste na remoção dos nós com o objetivo de diminuir a complexidade das árvores utilizadas e reduzir o *overfitting*, entre outras. (CHEN; GUESTRIN, 2016)

2.1.5 Medidores de Performance: F1 e Acurácia

Para que seja possível medir o desempenho dos modelos e quantificar a qualidade das predições é necessário que existam métricas padrão para avaliar todos de maneira igual e poder comparar resultados. Com este fim, duas métricas serão as principais avaliadas neste trabalho: a acurácia e o f1.

A acurácia é a métrica que diz respeito à performance dos modelos, avaliando o quão corretas são as predições feitas por ele e medindo a proporção de erros e acertos de acordo com o numero total de entradas.

A Equação 1 indica como é feito o cálculo da acurácia para classificação binária, sendo *tp* os valores que foram previstos como positivos e realmente eram positivos (ver-

dadeiros positivos), *tn* os valores previstos como negativos e eram negativos (verdadeiros negativos), *fp* e *fn* representando falsos positivos e falsos negativos, respectivamente.

$$acurácia = \frac{tp + tn}{tp + fp + tn + fn} \tag{1}$$

O F1 é uma métrica que depende de outras duas: precisão e *recall* (SOKOLOVA; LAPALME, 2009). A precisão define a proporção de positivos verdadeiros dentre todos os verdadeiros indicados pelo modelo, quantificando o quão preciso o modelo foi em prever corretamente o desfecho, representado pela Equação 2. Já o *recall* representa a taxa de positivos verdadeiros, medindo o quão bom o modelo é em identificar os valores positivos do desfecho, cuja fórmula está representada na Equação 3.

$$precisão = \frac{tp}{tp + fp} \tag{2}$$

$$recall = \frac{tp}{tp + fn} \tag{3}$$

A partir do cálculo de precisão e *recall*, podemos calcular a métrica F1 com base em ambas as métricas de acordo a Equação 4:

$$F1 = \frac{2 * (precisão * recall)}{precisão + recall} \tag{4}$$

2.2 CONCEITOS RELACIONADOS AO DESFECHO

2.2.1 Transtorno do Humor Bipolar

O Transtorno do Humor Bipolar - THB é um transtorno de humor que afeta de 1% a 2% da população adulta (FEARS; REUS, 2015). De acordo com (AMERICAN PSYCHIATRIC ASSOCIATION; ASSOCIATION *et al.*, 2013), existem três tipo de manifestações do THB: Transtorno Bipolar I: transtorno maníaco-depressivo que pode existir com e sem episódios psicóticos; Transtorno Bipolar II: episódios depressivos pelo menos um hipomaníacos durante a vida; Transtorno Ciclotímico: é um transtorno cíclico que causa breves episódios de hipomania e depressão.

Dos pacientes que apresentam o THB, entre 10-20% tiram a própria vida e aproximadamente um terço admite ao menos uma tentativa de suicídio (MÜLLER-OERLINGHAUSEN; BERGHÖFER; BAUER, 2002). As manifestações clínicas desse transtorno são muito diversas, e incluem muitas variações de humor, fazendo com que seu diagnóstico seja bastante complexo. Manifestações clínicas leves, por exemplo, são difíceis de distinguir de oscilações normais de humor e características da personalidade, principalmente em fases iniciais do transtorno. Porém, mesmo quando os sintomas apresentados são típicos do transtorno bipolar, ainda podem ser diagnosticados erroneamente como ansiedade comórbida, por exemplo. Tais fatores tornam o diagnóstico de transtorno

bipolar muito difícil, aumentando sua complexidade para indivíduos que o apresentam já na adolescência, quando o cérebro ainda está em fase de desenvolvimento. (GRANDE *et al.*, 2016)

2.3 TRABALHOS RELACIONADOS

A combinação de técnicas de *machine learning* e *data science* possibilita a construção de modelos preditivos, o que facilita o reconhecimento de padrões. Na área da saúde é possível analisar padrões clínicos com o objetivo de prever comportamentos e condições para que o diagnóstico de doenças mentais seja facilitado. Diversos trabalhos já utilizam essa abordagem para resolver ou mitigar problemas importantes relacionados à saúde mental. Portanto, foi realizada uma revisão exploratória da literatura, buscando trabalhos relacionados à proposta deste projeto.

Em um primeiro momento, foi feita uma busca no Google Scholar pelo termo *"machine learning AND bipolar disorder"* e alguns trabalhos similares foram encontrados, visto que além de abordarem o diagnóstico do transtorno do humor bipolar, também usam dados provenientes de questionários para a análise. Um deles foi (PEREZ ARRIBAS *et al.*, 2018), que entre pacientes saudáveis, com transtorno do humor bipolar ou transtorno de personalidade borderline visa classificá-los de acordo com seu transtorno e prever seus humores subsequentes. Para isso, foram utilizados dados vindos de um questionário no qual os participantes classificavam diariamente seu humor de acordo com seis diferentes categorias: ansiedade, euforia, tristeza, raiva, irritabilidade e energia. Tanto para prever o diagnóstico do paciente quanto seu humor, o trabalho aplicou regressão utilizando random forest e os resultados apresentados mostram que 75% dos participantes foram categorizados de acordo com seu diagnóstico original e na previsão do humor dos pacientes foram obtidos 89-98% de acurácia em pacientes saudáveis, 82-90% de acurácia para pacientes com transtorno do humor bipolar e 70-78% para pacientes com transtorno de personalidade borderline.

Outro trabalho encontrado foi (JADHAV *et al.*, 2019), que objetiva identificar pacientes que possuem o THB para auxiliar em seus diagnósticos. Nele é ressaltada a importância do diagnóstico do transtorno do humor bipolar, visto que esse transtorno persiste ao longo da vida de 4-5% da população geral. Para atingir o objetivo, o trabalho utiliza um questionário chamado *Mood Disorder Questionnaire* (MDQ), que é aplicado como uma abordagem inicial para o diagnóstico da presença de sintomas do THB e composto por perguntas abordando situações não comuns, que geralmente não ocorrem com pessoas consideradas saudáveis. Cada resposta pode ser preenchida com *"SIM"* ou *"NÃO"* e, de acordo com os resultados, outras perguntas são ou não aplicadas ao paciente. O trabalho também menciona pesquisas demonstrando uma uma taxa de 70% de acerto quando aplicado o MDQ para o diagnóstico do transtorno. Para a construção do modelo, foi implementada uma estrutura de árvore de decisão utilizando a biblioteca *SKlearn* da

linguagem Python, que internamente utiliza o algoritmo CART. Com isso, um *dataset* possuindo dados coletados de 983 testes de triagem aplicados, sendo 864 negativos para THB e 119 positivos, foi utilizado e dividido em duas partes: 60% dos dados foram usado para o treinamento do modelo e 40% para testá-lo. Os resultados apresentaram uma acurácia de 88.07% e identificou perda de acurácia nas situações em que menos de sete perguntas do MDQ eram respondidas.

Artigos mais recentes também foram encontrados, dentre eles, (R.SARANYA, 2022). No trabalho são mencionadas as fases do transtorno do humor bipolar ressaltando manifestações de episódios de mania e depressão. O conjunto de dados utilizado contém dados relativos a idade, gênero e sentimentos que os pacientes possuem, como impaciência e melancolia, por exemplo. Após isso, o autor discorre sobre a importância do pré processamento dos dados visto que os valores faltantes ou outros fatores podem inutilizar o *dataset*. Além disso, também é mencionado o aumento na qualidade e eficácia das descobertas por conta do pré processamento. Para realizá-lo foram necessárias duas etapas, sendo a primeira a normalização dos dados para que informações discrepantes fossem removidas e a segunda a utilização de *DecisionTreeRegressor* com *Standard scalar*. Após o pré processamento foi feita a seleção de features utilizando algoritmo de floresta aleatória e regressão logística. O modelo utilizado foi o de redes neurais convolucionais com memória de curto e longo prazo, que foi aprimorado com a otimização de Adam (Estimativa de Momento Adaptativo). Por fim, o modelo utilizado no artigo atingiu uma acurácia de 97.2%.

Também foram encontrados trabalhos relevantes que utilizam diferentes de tipos de dados para identificar o transtorno do humor bipolar, como (SCHNACK *et al.*, 2014). Nesse trabalho, por ser incerto o discernimento entre pacientes com transtorno do humor bipolar e esquizofrenia apenas pela análise de resultados de ressonância magnética, seu objetivo é auxiliar o diagnóstico e diferenciar esses dois transtornos a partir da análise do exame. Para isso, foram coletados dados de ressonância magnética de 66 pacientes diagnosticados com esquizofrenia, 66 pacientes diagnosticados com transtorno do humor bipolar e 66 pacientes saudáveis. Utilizando esses dados, foi realizado o treinamento de três modelos de máquinas de vetores de suporte (*Support Vector Machine - SVM*) para que os pacientes fossem separados de acordo com a densidade de massa cinzenta apresentada nas ressonâncias e, com o uso de validação cruzada, foi testado o poder preditivo dos modelos. Com isso, os resultados da acurácia média para discernir pacientes com THB de pacientes saudáveis foi de 61%, enquanto a acurácia de discernimento entre THB e esquizofrenia foi de 66%. O trabalho concluiu que os resultados apresentados sugerem que é possível diferenciar estes pacientes e que essa diferenciação poderia ser agregada aos seus diagnósticos.

Visto que o presente trabalho faz o uso de dados vindos de questionários e não foram encontrados muitos exemplos de trabalhos relacionados abordando o transtorno do humor bipolar com um conjunto de dados similar, foi necessário expandir a pesquisa

para *"machine learning AND mental illness"* e *"machine learning and mental diseases"* com o objetivo de identificar trabalhos que englobam tanto a área de doenças mentais quanto aprendizado de máquina. Dessa forma, foi possível encontrar trabalhos como (JHA *et al.*, 2021), por exemplo, cujo objetivo é identificar um conjunto de fatores que possam indicar uma predisposição ao desenvolvimento de um distúrbio mental durante a pandemia da COVID-19. Para prever esses fatores, foram utilizados dados de 17764 adultos coletados a partir de uma pesquisa sobre o impacto da COVID, que fornece dados dados socioeconômicos tais como cidade, estado, se ficou sem comida no período de entre os meses maio, abril e junho de 2020, se está empregado, idade, sexo, informação se o indivíduo se comunica com os amigos, entre outros dados que resultam de informações com cinco opções para o indivíduo selecionar a qual mais se identifica ou perguntas diretas com respostas de *"SIM"* ou *"NÃO"*. A partir de uma análise estatística inicial seguida pelo uso de redes bayesianas com abordagens clássicas de aprendizado de máquina para a construção do um modelo preditivo, foram identificados os principais fatores que afetam a saúde mental durante a pandemia da COVID e também prever quais participantes cujos indicadores de saúde mental os tornavam mais vulneráveis que os demais com uma acurácia de aproximadamente 80%. Também foi utilizado o teste de independência do qui-quadrado para verificar associações entre indicadores de saúde mental e demais variáveis.

Em (SRIVIDYA; MOHANAVALLI; BHALAJI, 2018), seu objetivo é a criação de uma ferramenta para determinar o estado mental de um indivíduo e auxiliar profissionais na realização de testes e diagnóstico de transtornos mentais de seus pacientes e menciona a importância do diagnóstico precoce de transtornos mentais para que seja possível manter um equilíbrio de vida adequado. Além disso, também é ressaltado que, com o avanço da tecnologia, o papel dos profissionais da saúde pode ser suplementado por modelos que fazem o uso de inteligência artificial. Nesse trabalho, foi usado um questionário contendo 20 questões e um *dataset* com 656 indivíduos e suas respostas, que foi dividido em 80% para o conjunto de treinamento e 20% para o conjunto de testes. Para a construção do modelo, foram usados diferentes algoritmos de classificação: Regressão logística, *Naïve bayes*, Máquinas vetoriais de suporte, Árvore de decisão e K-vizinhos mais próximos. Além esses algoritmos, também foram construídos modelos com *Ensmble bagging* e *Tree ensemble* utilizando *random forest*. As acurácias dos diferentes algoritmos classificadores para prever o estado mental dos pacientes foram: 84% para Regressão logística; 73% para *Naïve bayes*; 89% para SVM; 81% para árvore de decisão; 89% para o K-vizinhos mais próximos; 90% para *Ensemble bagging* e 90% para *Tree Ensemble* utilizando *Random Forest*. Por fim, o trabalho concluiu que considerando os resultados obtidos, é possível utilizar seus modelos como mecanismos auxiliadores para realizar a modelagem comportamental de um grupo de indivíduos.

2.4 CONSIDERAÇÕES FINAIS

A partir dos trabalhos apresentados, percebe-se que o diagnóstico do transtorno do humor bipolar depende de muitos fatores comportamentais e que com o uso de questionários é possível identificá-los, evidenciando a importância da inteligência artificial como ferramenta de apoio.

Em relação aos resultados, os algoritmos *Ensmble bagging* e *Tree ensemble* com *random forest* obtiveram uma acurácia de 90%, o que demonstra o quão promissoras podem ser suas aplicações. Além destes, os outros algoritmos mencionados devem ser considerados, visto que seus resultados também demonstraram potencial. Portanto, a tabela a seguir busca sumarizar as principais características identificadas nos trabalhos apresentados.

Tabela 1 – Trabalhos Relacionados

Trabalho	**Objetivo**	**Dados**	**Algoritmos**	**Resultados**
(SRIVIDYA; MOHANAVALLI; BHALAJI, 2018)	Criar ferramenta para determinar estado mental do paciente	Questionário com 20 perguntas	Regressão logística; *Naïve bayes*; Máquinas vetoriais de suporte; Árvore de decisão; K-vizinhos mais próximos; *Ensmble bagging* e *Tree ensemble* com *random forest*	Regressão logística: 84%; *Naïve bayes*: 73%; SVM: 89%; Árvore de decisão: 81%; K-vizinhos mais próximos: 89%; *Ensemble bagging*: 90%; *Tree Ensemble* com *Random Forest*: 90%
(PEREZ ARRIBAS *et al.*, 2018)	Classificar pacientes pelo transtorno e prever seu humor subsequente	Questionário classificando humor entre ansiedade, euforia, tristeza, raiva, irritabilidade e energia	Regressão utilizando *random forest*	Diagnóstico THB: 82–90% de acurácia Prever o humor: 75% de acurácia
(JADHAV *et al.*, 2019)	Prever THB no paciente	Questionário MDQ	Árvore de decisão	88.07% de acurácia
(JHA *et al.*, 2021)	Prever vulnerabilidade mental em pacientes durante pandemia de COVID	Questionário socioeconômico	redes bayesianas, teste de independência do qui-quadrado	ẽ80% de acurácia
(R.SARANYA, 2022)	Detectar o THB	Dados clínicos e de humor	Redes neurais convolucionais com memória de curto e longo prazo	97.2% de acurácia

Fonte: Elaborada pelo autor (2022).

3 PROPOSTA DE SOLUÇÃO

Este trabalho é desenvolvido a partir de uma metodologia inspirada no processo conhecido como *CRoss Industry Standard Process for Data Mining* (CRISP-DM) (WIRTH; HIPP, 2000), que é utilizada para representar em forma de etapas um projeto que utiliza mineração de dados de maneira que ele seja visto como um ciclo. Essas etapas são: compreensão do problema; compreensão dos dados; preparação dos dados; modelagem; avaliação e implantação.

Neste capítulo, será apresentada uma proposta de solução a partir da metodologia *CRISP-DM* para que o trabalho atinja os objetivos esperados. A Fig.2 ilustra o fluxo planejado.

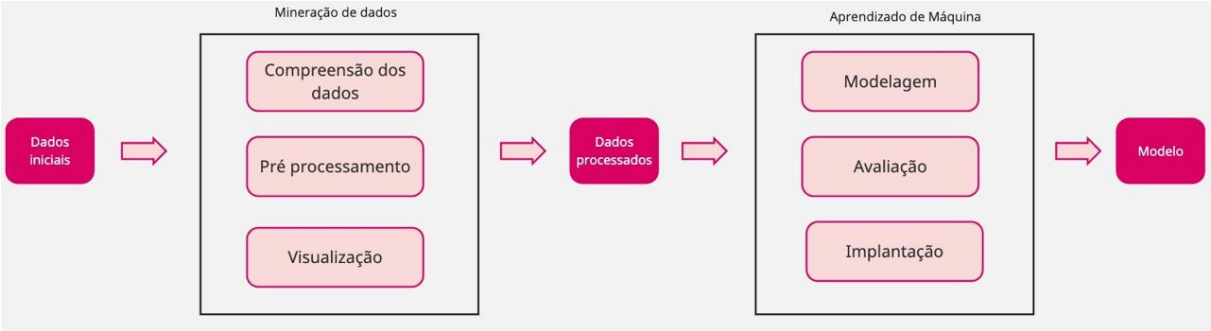

Figura 2 – Etapas do projeto inspiradas na metodologia CRISP-DM

Fonte: Elaborada pelo autor (2022).

Os parágrafos a seguir discutem detalhes do planejamento apresentado e abordam as principais etapas envolvidas neste trabalho.

3.1 PROCESSAMENTO DE DADOS E *FEATURE ENGINEERING*

3.1.1 Dados Utilizados

O grupo de pesquisadores do Programa de Pós-Graduação em Saúde e Comportamento (PPGSC) da Universidade Católica de Pelotas (UCPel) disponibilizou um conjunto de dados obtido a partir de um estudo de coorte transversal, uma modalidade de estudo observacional em ciências sociais e biológicas que analisa dados de um subconjunto representativo da população em um momento específico (COGGON; BARKER; ROSE, 1997). Tais dados contém informações a respeito do perfil de pacientes que passaram por atendimento psicológico, tais como sexo, idade, etnia, episódio de depressão, sentimentos que o paciente possui e marcadores genéticos, se possui ou não o Transtorno do Humor Bipolar (THB). Uma análise mais aprofundada dos dados utilizados será feita na seção 4.3.

3.1.2 Compreensão e Preparação dos Dados

Nesta etapa é necessário compreender como os dados fornecidos estão organizados e categorizados, obtendo informações sobre o *dataset* como número de linhas, tabelas e colunas. Tais informações auxiliarão na parte de análise inicial dos dados para que seja feito um pré-processamento, identificando o número de pacientes, quais os estados de humor com os quais mais se identificam, idade, sexo, dados socioeconômicos, entre outras informações que possam agregar ao trabalho.

Além disso, também é necessário preparar os dados para que eles sejam utilizados na construção do modelo. Por ser um conjunto de dados provenientes de questionário, é esperado que existam falhas em seu preenchimento, sendo assim necessário que haja um tratamento para dados faltantes. Dentre as opções para tratar os dados faltantes estão: eliminar os objetos ou atributos faltantes, ignorar os valores faltantes durante a análise ou estimar (TAN *et al.*, 2018). Ao eliminar ou ignorar tais valores a perda de dados seria considerável, então a imputação dos dados faltantes a partir de uma estimativa se mostrou a melhor opção para evitar perda de registros ou dados. O banco de dados possui tanto dados numéricos quanto categóricos, sendo assim é necessário que o cálculo de estimativas seja diferente para os dois tipos de dados, portanto, será utilizado o cálculo de média para os dados numéricos e moda para os dados categóricos.

Por ser um conjunto de dados clínicos e pelo fato de o Transtorno do Humor bipolar estar presente apenas em 1% a 2% da população adulta (FEARS; REUS, 2015) é esperado que a quantidade de amostras com a coluna de THB marcada como "sim"seja diferente e inferior das marcadas como "não". Portanto, será necessário fazer um balanceamento dos dados para que a incidência dos valores do atributo alvo não interfiram na acurácia do modelo.

Já com as informações primordiais referentes ao *dataset*, será feito um estudo sobre os atributos obtidos e iniciado o processo de *Feature Engineering*. Esse processo é essencial para projetos de aprendizado de máquina pois é incerto o caminho percorrido entre os dados iniciais até a obtenção de respostas. Por conta disso, é importante que eles contenham informações de alta compatibilidade com o contexto da análise que será realizada. Portanto, nessa etapa será feita a padronização dos dados, a filtragem atributos relevantes e que ajudarão a alcançar o objetivo sem que interfiram negativamente nos resultados e também serão criados atributos novos.

3.2 TREINAMENTO, AVALIAÇÃO DE DESEMPENHO E IMPLANTAÇÃO

3.2.1 Modelagem e Treinamento

Com os dados já separados, inicia-se a etapa de modelagem, na qual serão usados três modelos diferentes com o objetivo de prever a ocorrência do Transtorno do Humor bipolar: *Regressão Logística, Árvore de Decisão e XGBoost*. Para seu treinamento, será

dividido o *dataset* para que 80% dele seja usado para o conjunto de treinamento e os 20% restantes para o conjunto de testes. A partir dos modelos prontos, serão analisados os dados de acurácia e serão feitas relações com os modelos que foram utilizados nos trabalhos relacionados para identificar qual que melhor se adéqua ao objetivo do trabalho. Também será feita a otimização dos hiperparâmetros do modelo, buscando quais os melhores parâmetros para cada um deles e medindo o quão relevantes eles são para os resultados finais.

3.2.2 Avaliação

Também será feita a avaliação a partir dos padrões coletados pelo modelo com o objetivo de validar o que foi encontrado e verificar se o comportamento apresentado é o esperado. Caso os resultados não sejam satisfatórios, será necessário reavaliar o projeto a partir da primeira etapa novamente.

4 MINERAÇÃO DE DADOS

Iniciando a etapa de Mineração de dados conforme a solução proposta, foram observados os trabalhos relacionados e consideradas as necessidades de manipulação de arquivos para lidar com o conjunto de dados que foi fornecido pelo grupo de pesquisadores do Programa de Pós-Graduação em Saúde e Comportamento (PPGSC) da Universidade Católica de Pelotas (UCPel), que a linguagem Python se mostrou ideal para que as etapas de processamento, análise e visualização de dados para fossem desenvolvidas, visto que fornece as ferramentas necessárias partir de bibliotecas como Pandas, NumPy e ScikitLearn.

4.1 COMPREENSÃO DOS DADOS

Para acessar os dados, foi necessário carregar o banco de dados com a biblioteca Pandas. A partir dela, foi possível fazer a leitura do banco que está em formato de 'csv' para transformá-lo em uma estrutura de *DataFrame*, que possibilita a contagem e exibição de linhas e colunas. De início foi necessária fazer uma filtragem para selecionar apenas atributos informativos, removendo colunas que representavam indexação, maneira que o dado foi coletado, número de matrícula dos pacientes e nome. Também foram removidas colunas informativas a respeito da coleta de dados, por exemplo, existiam colunas que informavam se o DNA do paciente foi coletado ou não para experimentos ou se o IMC do paciente foi calculado ou não. Colunas preenchidas apenas por pacientes que já possuíam o THB também foram descartadas para a análise, visto que existiam colunas informando qual o episódio atual do paciente, se estava em um episódio de mania, depressão e grau de risco do transtorno. Foram encontradas colunas diferentes com informações repetidas que também foram removidas, assim como colunas que possuíam apenas um valor preenchido. Esse processo resultou em um conjunto de dados com 545 colunas e 808 registros.

Após esta etapa foi necessário fazer uma filtragem em relação às colunas que possuíam muitos dados faltantes, visto que muitas delas estavam totalmente em branco, dessa forma, foram descartadas todas as colunas com mais de 40% de falhas no preenchimento, resultando em 392 colunas. Ao fazer a separação de dados categóricos e dados numéricos surgiu a dificuldade de identificar quais dados pertenciam à qual categoria visto que por serem dados de questionário muitas colunas apresentaram respostas numeradas para representar as respostas das questões. Valores de 1 ou 0 em colunas identificadas como questionários geralmente representam perguntas de sim ou não, porém, em muitas colunas com essa identificação os valores apresentados possuíam muitas variações numéricas, gerando incerteza quanto ao tipo de dado ao qual a coluna se tratava. Por este motivo, foram removidos dados contendo essa característica.

Pelo fato de mesmo com as filtragens iniciais ainda não ser possível fazer uma análise mais profunda de todos os atributos e de como se relacionam com o desfecho por conta

da quantidade de informações, foi necessário limitar o conjunto de dados para facilitar este processo. Portanto, primeiro foram selecionadas informações a respeito dos sentimentos do paciente, sendo elas: Vontade, Raiva, Medo, Cautela, Inibição, Sensibilidade, Coping e Controle. Após isso, foi escolhida uma coluna referente a marcadores biológicos, a coluna SNPLEPR. Também foram escolhidos dados referentes ao sexo, etnia e idade do paciente para entender melhor o perfil deles. Por último, foram selecionados dados de questionários referentes à saúde mental do paciente, contendo sintomas e diagnósticos de outras doenças, que são representados pelas seguintes colunas: H1, I4E, SOMA_DROGASABUSO, episodios_abuso, G1, E6, Episodio_depressao, A4M, A5M e A4MA5M.

Fazendo uma análise do tipo de informação presente em cada coluna, é possível verificar que nas colunas referentes aos sentimentos que o paciente possui a classificação é feita de forma numérica, representando a intensidade de cada sentimento em um intervalo entre 4 e 56. Para as colunas de perfil do paciente, a coluna de sexo é representada pelos valores 'M' ou 'F', já a idade é a própria idade do paciente de maneira numérica e a etnia indica apenas dois valores possíveis: 'branco' ou 'não branco'. Já para os dados de questionário, percebe-se que as colunas A4M e A5M apesar de serem numéricas representam respostas de sim ou não por possuírem valores de 0 ou 1. Para as demais colunas referentes a dados de questionário e para a coluna do marcador biológico, a tabela a seguir mostra as opções presentes em cada uma

Tabela 2 – Dados presentes nas colunas de questionário

Coluna	**Dados presentes**
H1	Não, Fobia Social Atual
I4E	Não, TOC Atual
SOMA_DROGASABUSO	Não, Sim
G1	Não, Agorafobia
E6	Não, Transtorno de Pânico Vida Inteira
episodios_abuso	baixo, mínimo, moderado, severo
Episodio_depressao	controle, depressão bipolar, depressão unipolar
A4MA5M	Depressão, Sem depressão
SNPLEPR	AA, AG, GG

Fonte: Elaborada pelo autor (2023).

4.2 PRÉ-PROCESSAMENTO

Ao todo, nas 22 colunas selecionadas, 13 possuem dados categóricos e 9 numéricos. Com o entendimento dos dados e da análise inicial foi possível visualizar a existência de dados faltantes no *dataset* conforme mostram as figuras Fig.4 e Fig.3, sendo necessário

aplicar técnicas de imputação para tratar os dados a fim de que as falhas no preenchimento que geraram esses dados faltantes não interfiram na acurácia resultados das análises.

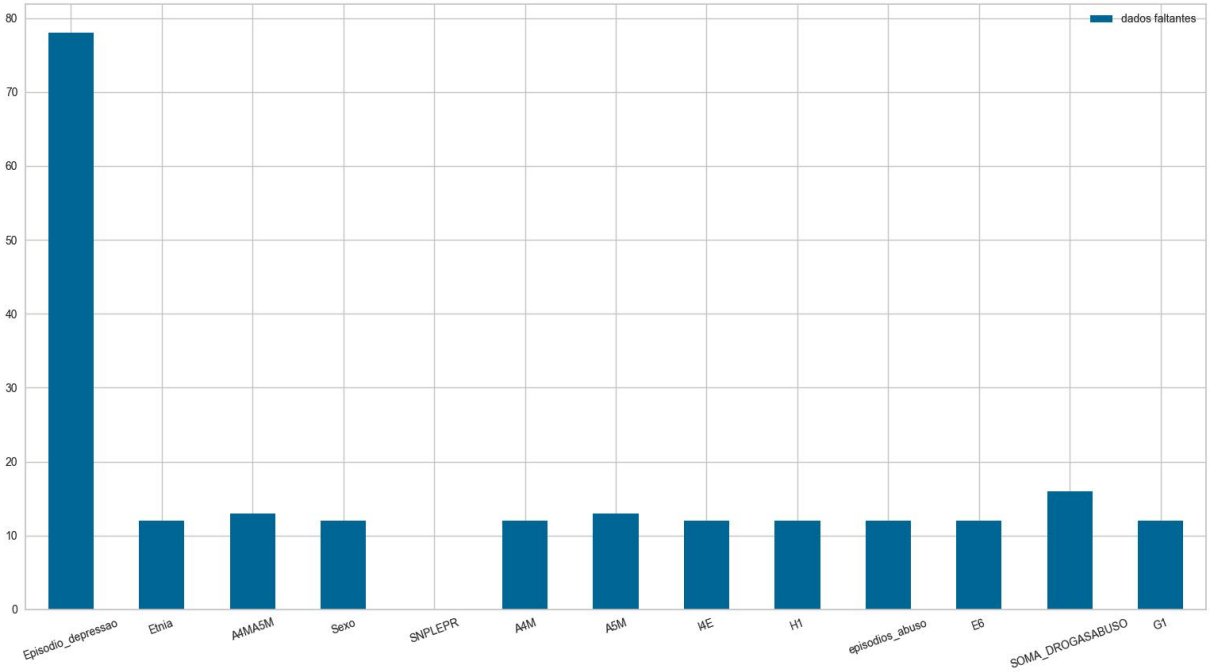

Figura 3 – Quantidade de dados categóricos faltantes

Fonte: Elaborada pelo autor (2023).

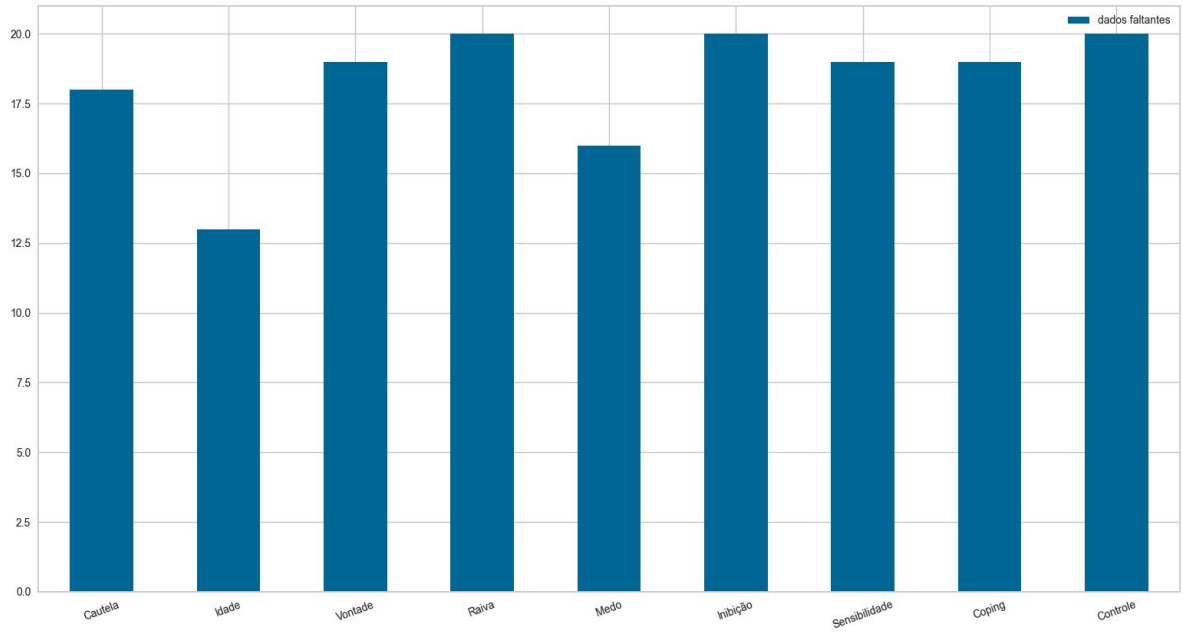

Figura 4 – Quantidade de dados numéricos faltantes

Fonte: Elaborada pelo autor (2023).

Para o pré-processamento também foi necessário ajustar os tipos de dados que estavam presentes no *dataset*. A biblioteca utilizada para modelagem, o *ScikitLearn*, não possui suporte para dados categóricos, tornando necessário tratá-los para que sejam representados de maneira numérica. Para isso, será feito um encodamento dos dados no qual cada valor de uma coluna será representado por um número. Para a coluna de *Sexo*, por exemplo, na qual temos valores M e F, o valor de M passará a ser representado pelo número 1 e o valor de F pelo número 0.

Conforme a seção 3, também se mostra necessário fazer o balanceamento do *dataset* para treinamento do modelo, visto que quando modelo prever que o paciente não possui o THB, ele estará 87.1% das vezes certo pois essa é a porcentagem real de pacientes que não possuem o transtorno no *dataset* utilizado (TAN *et al.*, 2018). Dessa maneira, a acurácia do modelo não representará o seu real poder de predição. O gráfico da figura a seguir ilustra o desbalanço entre a quantidade de pacientes que possuem ou não o THB no *dataset* utilizado. Para contornar este problema biblioteca *ScikitLearn* fornece um parâmetro de *class_weight='balanced'* para ser usado em seus modelos, que ajusta os pesos de cada classe para serem equivalentes a partir de um ajuste para elas sejam inversamente proporcionais às suas frequências nos dados de entrada, o que se dá a partir da seguinte fórmula: *n_amostras / (n_classes * np.bincount(y))* (PEDREGOSA *et al.*, 2011). Após a primeira implementação dos modelos foi verificado que um deles não possuía esse parâmetro, sendo necessário buscar uma nova abordagem para abranger a todos os modelos de maneira igual. Sabe-se que existem duas técnicas de balanceamento para equilibrar os dados, a subamostragem e a sobreamostragem. Enquanto a primeira consiste em diminuir as ocorrências da classe com a maior frequência, a segunda aumenta as com menor ocorrência. (TAN *et al.*, 2018). O método escolhido foi o de sobreamostragem, sendo implementado no código de solução com o auxilio do módulo *RandomOverSampler*, fornecido pela biblioteca *Imblearn* (LEMAÎTRE; NOGUEIRA; ARIDAS, 2017).

Também foi verificada a necessidade de normalização dos dados categóricos pois ao representá-los de forma numérica eles foram transformados para valores em um intervalo de [0, 4]. Isso faz com que os demais dados numéricos já presentes no *dataset* estejam fora de escala por estarem em um intervalo de [0.0, 56.0]. Essa diferença de escalas na representação de números pode gerar instabilidade numérica para algoritmos de aprendizado de máquina, visto que podem enfrentar problemas de precisão numérica, prejudicando sua eficácia. A fim de evitar este problema, a 3 também contempla a etapa de normalização dos dados, que fará o ajuste de todas as colunas numéricas para serem representadas em um intervalo de [0, 1]. A ferramenta utilizada para isso foi o MinMaxScaler, fornecido pelo Scikitlearn (PEDREGOSA *et al.*, 2011).

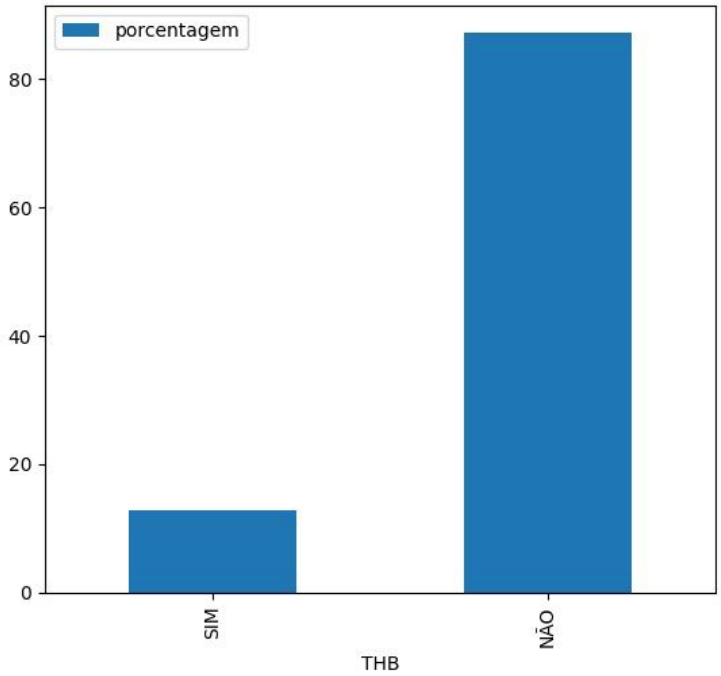

Figura 5 – Porcentagens de pacientes que possuem ou não o THB

Fonte: Elaborada pelo autor (2023).

4.3 ENGENHARIA DE ATRIBUTOS

Para esta etapa foi necessário entender quais variáveis mais se relacionavam com o desfecho deste trabalho. A primeira forma de visualizar isso foi a partir de um mapa de calor mostrando como se dá a correlação entre cada uma das variáveis numéricas com o THB, conforme a Fig. 6. Ao observar os valores, percebe-se que o THB se relaciona mais fortemente com os atributos de raiva e sensibilidade, embora ainda seja uma correlação fraca.

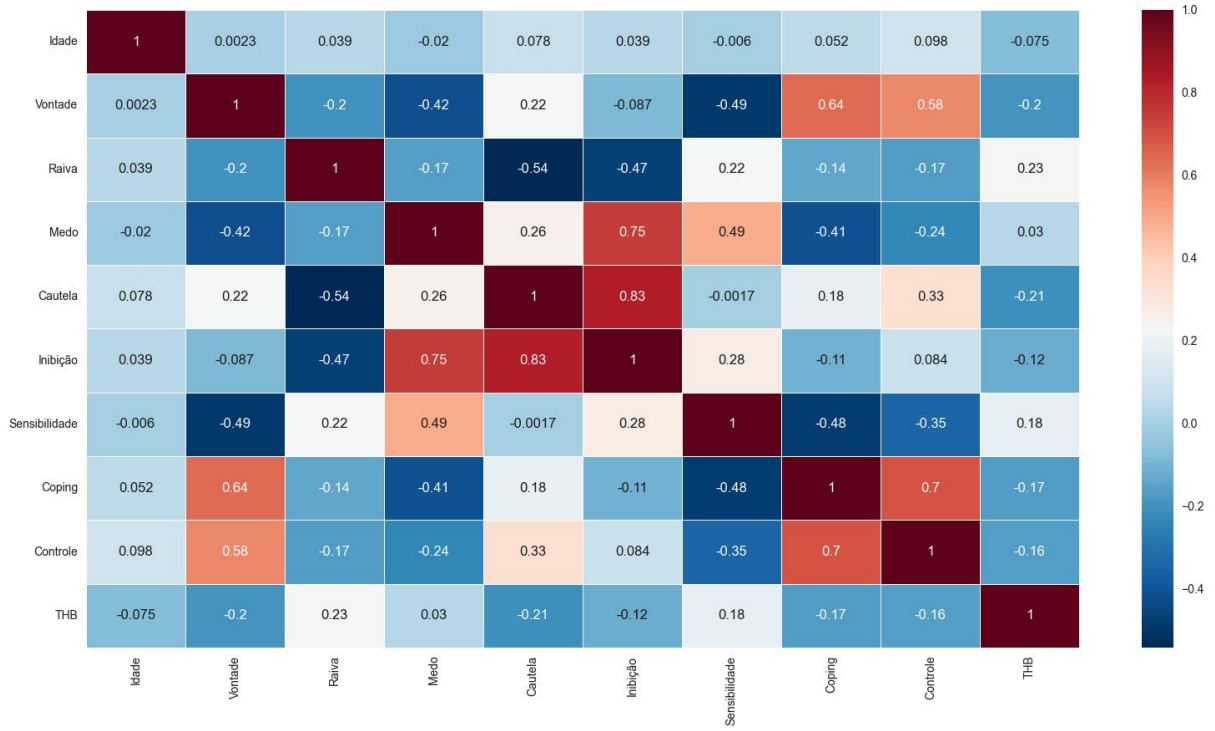

Figura 6 – Mapa de calor para correlação de dados numéricos com o THB

Fonte: Elaborada pelo autor (2023).

Foram criados dois gráficos para os dois atributos numéricos com o qual o desfecho tem maior correlação para uma melhor visualização da densidade de ocorrências THB. Na Fig. 7 é possível observar que ao se aproximar do valor 60 para a raiva há uma maior ocorrência do valor de 'sim' em relação ao 'não'. Na Fig. 8 nota-se que para ambos os valores o pico dos gráficos é atingido aproximadamente no valor 30, porém enquanto o valor de 'não' diminui drasticamente no intervalo de 30 a 60, o valor de 'sim' possui uma leve queda entre 30 e 50 e que se acentua entre 50 e 60.

Já para os dados categóricos a forma de visualizar a interação com o THB foi a partir de gráficos de barras mostrando a distribuição da ocorrência do transtorno do humor bipolar para cada uma delas. Pode-se perceber no gráfico de distribuição para a coluna sexo, por exemplo, que a distribuição entre os dois gêneros considerados para as respostas, 'masculino' e 'feminino', possuem uma porcentagem aproximada equivalente para a ocorrência do THB. Em contrapartida, no gráfico do atributo I4E, a resposta 'TOC atual' apresenta uma porcentagem muito maior de ocorrência do THB do que a resposta 'Não', tendo maior relevância para as análises.

No gráfico da Fig.11, que representa a distribuição do atributo de *Episodio_depressao* é possível observar que em todas as ocorrências do valor de *depressao_bipolar* há ocorrência do THB. Porém, ao observar os gráfico de dados faltantes na tabela da Fig.3 pode-se notar o quanto essa coluna se sobressai em relação às demais,

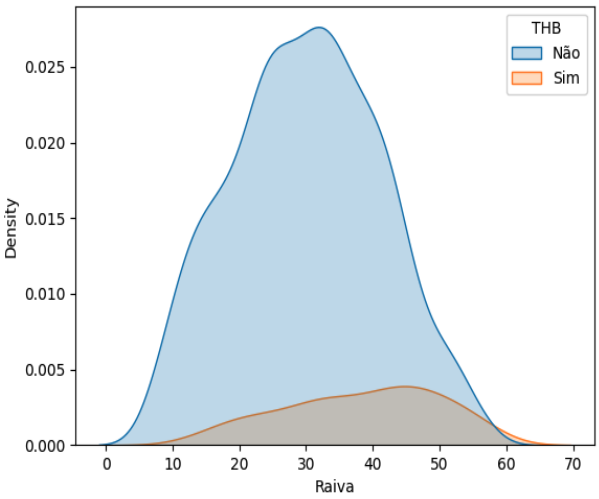

Figura 7 – Distribuição do THB para o atributo 'Raiva'

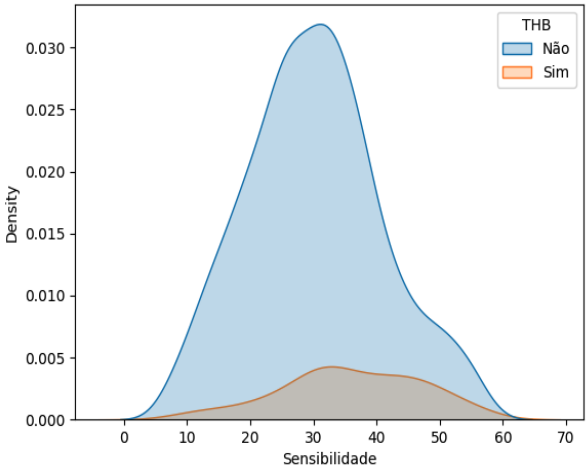

Figura 8 – Distribuição do THB para o atributo 'Sensibilidade'

Fonte: Elaborada pelo autor (2023).

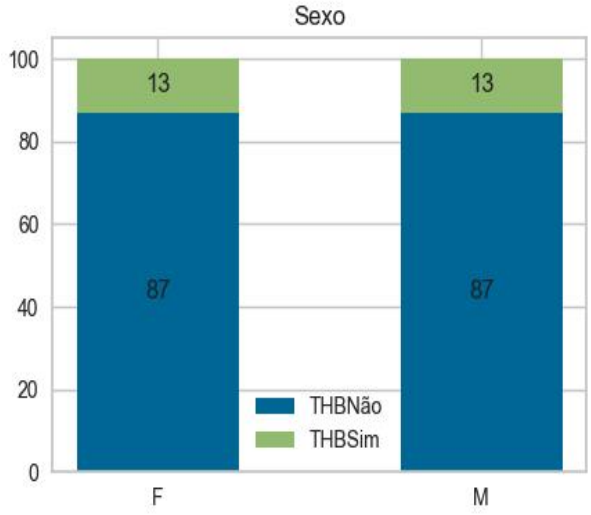

Figura 9 – Distribuição do THB para o atributo 'Sexo'

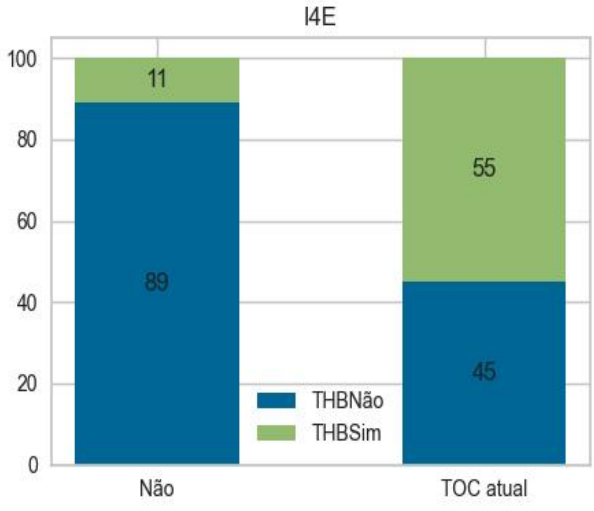

Figura 10 – Distribuição do THB para o atributo 'I4E'

Fonte: Elaborada pelo autor (2023).

possuindo um total de 78 dados faltantes. Destes, 100% estão presentes no valor de *sim* para o THB. Dessa forma, não é possível inferir que sempre que houver *depressao_bipolar* haverá a ocorrência do THB. Por este fato, esta coluna foi removida da análise para que não influencie nos resultados do modelo preditivo.

Também foi utilizado um recurso para a seleção de atributos a partir da utilização de um modelo de *RandomForestRegressor* (PEDREGOSA *et al.*, 2011) que fornece a informação dos atributos de maior importância para o conjunto de dados, gerando o gráfico da Fig.12 e ressaltando o quão mais relevantes são os atributos de *Raiva* e *E6*

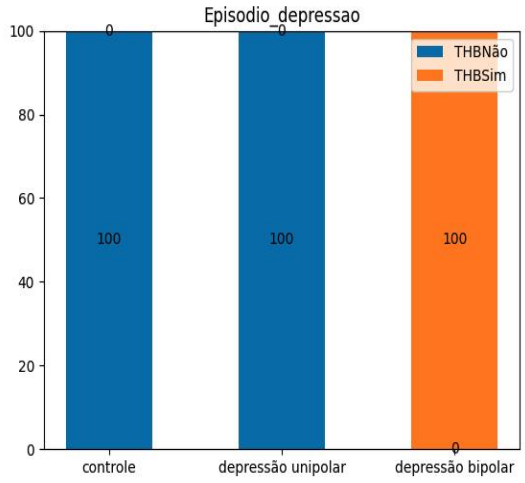

Figura 11 – Distribuição do THB para o atributo 'Episodio_depressao'

Fonte: Elaborada pelo autor (2023).

em relação aos demais e também mostrando o atributo de menor importância: o *H1*. Em seguida foi feita uma eliminação recursiva de atributos com a ferramenta *RFECV* (PEDREGOSA *et al.*, 2011) para que a partir da validação cruzada a seleção de quais atributos utilizar fosse feita. Como resultado, a ferramenta optou por não selecionar apenas o atributo *H1*, que também foi removido do conjunto de dados.

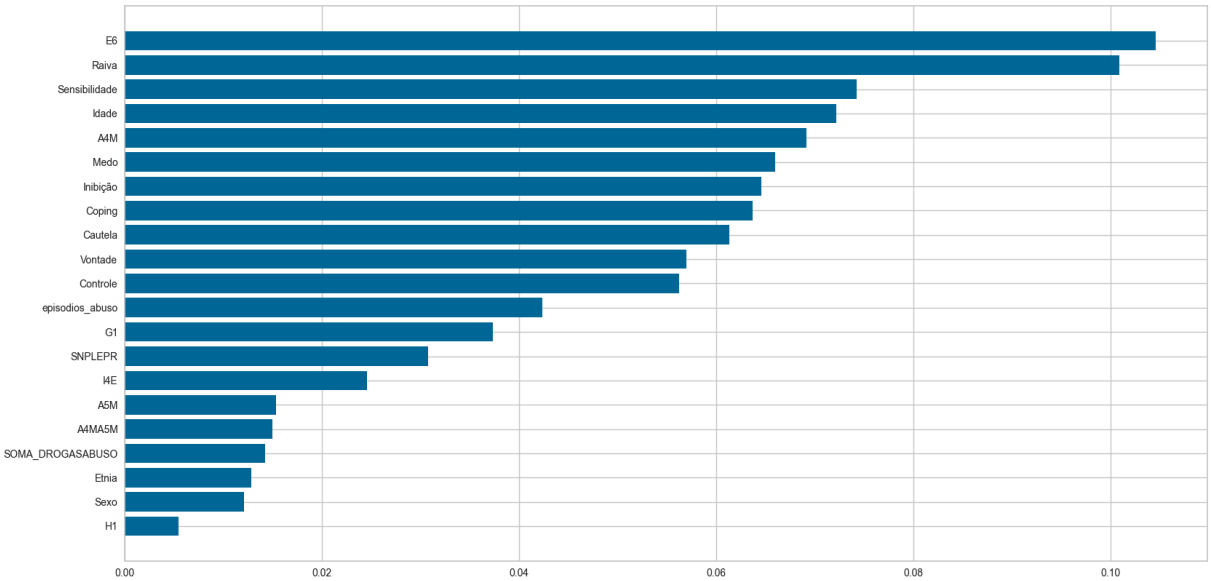

Figura 12 – Gráfico de importância de atributos de acordo com *RandomForestRegressor*

Fonte: Elaborada pelo autor (2023).

Outra etapa da engenharia de atributos consiste na criação de novos atributos a partir dos já existentes. Pelo fato de o *dataset* tratar de dados relacionados á saúde

mental e comportamentos, um conhecimento mais profundo sobre suas relações seria necessário para entender suas interações e a partir disso acrescentar novas informações para o modelo. Em vista disso, foi utilizado um recurso disponibilizado pelo *ScikitLearn* chamado *PolynomialFeatures* (PEDREGOSA *et al.*, 2011), cujo objetivo é gerar uma nova matriz de atributos contendo todas as combinações polinomiais de acordo com o grau especificado. Por exemplo, se uma amostra de entrada for bidimensional e da forma [a, b], os recursos polinomiais de grau 2 serão

$$1, a, b, a^2, ab, b^2$$

. O conjunto de dados utilizado para este trabalho possui 20 atributos que serão utilizados e após aplicar o *PolynomialFeatures*, esses atributos passam a ser 231.

Apesar de não haver conhecimento para uma análise mais aprofundada, ainda foi feita uma tentativa de criar novos atributos como uma alternativa à *PolynomialFeatures*. Observando o gráfico da Fig.12 de importância de atributos, percebe-se a relevância de raiva e sensibilidade. Por serem dados contínuos, é possível classificá-los em intervalos para que a noção de intensidade seja adicionada às características dos dados. Dessa maneira, foram criados dois atributos: *Intensidade-Sensibilidade* e *Intensidade-Raiva*, ambos possuindo os valores *pouca*, *média* e *alta*. A tabela da 3 mostra os intervalos presentes em raiva e sensibilidade que foram considerados para a criação dos novos valores para as duas novas colunas.

Também foi criado um novo atributo chamado *jovem-adulto* para as informações referentes à idade. Pôde-se observar que as idades presentes nos dados estão entre 15 e 36 anos. Dessa forma, foi criada uma nova coluna a partir dela que contém o valor *jovem* para pacientes com idades entre 15 e 25 anos e como *adulto* para os pacientes com idades entre 25 e 36 anos.

Tabela 3 – Campos criados para as colunas de Intensidade-Sensibilidade e Intensidade-Raiva

Intervalo	**Valor**
0 a 19	pouca
20 a 40	média
40 a 56	muita

Fonte: Elaborada pelo autor (2023).

Após a inserção das novas colunas ao *dataset* foi usado o mesmo método mencionado anteriormente para verificar e visualizar a importância dos atributos. Pode-se perceber na Fig. 13 que os atributos criados ficaram com uma importância baixa em relação aos demais, especialmente *Intensidade-Sensibilidade* e *jovem-adulto*. Já o atributo *Intensidade-Raiva* se saiu um pouco melhor do que um dos atributos originais do conjunto de dados, porém ainda assim com pouca relevância.

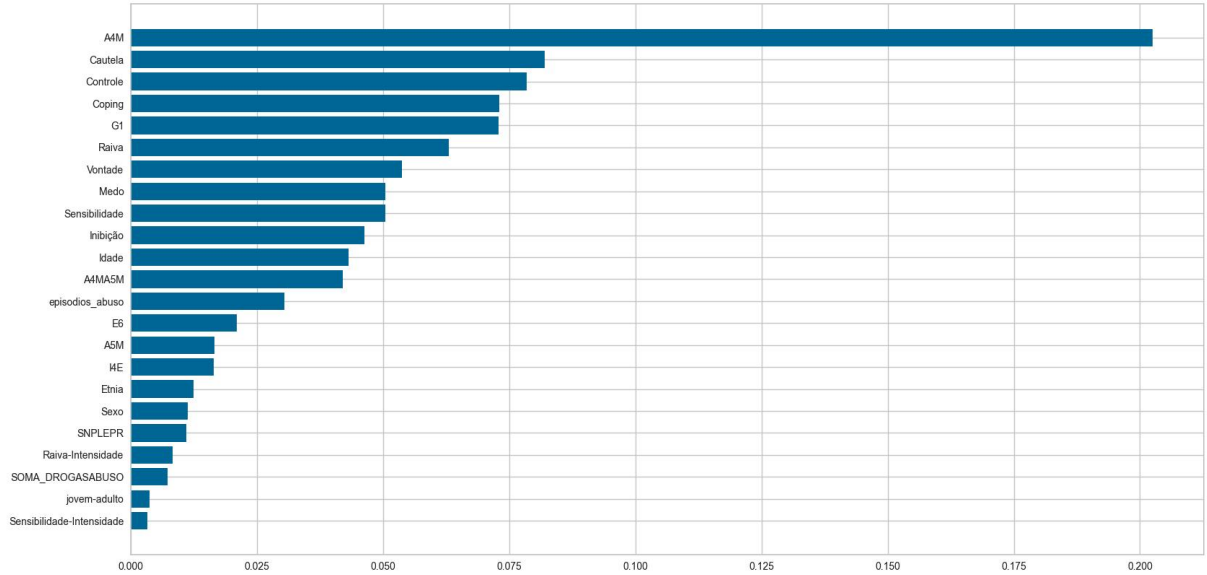

Figura 13 – Gráfico de importância de atributos após novos dados

Fonte: Elaborada pelo autor (2023).

5 APRENDIZADO DE MÁQUINA

5.1 MODELOS E TREINAMENTO

Para iniciar a etapa de modelagem foi separado 20% do *dataset* para testes do modelo, deixando os 80% restantes para o treinamento, padrão este que será aplicado para todos os modelos que serão apresentados neste capítulo. Para fazer a divisão desses conjuntos foi utilizado o método de *train_test_split* da biblioteca ScikitLearn (PEDREGOSA *et al.*, 2011), sendo X os atributos que serão utilizados para fazer as predições e y o conjunto de dados referentes ao desfecho, o THB. Como resultado, *x_train* e *y_train* representam os atributos e desfecho do modelo de treinamento enquanto *x_test* e *y_test* representam-nos para o modelo de testes.

```
% x_train, x_test, y_train, y_test =
%                 train_test_split(X, y, test_size=0.2, random_state=1)
```

Utilizando esses dados, os três modelos definidos para este trabalho em 3 foram treinados de maneira progressiva com o objetivo de isolar os comportamentos de cada etapa aplicada em 4 para que seja possível visualizar e comparar a interferência de cada uma delas nos resultados dos modelos. Com o mesmo objetivo, os modelos foram treinados primeiramente com seus atributos padrão para posteriormente ser feito um ajuste de hiperparâmetros. Os modelos utilizados foram o *Gradient Boosting Classifier* (GBC), Regressão Logística (LR) e Árvore de Decisão (DT), todos fornecidos pela biblioteca *ScikitLearn* (PEDREGOSA *et al.*, 2011). Além do conjunto de dados da UcPel os modelos também foram treinados e avaliados com um outro *dataset* que utiliza dados provenientes de questionários e que dizem a respeito de saúde mental. O objetivo disso foi validar os passos feitos até então, passando desde o pré processamento até a otimização de hiperparâmetros e verificar se possuem eficácia para um conjunto de dados diferente mas com o mesmo tipo de dados e desfecho similar.

5.2 AVALIAÇÃO E RESULTADOS

Durante a primeira etapa da avaliação não foi feito nenhum tipo de ajuste relativo ao pré processamento e engenharia de atributos para os conjuntos de treino e teste.

A tabela 4 mostra os resultados de acurácia, f1 e precisão obtidos para os três modelos quando treinados com seus valores padrão e sem nenhum ajuste de pré processamento ou engenharia de atributos. Neles, pode-se perceber que todos os valores de precisão foram 1.0 e as demais métricas permaneceram altas também, com exceção de F1 para LR. De maneira geral, esses valores altos indicam que existem dados tendenciando o modelo.

Analisando a Árvore de Decisão é possível verificar os caminhos que ela seguiu até chegar em uma tomada de decisão, nota-se que na Fig. 15 apenas o atributo *Episodio_depressao* foi utilizado, o que significa que está tendenciando o modelo visto que

Tabela 4 – Acurácia dos modelos de teste sem ajuste

Score	**Árvore de decisão**	**Regressão Logística**	**Gradient Boosting Classifier**
Acurácia	1.0	0.9675	1.0
F1	1.0	0.4000	1.0
Precisão	1.0	1.0	1.0

Fonte: Elaborada pelo autor (2023).

sozinho ele já possibilita a separação entre os casos que possuem ou não o THB. Ele foi mencionado em 4.3 como uma possível causa de tendência visto que possui muitos dados faltantes para um determinado valor do desfecho.

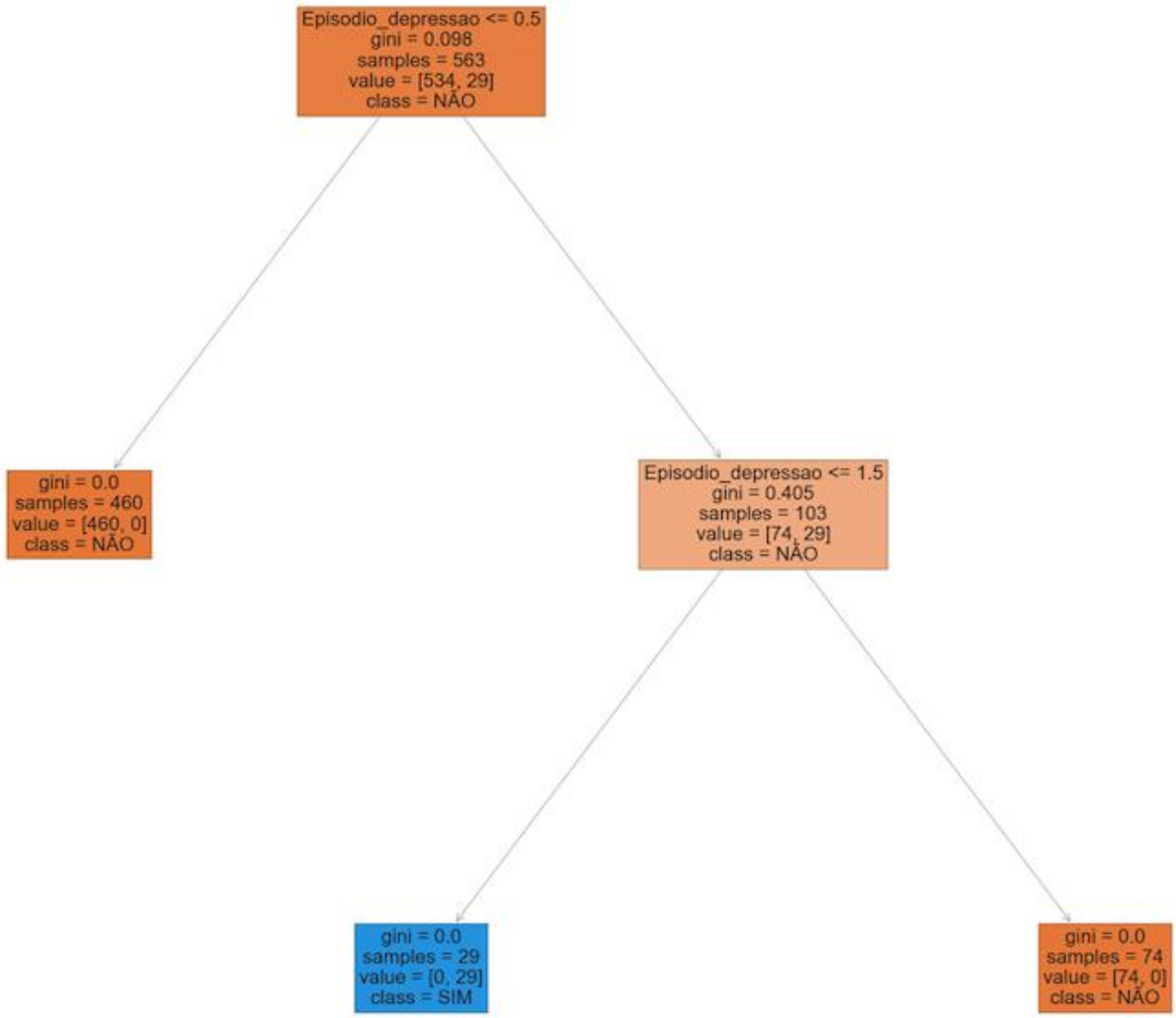

Figura 14 – Árvore de decisão

Fonte: Elaborada pelo autor (2023).

Por este motivo, essa etapa foi reavaliada excluindo este atributo, visto que os resultados não se mostraram confiáveis. Observando a tabela 5, estão os novos resultados

obtidos.

Tabela 5 – Acurácia dos modelos de teste sem ajuste após remoção de *Episodio_depressao*

Score	**Árvore de decisão**	**Regressão Logística**	**Gradient Boosting Classifier**
Acurácia	0.8692	0.8627	0.8627
F1	0.4700	0.3225	0.3345
Precisão	0.6000	0.6250	0.6360

Fonte: Elaborada pelo autor (2023).

Analisando os caminhos feitos pela nova Árvore de decisão nessa avaliação do modelo pode-se perceber que foram feitos mais caminhos até que o modelo chegasse a uma conclusão. Diferentes atributos foram considerados como os mais relevantes, destacando-se o atributo *A4M*, que se encontra no nó raiz da árvore com um *gini* de 0.203, seguido pelo atributo *E6* e *Raiva*.

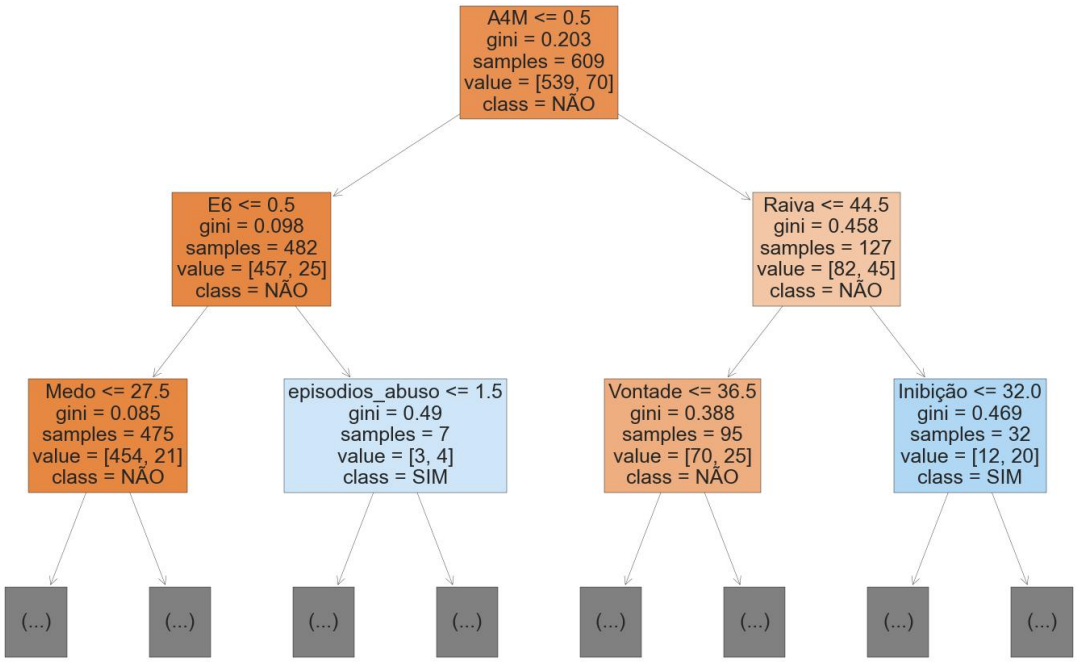

Figura 15 – Árvore de decisão sem o atributo *Episodio_depressao*

Fonte: Elaborada pelo autor (2023).

Também foram coletados os resultados do modelo de árvore de decisão quando utilizados os conjuntos de treino. Pôde-se perceber que para as três métricas avaliadas os resultados foram de 1.0, indicando que pode há um *overfitting*, ou seja, o modelo aprendeu demasiadamente os dados que foram fornecidos para que ele treinasse suas

decisões, fazendo com que ele tenha dificuldade de lidar com dados que sejam diferentes do treino visto que aprendeu tanto os padrões deste conjunto que ao tentar aplicá-los para um semelhante acaba se apegando à nuanças ou variações aleatórias ao invés de padrões de relacionamento importantes. (MITCHELL, 1997)

Tabela 6 – Resultados de treino para Árvore de decisão

Score de treino	**Árvore de decisão**
Acurácia	1.0
F1	1.0
Precisão	1.0

Fonte: Elaborada pelo autor (2023).

A próxima avaliação é para identificar como o pré processamento afeta os resultados, sendo assim, os modelos foram treinados com seus parâmetros padrão com os dados passando pela etapa de imputação. A tabela 7 a seguir mostra os resultados obtidos para os três modelos. Ao fazer um comparativo com os resultados dos modelos sem nenhum ajuste 5 verifica-se uma redução em todas as métricas para todos os modelos, exceto em F1 para LR que teve um pequeno aumento.

Tabela 7 – Acurácia dos modelos de teste com pré processamento e sem engenharia de atributos

Score	**Árvore de decisão**	**Regressão Logística**	**Gradient Boosting Classifier**
Acurácia	0.8000	0.8500	0.8500
F1	0.3200	0.3333	0.3684
Precisão	0.3200	0.4615	0.4666

Fonte: Elaborada pelo autor (2023).

Também foi gerada uma representação da Árvore de Decisão para esta etapa, sendo que para melhor visualização, ela está com profundidade 2 na Fig. 16, visto que sua profundidade real atingiu um valor de 13, ou seja, em seu pior cenário percorreu 13 caminhos até encontrar uma decisão. Nessa nova árvore, percebe-se que o atributo *Raiva* não estava mais no topo como na anterior. No lugar dele, foi utilizado o atributo *G1*.

A engenharia de atributos também foi verificada, sendo a tabela 8 a sumarização dos resultados. Observa-se que o modelo de *Gradient Boosting Classifier* teve um ganho em todas as suas métricas, se beneficiando bastante com esta etapa, enquanto a Regressão Logística apresentou redução. Já para a Árvore de Decisão, houve um leve ganho para acurácia e precisão.

Na seção 4 é evidenciada a importância do balanceamento para que isso não influencie na acurácia do modelo. Portanto, os modelos também foram ajustados para considerar as classes de maneira balanceada, vale ressaltar que este processo foi feito

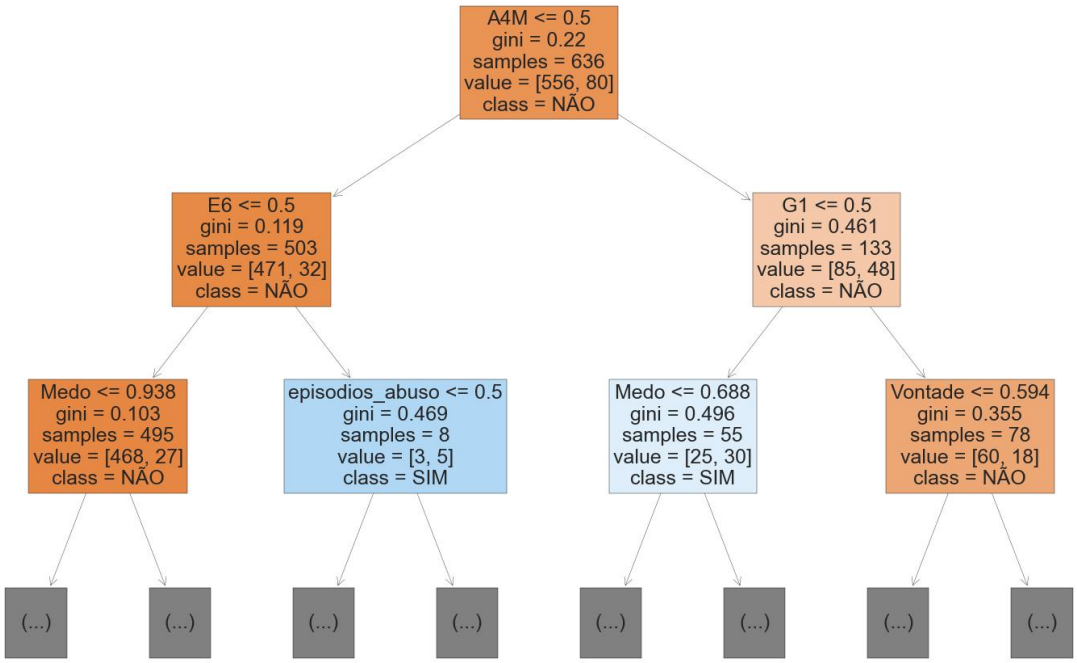

Figura 16 – Árvore de decisão

Fonte: Elaborada pelo autor (2023).

Tabela 8 – Acurácia dos modelos de teste com pré processamento e com engenharia de atributos

Score	**Árvore de decisão**	**Regressão Logística**	**Gradient Boosting Classifier**
Acurácia	0.8125	0.8375	0.8687
F1	0.2857	0.3157	0.4000
Precisão	0.3333	0.4228	0.6363

Fonte: Elaborada pelo autor (2023).

apenas no conjunto de treinamento dos modelos. Assim, os resultados obtidos com o uso do balanceamento obtidos estão na tabela 9. Comparando-os com os resultados sem balanceamento houve aumento de F1 para *Gradient Boosting Classifier*, enquanto acurácia, precisão tiveram uma redução. Para Árvore de Decisão, houveram ganhos em F1 e precisão, enquanto a acurácia teve uma leve queda. Já para Regressão Logística, F1 apresentou ganhos e as outra métricas reduziram.

Utilizar os modelos com parâmetros padrão fornecidos pela biblioteca funciona de maneira generalizada, ou seja, atendem uma variedade grande de problemas porém não possuem a capacidade de se adaptar especificamente para cada caso. Para isso foi aplicada uma técnica de ajuste de hiperparâmetros que a partir de combinações de valores

Tabela 9 – Acurácia dos modelos de teste com pré processamento e balanceamento

Score	**Árvore de decisão**	**Regressão Logística**	**Gradient Boosting Classifier**
Acurácia	0.8072	0.7875	0.8312
F1	0.3240	0.4137	0.4705
Precisão	0.3434	0.3529	0.4444

Fonte: Elaborada pelo autor (2023).

e validação cruzada objetiva encontrar os parâmetros que otimizem o desempenho e resultados do modelo. A biblioteca *ScikitLearn* (PEDREGOSA *et al.*, 2011) fornece uma ferramenta com esse fim, o *GridSearchCV*. Para utilizá-lo, foi necessário separar para cada modelo uma estrutura contemplando todas as variações desejadas de cada parâmetro escolhido para que a ferramenta conseguisse encontrar dentre as opções qual possuía o resultado mais satisfatório. A tabela 10 mostra quais os parâmetros e variações escolhidas.

Tabela 10 – Parâmetros e valores escolhidos

Árvore de decisão	
Parâmetro	**Valores**
ccp_alpha	[0.1, 0.01, 0.001]
max_depth	[5, 6, 7, 8, 9, None]
max_features	['sqrt', 'log2', None]
Regressão logística	
Parâmetro	**Valores**
penalty	['elasticnet', None, 'l1', 'l2']
C	[0.1, 0.5, 1.0, 1.5, 2.0]
tol	[1e-5, 1e-4, 1e-3, 1e-2]
solver	['lbfgs', 'newton-cg', 'newton-cholesky', 'sag', 'saga', 'liblinear']
max_iter	[200, 300, 400, 500, 600, 700]
Gradient Boosting Classifier	
Parâmetro	**Valores**
learning_rate	[0.025, 0.05, 0.075, 0.1, 0.15, 0.2]
max_depth	[1, 2, 3, 4, 5]
max_features	["log2", "sqrt", None]
n_estimators	[50, 100, 150, 200]

Fonte: Elaborada pelo autor (2023).

Para cada um dos modelos foi criado um objeto de *GridSearchCV* conforme mostra o trecho de código a seguir, sendo *estimator* um objeto que representa cada um dos três modelos escolhidos e construídos da maneira padrão, *param_grid* uma estrutura de dicionário com todos os parâmetros escolhidos e seus respectivos valores, e *scoring* a métrica escolhida para ser avaliada.

Tabela 11 – Resultados GridSearchCV para melhores resultados de F1

Modelo	**Parâmetros**
Árvore de Decisão	ccp_alpha=0.001, max_depth=None, max_features='sqrt'
Regressão Logística	C=2.0, max_iter=600, penalty='l1', solver='libnear', tol=0.01
Gradient Boosting Classifier	learning_rate=0.1, max_depth=5, max_features='sqrt', n_estimators=200

Fonte: Elaborada pelo autor (2023).

Os modelos foram treinados novamente com os parâmetros encontrados para verificar os resultados obtidos para o modelo de testes, sendo necessário realizar um aumento no parâmetro *ccp_alpha* da árvore de decisão e uma diminuição do **learning_rate** e **max_depth** do GBC para evitar o *overfitting*. Sendo assim, os resultados finais dos modelos após passarem por todas as etapas estão resumidos na tabela 12

Os resultados finais da análise para o segundo conjunto de dados avaliado foram superiores, evidenciando que existe uma melhoria necessária no processo de engenharia de atributos para o primeiro conjunto de dados com o objetivo de que novos atributos com maior relevância sejam criados a partir dos já existentes para que possam melhorar os resultados obtidos nas predições dos modelos.

Tabela 12 – GridSearchCV ajustado para evitar overfitting

Modelo	**Parâmetros**	**Resultados**
Árvore de Decisão	ccp_alpha=0.002, max_depth=None, max_features='sqrt'	**Acurácia**: 0.8125 teste e 0.9802 treino **F1**: 0.4444 teste e 0.9805 treino **Precisão**: 0.4 teste e 0.9619 treino
Regressão Logística	C=2.0, max_iter=500, penalty='l1', solver='libnear', tol=0.01	**Acurácia**: 0.8312 teste e 0.9514 treino; **F1**: 0.4489 teste e 0.9522 treino; **Precisão**: 0.44 teste e 0.9372 treino;
Gradient Boosting Classifier	learning_rate=0.01, max_depth=5, max_features='sqrt', n_estimators=200	**Acurácia**: 0.8562 teste e 0.9873 treino; **F1**: 0.4651 teste e 0.9873 treino; **Precisão**: 0.5263 teste e 0.9846 treino;

Fonte: Elaborada pelo autor (2023).

6 CONSIDERAÇÕES FINAIS

6.1 CONCLUSÃO

Neste trabalho foram introduzidos os conceitos do Transtorno do Humor bipolar e das técnicas Mineração de dados e Aprendizado de Máquina. Também foi explicado seu objetivo: utilizar essas técnicas em um conjunto de dados fornecido pelo grupo de pesquisadores da Universidade Católica de Pelotas para prever a ocorrência do THB.

O estudo de literaturas existentes para prever o THB com o uso de questionários e técnicas de mineração de dados e aprendizado de máquinas demonstrou resultados satisfatórios e com uma boa acurácia a partir dos métodos utilizados, evidenciando a possibilidade de prever o THB a partir de questionários utilizando os algoritmos mencionados.

Com isso, a proposta de solução considerou a metodologia CRISP-DM para que o ciclo de suas etapas fosse seguido no desenvolvimento do projeto. O entendimento de dados, pré processamento e engenharia de atributos foram descritos na seção 4 que mostrou todo o processo de separação dos dados que seriam utilizados além de mostrar como eles foram tratados para serem representados todos de maneira numérica e as técnicas de imputação adotadas. Também descreveu o processo de engenharia de atributos, que se mostrou um pouco mais complexo por conta da dificuldade em relacionar os atributos já existentes para a criação de novos, porém, apesar da dificuldade foram criados três novos atributos e utilizada a ferramenta de PolynomialFeatures para gerar novos a partir de combinações. As etapas da mineração de dados tiveram seus impactos medidos na seção 5, demonstrando que o pré processamento teve um impacto positivo nos resultados dos modelos.

Por fim, o melhor F1 encontrado após todas as etapas definidas na seção 3 foi para o modelo Gradient Boosting Classifier atingindo um valor de 47,05%, juntamente com uma acurácia de 83,12%. Apesar de todas as etapas terem sido seguidas e o melhor modelo encontrado ter passado por todas elas, ao analisar os resultados percebe-se que ainda há espaço para que melhorias nos processos sejam feitas, sugerindo uma melhoria no processo de engenharia de atributos principalmente.

Ao fazer uma comparação com os trabalhos relacionados de mesmo desfecho, em (JADHAV *et al.*, 2019) não são apresentados resultados referentes à outras métricas que não sejam acurácia, limitando as comparações. Já em (R.SARANYA, 2022), o trabalho apresenta as mesmas métricas, sendo que seus resultados foram mais promissores, porém, existem diferenças entre o tipo de dado utilizado para diagnosticar o THB, visto que o presente trabalho usa o diagnóstico de sim ou não, o trabalho relacionado utiliza o tipo de episódio depressivo no qual o paciente se encontra, o que pode explicar a diferença entre os resultados. A tabela 13 mostra a comparação entre os valores dos trabalhos mencionados.

Tabela 13 – Comparação de resultados com os trabalhos relacionados

Score	**Este trabalho**	**(JADHAV *et al.*, 2019)**	**(R.SARANYA, 2022)**
Acurácia	83,12%	88,07%	96%
F1	44,44%	Não menciona	95%
Precisão	44,44%	Não menciona	96%

Fonte: Elaborada pelo autor (2023).

6.2 MELHORIAS E TRABALHOS FUTUROS

Apesar de os resultados de acurácia apresentados no trabalho terem sido satisfatórios, o fato de os resultados de F1 e precisão estarem baixos indica que é necessária uma melhora no processo de engenharia de atributos, o que foi evidenciado ao testar o modelo com um conjunto de dados diferente. Essa foi a maior dificuldade encontrada no decorrer do desenvolvimento deste trabalho visto que, por lidar com dados de questionário que tratam de padrões comportamentais, transtornos, traumas e sentimentos, é necessário que um profissional especializado seja consultado para entender melhor como se dão as interações entre eles para que novos atributos relevantes sejam criados.

Também foi observada a presença de *overfitting* nos resultados, que também pode ser amenizado com uma engenharia de atributos mais elaborada. Portanto, seria interessante trabalhos que se aprofundem na questão de como buscar uma melhora na engenharia de atributos para dados de questionários para prever o THB e verificar se com isso o *overfitting* pode ser evitado ou se seriam necessárias outras técnicas.

REFERÊNCIAS

AGGARWAL, Charu C *et al.* **Data mining: the textbook**. [*S.l.*]: Springer, 2015. v. 1.

AMERICAN PSYCHIATRIC ASSOCIATION, DS;
ASSOCIATION, American Psychiatric *et al.* **Diagnostic and statistical manual of mental disorders - DSM-5**. [*S.l.*: *s.n.*], 2013. v. 21, p. 591–643.

AZAR, Ghassan *et al.* Intelligent data mining and machine learning for mental health diagnosis using genetic algorithm. *In:* IEEE. 2015 IEEE International Conference on Electro/Information Technology (EIT). [*S.l.*: *s.n.*], 2015. P. 201–206.

CHEN, Tianqi; GUESTRIN, Carlos. XGBoost: A Scalable Tree Boosting System. *In:* PROCEEDINGS of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. San Francisco, California, USA: ACM, 2016. (KDD '16), p. 785–794. DOI: 10.1145/2939672.2939785. Disponível em: http://doi.acm.org/10.1145/2939672.2939785.

COGGON, David; BARKER, David; ROSE, Geoffrey. **Epidemiology for the Uninitiated**. [*S.l.*]: BMJ (British Medical Journal), 1997. Chapter 8, "Case-control and cross-sectional studies".

FEARS, Scott C.; REUS, Victor I. Chapter 104 - Bipolar Disorder. *In:* ROSENBERG, Roger N.; PASCUAL, Juan M. (Ed.). **Rosenberg's Molecular and Genetic Basis of Neurological and Psychiatric Disease (Fifth Edition)**. Fifth Edition. Boston: Academic Press, 2015. P. 1275–1291. ISBN 978-0-12-410529-4. DOI: https://doi.org/10.1016/B978-0-12-410529-4.00104-2. Disponível em: https://www.sciencedirect.com/science/article/pii/B9780124105294001042.

GRANDE, Iria *et al.* Bipolar disorder. **The Lancet**, Elsevier, v. 387, n. 10027, p. 1561–1572, 2016.

HASTIE, Trevor; TIBSHIRANI, Robert; FRIEDMAN, Jerome. **The Elements of Statistical Learning**. New York, NY, USA: Springer New York Inc., 2001. (Springer Series in Statistics).

HILBE, Joseph M. **Practical Guide to Logistic Regression**. Arizona, USA: Taylor Francis Group, 2015. (Springer Series in Statistics).

JADHAV, Ranjana *et al.* Mental disorder detection: Bipolar disorder scrutinization using machine learning. *In:* IEEE. 2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence). [*S.l.*: *s.n.*], 2019. P. 304–308.

JHA, Indra Prakash *et al.* Learning the mental health impact of covid-19 in the united states with explainable artificial intelligence: Observational study. **JMIR mental health**, JMIR Publications Inc., Toronto, Canada, v. 8, n. 4, e25097, 2021.

LEMAÎTRE, Guillaume; NOGUEIRA, Fernando; ARIDAS, Christos K. Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning. **Journal of Machine Learning Research**, v. 18, n. 17, p. 1–5, 2017. Disponível em: `http://jmlr.org/papers/v18/16-365.html`.

MITCHELL, Tom. **Machine Learning. 1. ed**. [*S.l.*]: McGraw-Hill, 1997.

MÜLLER-OERLINGHAUSEN, Bruno; BERGHÖFER, Anne; BAUER, Michael. Bipolar disorder. **The Lancet**, Elsevier, v. 359, n. 9302, p. 241–247, 2002.

PEDREGOSA, F. *et al.* Scikit-learn: Machine Learning in Python. **Journal of Machine Learning Research**, v. 12, p. 2825–2830, 2011.

PEREZ ARRIBAS, Imanol *et al.* A signature-based machine learning model for distinguishing bipolar disorder and borderline personality disorder. **Translational psychiatry**, Nature Publishing Group, v. 8, n. 1, p. 1–7, 2018.

R.SARANYA, Dr.S. Niraimathi. BD-MDL: BIPOLAR DISORDER DETECTION USING MACHINE LEANRING AND DEEP LEARNING. **Journal of Pharmaceutical Negative Results**, v. 13, p. 5892–5905, 2022.

RUSSEL, STUART J., NORVIG, Peter. **Artificial Intelligence: A Modern Approach.** New Jersey, 2010.

SCHNACK, Hugo G *et al.* Can structural MRI aid in clinical classification? A machine learning study in two independent samples of patients with schizophrenia, bipolar disorder and healthy subjects. **Neuroimage**, Elsevier, v. 84, p. 299–306, 2014.

SOKOLOVA, Marina; LAPALME, Guy. A systematic analysis of performance measures for classification tasks. **Information Processing & Management**, Elsevier, v. 45, n. 4, p. 427–437, 2009.

SRIVIDYA, M; MOHANAVALLI, Subramaniam; BHALAJI, Natarajan. Behavioral modeling for mental health using machine learning algorithms. **Journal of medical systems**, Springer, v. 42, n. 5, p. 1–12, 2018.

STUART RUSSELL, Peter Norvig. **Artificial Intelligence: A Modern Approach**. 1st. [*S.l.*]: Prentice Hall, 1995. ISBN 0131038052; 9780131038059.

TAN, Pang-Ning *et al.* **Introduction to Data Mining**. [*S.l.*]: Pearson, 2018.

WIRTH, Rüdiger; HIPP, Jochen. CRISP-DM: Towards a standard process model for data mining. *In:* MANCHESTER. PROCEEDINGS of the 4th international conference on the practical applications of knowledge discovery and data mining. [*S.l.*: *s.n.*], 2000. P. 29–40.

ZOABI, Yazeed; DERI-ROZOV, Shira; SHOMRON, Noam. Machine learning-based prediction of COVID-19 diagnosis based on symptoms. **npj digital medicine**, Nature Publishing Group, v. 4, n. 1, p. 1–5, 2021.

Apêndices

APÊNDICE A – CÓDIGO FONTE

A.1 CÓDIGO PRINCIPAL

```
import pandas as pd
import seaborn as sb
import numpy as np
import matplotlib.pyplot as plt
import warnings

from imputation import imputation_numeric_data,
    imputation_categorical_data
from imblearn.over_sampling import RandomOverSampler
from constants import *

from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestRegressor,
    GradientBoostingClassifier
from sklearn.feature_selection import RFECV
from sklearn import metrics, tree
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.linear_model import LogisticRegression
from sklearn.preprocessing import MinMaxScaler

from matplotlib.ticker import MaxNLocator
from sklearn.impute import KNNImputer
from sklearn.preprocessing import PolynomialFeatures

plt.rcParams.update({'font.size': 25})



def read_database():
    return pd.read_csv(DATASET_PATH)


def encode_data(df):
    new_df = pd.DataFrame()
    for column in TO_ENCODE:
        new_df[column] = df[column].astype('category').cat.codes

    df.drop(columns=TO_ENCODE)
    for column in TO_ENCODE:
        new_df = new_df[new_df[column] != -1]
        df[column] = new_df[column]

    return df
```

```
42
43
44 def gradient_boost_classifier(x_train, x_test, y_train, y_test):
45     gbc = GradientBoostingClassifier()
46     gbc.fit(x_train, y_train)
47     y_pred_train = gbc.predict(x_train)
48     y_pred = gbc.predict(x_test)
49
50     print("Gradient Boosting Classifier Accuracy:", metrics.
    accuracy_score(y_test, y_pred))
51     print("Gradient Boosting Classifier Accuracy train:", metrics.
    accuracy_score(y_train, y_pred_train))
52     print("Gradient Boosting Classifier f1:", metrics.f1_score(y_test,
    y_pred, average=AVERAGE))
53     print("Gradient Boosting Classifier f1 train:", metrics.f1_score(
    y_train, y_pred_train, average=AVERAGE))
54     print("Gradient Boosting Classifier precision:", metrics.
    precision_score(y_test, y_pred, average=AVERAGE))
55     print("Gradient Boosting Classifier precision train:",
56           metrics.precision_score(y_train, y_pred_train, average=AVERAGE
    ))
57
58     if BOOST_PARAMETERS:
59         grid = GridSearchCV(estimator=gbc,
60                             param_grid=GRADIENT_CLASSIFIER,
61                             scoring=BOOST_SCORE, cv=5)
62         result = grid.fit(x_train, y_train)
63         print("Best: {} using {}".format(result.best_score_, result.
    best_params_))
64
65
66 def gradient_boost_classifier_pol(x_train, x_test, y_train, y_test):
67     gbc = GradientBoostingClassifier(learning_rate=0.02, max_depth=5,
    max_features='sqrt', n_estimators=200)
68
69     pol = PolynomialFeatures(degree=2)
70     x_pol_train = pol.fit_transform(x_train)
71     x_pol_test = pol.transform(x_test)
72
73     gbc.fit(x_pol_train, y_train)
74
75     y_pred = gbc.predict(x_pol_test)
76     y_pred_train = gbc.predict(x_pol_train)
77
78     print("Gradient Boosting Classifier Accuracy:", metrics.
    accuracy_score(y_test, y_pred))
79     print("Gradient Boosting Classifier train:", metrics.accuracy_score(
```

```
    y_train, y_pred_train))
80     print("Gradient Boosting Classifier f1:", metrics.f1_score(y_test,
    y_pred, average=AVERAGE))
81     print("Gradient Boosting Classifier f1 train:", metrics.f1_score(
    y_train, y_pred_train, average=AVERAGE))
82     print("Decision Tree precision:", metrics.precision_score(y_test,
    y_pred, average=AVERAGE))
83     print("Decision Tree precision train:", metrics.precision_score(
    y_train, y_pred_train, average=AVERAGE))
84
85     if BOOST_PARAMETERS:
86         grid = GridSearchCV(estimator=gbc,
87                             param_grid=GRADIENT_CLASSIFIER,
88                             scoring=BOOST_SCORE, cv=5)
89         result = grid.fit(x_train, y_train)
90         print("Best: {} using {}".format(result.best_score_, result.
    best_params_))
91
92
93 def decision_tree(x_train, x_test, y_train, y_test):
94     clf = DecisionTreeClassifier()
95
96     clf = clf.fit(x_train, y_train)
97
98     y_pred = clf.predict(x_test)
99     y_pred_train = clf.predict(x_train)
100
101    print("Decision Tree Accuracy:", metrics.accuracy_score(y_test,
    y_pred))
102    print("Decision Tree Accuracy train:", metrics.accuracy_score(
    y_train, y_pred_train))
103
104    print("Decision Tree f1:", metrics.f1_score(y_test, y_pred, average=
    AVERAGE))
105    print("Decision Tree f1 train:", metrics.f1_score(y_train,
    y_pred_train, average=AVERAGE))
106
107    print("Decision Tree precision:", metrics.precision_score(y_test,
    y_pred, average=AVERAGE))
108    print("Decision Tree precision train:", metrics.precision_score(
    y_train, y_pred_train, average=AVERAGE))
109
110    fig = plt.figure(figsize=(15, 10))
111    _ = tree.plot_tree(clf,
112                       feature_names=x_train.columns,
113                       class_names=CLASS_NAMES,
114                       filled=True,
```

```
115                         max_depth=2, fontsize=18)
116     plt.tight_layout()
117     plt.show()
118     fig.savefig("./decistion_tree.png")
119     if BOOST_PARAMETERS:
120         grid = GridSearchCV(estimator=clf,
121                             param_grid=DECISION_TREE_PARAMETERS,
122                             scoring=BOOST_SCORE, cv=5)
123         result = grid.fit(x_train, y_train)
124 
125         print("DecisionTree {} {}".format(result.best_score_, result.
    best_params_))
126 
127 
128 def decision_tree_pol(x_train, x_test, y_train, y_test):
129     clf = DecisionTreeClassifier(ccp_alpha=0.002, max_features='sqrt')
130 
131     pol = PolynomialFeatures(degree=2)
132     x_pol_train = pol.fit_transform(x_train)
133     x_pol_test = pol.transform(x_test)
134     print(x_train.shape)
135     print(x_pol_train.shape)
136 
137     clf = clf.fit(x_pol_train, y_train)
138 
139     y_pred = clf.predict(x_pol_test)
140     y_pred_train = clf.predict(x_pol_train)
141 
142     print("Decision Tree Accuracy:", metrics.accuracy_score(y_test,
    y_pred))
143     print("Decision Tree Accuracy train:", metrics.accuracy_score(
    y_train, y_pred_train))
144 
145     print("Decision Tree f1:", metrics.f1_score(y_test, y_pred, average=
    AVERAGE))
146     print("Decision Tree f1 train:", metrics.f1_score(y_train,
    y_pred_train, average=AVERAGE))
147 
148     print("Decision Tree precision:", metrics.precision_score(y_test,
    y_pred, average=AVERAGE))
149     print("Decision Tree precision train:", metrics.precision_score(
    y_train, y_pred_train, average=AVERAGE))
150 
151     if BOOST_PARAMETERS:
152         grid = GridSearchCV(estimator=clf,
153                             param_grid=DECISION_TREE_PARAMETERS,
154                             scoring=BOOST_SCORE, cv=5)
```

```
155         result = grid.fit(x_pol_train, y_train)
156 
157         print("DecisionTree pol {} {}".format(result.best_score_, result
    .best_params_))
158 
159 
160 def correlation_heatmap(df):
161     pearsoncorr = df.corr(method='pearson', min_periods=1)
162 
163     sb.heatmap(pearsoncorr,
164                xticklabels=pearsoncorr.columns,
165                yticklabels=pearsoncorr.columns,
166                cmap='RdBu_r',
167                annot=True,
168                linewidth=0.5).plot()
169     plt.tight_layout()
170     plt.show()
171 
172 
173 def logistic_regression(x_train, x_test, y_train, y_test):
174     lr = LogisticRegression()
175 
176     lr.fit(x_train, y_train)
177     y_pred_train = lr.predict(x_train)
178     y_pred = lr.predict(x_test)
179     print("Logistic Regression Accuracy:", metrics.accuracy_score(y_test
    , y_pred))
180     print("Logistic Regression Accuracy train:", metrics.accuracy_score(
    y_train, y_pred_train))
181 
182     print("Logistic Regression f1:", metrics.f1_score(y_test, y_pred,
    average=AVERAGE))
183     print("Logistic Regression f1 train:", metrics.f1_score(y_train,
    y_pred_train, average=AVERAGE))
184 
185     print("Logistic Regression precision:", metrics.precision_score(
    y_test, y_pred, average=AVERAGE))
186     print("Logistic Regression precision train:", metrics.
    precision_score(y_train, y_pred_train, average=AVERAGE))
187 
188     if BOOST_PARAMETERS:
189         grid = GridSearchCV(estimator=lr,
190                             param_grid=LOGISTIC_REGRESSION_PARAMETERS,
191                             scoring=BOOST_SCORE, cv=5)
192         grid.fit(x_train, y_train)
193         result = grid.fit(x_train, y_train)
194         print("Logistic Regression {} {}".format(result.best_score_,
```

```
    result.best_params_))
195
196
197 def logistic_regression_pol(x_train, x_test, y_train, y_test):
198     lr = LogisticRegression(C=2.0, max_iter=600, penalty='l1', solver='
    liblinear', tol=0.01)
199
200     pol = PolynomialFeatures(degree=2)
201     x_pol_train = pol.fit_transform(x_train)
202     x_pol_test = pol.transform(x_test)
203
204     lr.fit(x_pol_train, y_train)
205     y_pred_train = lr.predict(x_pol_train)
206     y_pred = lr.predict(x_pol_test)
207     print("Logistic Regression Accuracy:", metrics.accuracy_score(y_test
    , y_pred))
208     print("Logistic Regression Accuracy train:", metrics.accuracy_score(
    y_train, y_pred_train))
209
210     print("Logistic Regression f1:", metrics.f1_score(y_test, y_pred,
    average=AVERAGE))
211     print("Logistic Regression f1 train:", metrics.f1_score(y_train,
    y_pred_train, average=AVERAGE))
212
213     print("Logistic Regression precision:", metrics.precision_score(
    y_test, y_pred, average=AVERAGE))
214     print("Logistic Regression precision train:", metrics.
    precision_score(y_train, y_pred_train, average=AVERAGE))
215
216     if BOOST_PARAMETERS:
217         grid = GridSearchCV(estimator=lr,
218                             param_grid=LOGISTIC_REGRESSION_PARAMETERS,
219                             scoring=BOOST_SCORE, cv=5)
220         grid.fit(x_train, y_train)
221         result = grid.fit(x_train, y_train)
222         print("Logistic Regression pol {} {}".format(result.best_score_,
     result.best_params_))
223
224
225 def balance(df):
226     # Contagem de registros de THB antes do balanceamento
227     value_counts = dict(df[CLASS_TO_PREDICT].value_counts())
228     print(value_counts)
229     pd.DataFrame({CLASS_TO_PREDICT: ['SIM', 'N O'], 'ocorr ncias': [(
    value_counts[1]), (value_counts[0])]}).plot(
230         kind='bar', x=CLASS_TO_PREDICT, y='ocorr ncias', ec='black')
231
```

```
232
233 def class_occurrences_by_numeric_column_name(df, column_name):
234     d = class_by_occurrence(df, column_name)
235     variations = dict(df[column_name].value_counts())
236     d[column_name] = variations.keys()
237     sb.kdeplot(data=df, x=column_name, hue='THB', fill=True)
238     plt.show()
239
240
241 def class_occurrences_by_categorical_column_name(df, column_name):
242     d = class_by_occurrence(df, column_name)
243     counts = dict(df[column_name].value_counts())
244     width = 0.5
245     fig, ax = plt.subplots()
246     ax.xaxis.set_major_locator(MaxNLocator(integer=True))
247     bottom = np.zeros(len(counts))
248     rows_total_count = []
249
250     for row in range(0, len(d)):
251         rows_total_count.append(sum(d.iloc[row]))
252
253     for column in d.columns:
254         result = [round((item / total_in_row) * 100) if total_in_row > 0
    or round(
255             (item / total_in_row) * 100) != 0 else np.NAN for item,
    total_in_row in zip(d[column], rows_total_count)]
256         p = ax.bar(counts.keys(), result, width, label=column, bottom=
    bottom)
257         bottom += result
258         ax.bar_label(p, label_type='center')
259     ax.set_title(column_name)
260     ax.legend()
261     plt.show()
262
263
264 def class_by_occurrence(df, column_name):
265     counts = dict(df[column_name].value_counts())
266     counts_class = dict(df[CLASS_TO_PREDICT].value_counts())
267     print(counts)
268     print(counts_class)
269     column_names = []
270     d = pd.DataFrame()
271     for key in counts_class:
272         column_names.append(CLASS_TO_PREDICT + key)
273     for key in counts_class:
274         key_list = []
275         for key_occ in counts:
```

```
276             key_list.append(len(df[(df[column_name] == key_occ) & (df[
    CLASS_TO_PREDICT] == key)]))
277         d[CLASS_TO_PREDICT + key] = key_list
278     return d
279
280
281 def feature_selection(x_train, y_train, columns):
282     rf = RandomForestRegressor(random_state=1)
283     rf.fit(x_train, y_train)
284
285     dict_importances = list(zip(columns, rf.feature_importances_))
286     dict_importances.sort(key=lambda x: x[1])
287     plt.barh([x[0] for x in dict_importances], [x[1] for x in
    dict_importances])
288
289     rfe = RFECV(rf, cv=5, scoring="neg_mean_squared_error")
290
291     rfe.fit(x_train, y_train)
292
293     selected_features = np.array(columns)[rfe.get_support()]
294     print(selected_features)
295
296     plt.show()
297
298
299 def main():
300     warnings.simplefilter("ignore")
301
302     df = read_database()
303     df = df.drop(columns=COLUMNS_TO_DROP)
304     print(df.columns)
305     for data in CATEGORICAL_DATA:
306         class_occurrences_by_categorical_column_name(df, data)
307
308     for data in NUMERICAL_DATA:
309         class_occurrences_by_numeric_column_name(df, data)
310
311     imputation_numeric_data(df)
312     imputation_categorical_data(df)
313     df = df.dropna()
314
315     df_one_hot_encoded = encode_data(df)
316
317     if NORMALIZE:
318         scaler = MinMaxScaler()
319         df_one_hot_encoded[COLUMNS_TO_NORMALIZE] = scaler.fit_transform(
    df_one_hot_encoded[COLUMNS_TO_NORMALIZE])
```

```
320     correlation_heatmap(df_one_hot_encoded[HEATMAP])
321
322     if WITH_KNN_IMPUTER:
323         imputer = KNNImputer(n_neighbors=5)
324         df_one_hot_encoded = pd.DataFrame(imputer.fit_transform(
    df_one_hot_encoded), columns=df_one_hot_encoded.columns)
325
326     df_without_nan = df_one_hot_encoded.dropna()
327
328     y = df_without_nan[CLASS_TO_PREDICT]
329     df_without_thb = df_without_nan.drop(columns=[CLASS_TO_PREDICT])
330
331     feature_columns = df_without_thb.columns
332
333     x_train, x_test, y_train, y_test = train_test_split(df_without_thb,
    y, test_size=0.2, random_state=1)
334
335     if WITH_BALANCE:
336         os = RandomOverSampler()
337         x_train, y_train = os.fit_resample(x_train, y_train)
338
339     feature_selection(x_train, y_train, feature_columns)
340
341     print("---------------------------------------\n")
342     decision_tree(x_train, x_test, y_train, y_test)
343     print("---------------------------------------\n")
344     decision_tree_pol(x_train, x_test, y_train, y_test)
345     print("---------------------------------------\n")
346     logistic_regression(x_train, x_test, y_train, y_test)
347     print("---------------------------------------\n")
348     logistic_regression_pol(x_train, x_test, y_train, y_test)
349     print("---------------------------------------\n")
350     gradient_boost_classifier(x_train, x_test, y_train, y_test)
351     print("---------------------------------------\n")
352     gradient_boost_classifier_pol(x_train, x_test, y_train, y_test)
353
354
355 main()
```

Listing A.1 – main.py

A.2 CÓDIGO COM AS REGRAS DE IMPUTAÇÃO

```
1 import numpy as np
2 from sklearn.impute import SimpleImputer
3 from constants import *
4 import pandas as pd
```

```
5 import matplotlib.pyplot as plt
6 plt.rcParams.update({'font.size': 25})
7
8
9
10
11 def imputation_numeric_data(dataset_principais):
12     numerical_imputer = SimpleImputer(strategy=
    NUMERIC_IMPUTATION_STRATEGY, missing_values=np.nan)
13     missing_data_count = []
14     all_data_count = []
15     if WITH_IMPUTATION:
16         for column in NUMERICAL_DATA:
17             all_data_in_column = len(dataset_principais[column])
18             all_data_count.append(all_data_in_column)
19             missing_data_in_column = dataset_principais[column].isnull()
    .sum()
20             column_len = len(dataset_principais[column])
21             missing_data_count.append(missing_data_in_column)
22             if missing_data_in_column < MISSING_DATA_MAX_PERCENT *
    column_len:
23                 dataset_principais[column] = numerical_imputer.
    fit_transform(dataset_principais[column].values.reshape(-1, 1))[:, 0]
24
25     d = pd.DataFrame({"Atributos Num ricos": NUMERICAL_DATA, "dados
    faltantes": missing_data_count}).plot(kind='bar', x="Atributos
    Num ricos", y='dados faltantes')
26     d.set_ylabel("Quantidade")
27     pd.DataFrame({"coluna": NUMERICAL_DATA, "n mero de registros":
    all_data_count}).plot(kind='bar', x="coluna", y='n mero de registros
    ')
28
29     plt.xticks(range(len(NUMERICAL_DATA)), NUMERICAL_DATA, rotation=20,
    )
30     plt.show()
31
32
33 def imputation_categorical_data(dataset_principais):
34     categorical_imputer = SimpleImputer(strategy=
    CATEGORICAL_IMPUTATION_STRATEGY, missing_values=np.nan)
35     missing_data_count = []
36     all_data_count = []
37     if WITH_IMPUTATION:
38         for column in CATEGORICAL_DATA:
39             all_data_in_column = len(dataset_principais[column])
40             all_data_count.append(all_data_in_column)
41             missing_data_in_column = dataset_principais[column].isnull()
```

```
    .sum()
42              column_len = len(dataset_principais[column])
43              missing_data_count.append(missing_data_in_column)
44              if missing_data_in_column < MISSING_DATA_MAX_PERCENT *
    column_len:
45                  dataset_principais[column] = categorical_imputer.
    fit_transform(dataset_principais[column].values.reshape(-1, 1))[:, 0]
46
47          d =pd.DataFrame({"Atributos Categ ricos": CATEGORICAL_DATA, "
    dados faltantes": missing_data_count}).plot(kind='bar', x="Atributos
    Categ ricos", y='dados faltantes')
48          d.set_ylabel("Quantidade")
49
50          pd.DataFrame({"coluna": CATEGORICAL_DATA, "n mero de registros"
    : all_data_count}).plot(kind='bar', x="coluna", y='n mero de
    registros')
51
52          plt.xticks(range(len(CATEGORICAL_DATA)), CATEGORICAL_DATA,
    rotation=20)
53          plt.tight_layout()
54          plt.show()
```

Listing A.2 – imputation.py

A.3 CÓDIGO COM AS CONSTANTES UTILIZADAS

```
1 CLASS_TO_PREDICT = 'THB'
2
3 COLUMNS_TO_NORMALIZE= ['Raiva-Intensidade', 'Sensibilidade-Intensidade',
      'jovem-adulto', 'Idade','Cautela', 'Vontade',
4                      'Raiva', 'Medo', 'Inibi  o', 'Sensibilidade', '
    Coping', 'Controle', 'Etnia', 'A4MA5M', 'Sexo',
5                      'SNPLEPR',  'A4M', 'A5M', 'I4E',  '
    episodios_abuso', 'E6','SOMA_DROGASABUSO', 'G1']
6
7 CLASS_NAMES = ['N  O', 'SIM']
8
9 CATEGORICAL_OCCURRENCE_TO_PLOT = 'Episodio_depressao'
10
11 NUMERICAL_OCCURRENCE_TO_PLOT = 'Raiva'
12
13 HEATMAP = ['Cautela','Idade', 'Vontade', 'Raiva', 'Medo',  'Inibi  o',
      'Sensibilidade', 'Coping', 'Controle', 'THB']
14
15 DATASET_PATH = '/Users/mguidolin/Downloads/principais-6.csv'
16
```

```
17 CATEGORICAL_DATA = ['Etnia', 'A4MA5M','Sexo','SNPLEPR', 'A4M', 'A5M', '
   I4E', 'episodios_abuso', 'E6','SOMA_DROGASABUSO',
18                    'G1','Raiva-Intensidade', 'Sensibilidade-Intensidade
   ', 'jovem-adulto']
19
20 NUMERICAL_DATA = ['Cautela','Idade', 'Vontade', 'Raiva', 'Medo', '
   Inibi o', 'Sensibilidade', 'Coping', 'Controle']
21
22 TO_ENCODE = ['Raiva-Intensidade', 'Sensibilidade-Intensidade', 'jovem-
   adulto', 'Sexo', 'Etnia', 'SNPLEPR', 'A4M', 'A5M',
23              'A4MA5M', 'I4E', 'SOMA_DROGASABUSO', 'episodios_abuso', 'G1
   ', 'E6', 'THB']
24
25 COLUMNS_TO_DROP = ['QUEST', 'Episodio_depressao', 'H1']
26
27 NUMERIC_IMPUTATION_STRATEGY = 'median'
28
29 CATEGORICAL_IMPUTATION_STRATEGY = 'most_frequent'
30
31 MISSING_DATA_MAX_PERCENT = 0.4
32
33 WITH_IMPUTATION = True
34
35 WITH_KNN_IMPUTER = False
36
37 WITH_BALANCE = False
38
39 BOOST_PARAMETERS = False
40
41 BOOST_SCORE = 'f1'
42
43 AVERAGE = "binary"
44
45 NORMALIZE = True
46
47 LOGISTIC_REGRESSION_PARAMETERS = [
48
49 {
50         "penalty": ['l2'],
51         "tol": [1e-5, 1e-4, 1e-3, 1e-2],
52         "C": [0.0, 0.5, 1.0, 1.5, 2.0],
53         "solver": ['lbfgs', 'newton-cg', 'newton-cholesky', 'sag', 'saga
   ', 'liblinear'],
54         "max_iter": [200, 300, 400, 500, 600, 700],
55     },
56     {
57         "penalty": ['l1'],
```

```
58          "tol": [1e-5, 1e-4, 1e-3, 1e-2],
59          "C": [0.0, 0.5, 1.0, 1.5, 2.0],
60          "solver": ['saga', 'liblinear'],
61          "max_iter": [200, 300, 400, 500, 600, 700],
62      },
63      {
64          "penalty": [None],
65          "tol": [1e-5, 1e-4, 1e-3, 1e-2],
66          "C": [0.0, 0.5, 1.0, 1.5, 2.0],
67          "solver": ['lbfgs', 'newton-cg', 'newton-cholesky', 'sag', 'saga
        '],
68          "max_iter": [200, 300, 400, 500, 600, 700],
69      },
70      {
71          "penalty": ['elasticnet'],
72          "tol": [1e-5, 1e-4, 1e-3, 1e-2],
73          "C": [0.0, 0.5, 1.0, 1.5, 2.0],
74          "class_weight": ['balanced', None],
75          "solver": ['saga'],
76          "max_iter": [200, 300, 400, 500, 600, 700],
77      },
78  ]
79
80  DECISION_TREE_PARAMETERS = [
81      {
82          'ccp_alpha': [0.09, 0.02, 0.03, 0.04, 0.05, 0.01, 0.001],
83          'max_depth': [5, 6, 7, 8, 9, None],
84          'max_features': ['sqrt', 'log2', None]
85      }
86  ]
87
88  GRADIENT_CLASSIFIER = [
89      {
90          "learning_rate": [0.025, 0.05, 0.075, 0.1, 0.15, 0.2],
91          "max_depth":[1, 2, 3, 4, 5],
92          "max_features":["log2", "sqrt", None],
93          "n_estimators":[50, 100, 150, 200]
94      }
95  ]
96
97  def get_values(df, column_from):
98      return [*set(df[column_from].values)]
```

Listing A.3 – constants.py

APÊNDICE B – ARTIGO

Neste apêndice será apresentado o artigo desenvolvido a partir deste trabalho seguindo o padrão da Sociedade Brasileira de Computação (SBC)

Uso de técnicas de Mineração de Dados e Aprendizado de Máquina para identificar casos suspeitos de Transtorno do Humor Bipolar

Marina Pereira das Neves Guidolin[1], **Jonata Tyska**[1], **Mateus Grellert**[2]

[1]Departamento de Informatica e Estatística – Universidade Federal de Santa Catarina (UFSC) – Florianopolis – SC – Brasil

[2]Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS) Caixa Postal 15.064 – 91.501-970 – Porto Alegre – RS – Brazil

Abstract. *Bipolar Disorder (BD) is a condition in which the patient has mood swings and sudden behavioral changes. The diagnosis is extremely important, as it allows patients who suffer from this disorder to seek adequate treatment for a better quality of life. The diagnosis is made by a professional based on the analysis of the patient's profile. However, this process involves analyzing several symptoms and biological markers related to the disorder. Diagnosing a patient with BD depends on a thorough analysis of interactions between different characteristics, which makes this process quite complex even for specialists. The literature points to the use of machine learning (ML) tools and data mining (DM) to aid in the diagnosis of diseases and disorders, achieving satisfactory results for numerous applications. The objective of this work is to develop a solution for identifying THB and its main determinants using learning techniques machine learning and data mining to contribute to mental health professionals.*

Resumo. *O Transtorno do Humor Bipolar (THB) é uma condição na qual o paciente apresenta oscilações de humor e alterações comportamentais repentinas. O diagnóstico é de extrem importância, pois possibilita que pacientes que sofrem com este transtorno possam buscar o tratamento adequado para uma qualidade de vida maior. O diagnóstico é feito por um profissional a partir da análise do perfil do paciente. Todavia, este processo envolve analisar diversos sintomas e marcadores biológicos relacionados ao transtorno. Diagnosticar um paciente com THB depende de uma análise minuciosa de interações entre diversas características, o que torna este processo bastante complexo mesmo para especialistas. A literatura aponta o uso de ferramentas de aprendizado de máquina (Machine Learning - ML) e mineração de dados (Data Mining - DM) para auxiliar no diagnóstico de doenças e transtornos, atingindo resultados satisfatórios para inúmeras aplicações. O objetivo do presente trabalho é desenvolver uma solução para identificação de THB e seus principais determinantes utilizando técnicas de aprendizado de máquina e mineração de dados para contribuir com profissionais de saúde mental.*

1. Introdução

A Inteligência Artificial (IA) é uma área da computação definida como o estudo de agentes que recebem percepções do ambiente e tomam decisões ou realizam ações. Para isso,

cada um desses agentes implementa uma função que mapeia sequências de percepção para estas decisões [RUSSEL, Stuart J. 2010].

A IA tem se desenvolvido muito nos últimos anos, em parte devido ao crescimento de outra área de pesquisa conhecida como Mineração de Dados (do inglês *Data Mining* - DM). Esta área estuda técnicas e ferramentas para coleta, tratamento, análise e extração de informação presente em dados de diversos tipos (textos, valores formatados, imagens etc) [Aggarwal et al. 2015].

As técnicas de DM são particularmente eficientes quando combinadas com algoritmos de Aprendizado de Máquina (do inglês *Machine Learning* - ML). De acordo com [MITCHELL 1997], soluções baseadas em ML aprendem através de experiência (ou dados) a desenvolver certas tarefas, com base em uma medida de desempenho, se esse desempenho melhora a partir da experiência. Essas duas áreas de estudo são tão próximas que por vezes são utilizadas como sinônimos.

A combinação entre DM e ML possibilita que sejam construídos diversos tipos de modelos de aprendizado. Dentre eles, destacam-se os modelos preditivos, os quais conseguem definir tendências a partir dos padrões identificados e prever o valor de uma variável alvo. Tais modelos possuem diversas aplicações, sobretudo na área da saúde, na qual auxiliam os profissionais no diagnóstico de doenças como COVID-19 [Zoabi et al. 2021], assim como doenças e transtornos mentais [Azar et al. 2015]. Dentre os transtornos mentais, podemos destacar o Transtorno do Humor Bipolar (THB), uma condição que afeta de 1% a 2% da população adulta [Fears and Reus 2015]. O diagnóstico do THB ainda é feito de forma clínica e é tópico intensa de pesquisa, pois ele envolve muitas variáveis e inúmeras interações entre elas. Com isso, alguns trabalhos publicados na literatura já propõem o uso de ML para auxiliar no diagnóstico de THB [Perez Arribas et al. 2018, Jadhav et al. 2019].

O presente artigo busca trazer contribuições nesta direção, propondo uma solução que combina mineração de dados e aprendizado de máquina para auxiliar no diagnóstico do Transtorno do Humor Bipolar.

2. Conceitos básicos e trabalhos relacionados

2.1. Conceitos básicos

2.1.1. Transtorno do Humor Bipolar

O Transtorno do Humor Bipolar - THB é um transtorno de humor que afeta de 1% a 2% da população adulta [Fears and Reus 2015]. De acordo com [American Psychiatric Association et al. 2013], existem três tipo de manifestações do THB: Transtorno Bipolar I: transtorno maníaco-depressivo que pode existir com e sem episódios psicóticos; Transtorno Bipolar II: episódios depressivos pelo menos um hipomaníacos durante a vida; Transtorno Ciclotímico: é um transtorno cíclico que causa breves episódios de hipomania e depressão.

Dos pacientes que apresentam o THB, entre 10-20% tiram a própria vida e aproximadamente um terço admite ao menos uma tentativa de suicídio [Müller-Oerlinghausen et al. 2002]. As manifestações clínicas desse transtorno são muito diversas, e incluem muitas variações de humor, fazendo com que seu diagnóstico seja

bastante complexo. Manifestações clínicas leves, por exemplo, são difíceis de distinguir de oscilações normais de humor e características da personalidade, principalmente em fases iniciais do transtorno. Porém, mesmo quando os sintomas apresentados são típicos do transtorno bipolar, ainda podem ser diagnosticados erroneamente como ansiedade comórbida, por exemplo. Tais fatores tornam o diagnóstico de transtorno bipolar muito difícil, aumentando sua complexidade para indivíduos que o apresentam já na adolescência, quando o cérebro ainda está em fase de desenvolvimento. [Grande et al. 2016]

2.1.2. Aprendizado de Máquina

Segundo [MITCHELL 1997], um processo de aprendizado se dá a partir da experiência adquirida a respeito de uma classe de tarefas, fazendo que, com o decorrer do tempo, sua performance de execução aumente conforme mais experiências são adquiridas. Quando esse processo de aprendizado é estendido para um algoritmo, tem-se o conceito de Aprendizado de Máquina (do inglês *Machine Learning* - ML).

Com isso, pode-se levantar o questionamento sobre como devem ser construídos programas de computador que possuam essa capacidade de aprender. Essa construção deve levar em consideração que uma abordagem de aprendizado de máquina contempla decisões envolvendo padrões para modelagem do projeto, treinamento para chegar ao aprendizado esperado, definição de qual será a tarefa a ser aprendida, busca de um exemplo sobre como é essa tarefa e um algoritmo para que a partir do exemplo da tarefa seja possível realizar o processo de aprendizado juntamente aos treinamentos escolhidos. [MITCHELL 1997]

As técnicas de aprendizado de máquina podem ser categorizadas de acordo com três tipos [Stuart Russell 1995], o primeiro deles é o **aprendizado não supervisionado**, no qual ao aprender a executar o exemplo de uma tarefa não se sabe qual o resultado esperado para ela. Já o segundo chama-se **aprendizado por reforço**, quando existe a necessidade de avaliar o contexto de uma tarefa e gerar um reforço positivo ou negativo para que o agente executando tome uma decisão. Por último, a técnica de **aprendizado supervisionado** diz respeito à situações em que os exemplos de tarefas a serem realizadas também já possuem os resultados esperados.

2.1.3. Árvore de Decisão

O aprendizado de máquina a partir de árvores de decisão é um dos métodos mais utilizados e práticos para inferência indutiva [MITCHELL 1997], no qual os dados são analisados com o objetivo de reconhecimento de padrões. Seu aprendizado se dá a partir do uso de dados de entrada, com os quais são feitos testes lógicos para verificar os padrões existentes e construir um caminho a partir dos padrões considerados como verdade até chegar em uma tomada de decisão [Stuart Russell 1995]. Essa sequência de passos é construída de maneira hierárquica pela importância de cada atributo, sendo que para cada dado de entrada é feita uma validação com seus possíveis valores com o objetivo de encontrar uma comparação verdadeira. Até encontrar comparações verdadeiras a árvore de decisão vai criando sua estrutura a partir de nós internos que representam as condições que foram avaliadas. Quando uma comparação é verdadeira, é criado um 'nó folha', que representa

o desfecho.

A Fig.1 representa a estrutura de decisão montada por uma árvore. Dentro de cada um dos nós é possível identificar o nome do atributo ao qual ele se refere, seguido pelo *gini*, que mede a qualidade da bifurcação para aquele nó a partir do cálculo de impureza ou incerteza dos dados. Dessa maneira, o passo a passo para uma tomada de decisão se dá da seguinte forma:

- Identifica o atributo de maior relevância observando o valor de *gini*;
- Bifurca o atributo criando dois novos nós;
- Repete o processo até encontrar um nó folha ou atingir a profundidade máxima.

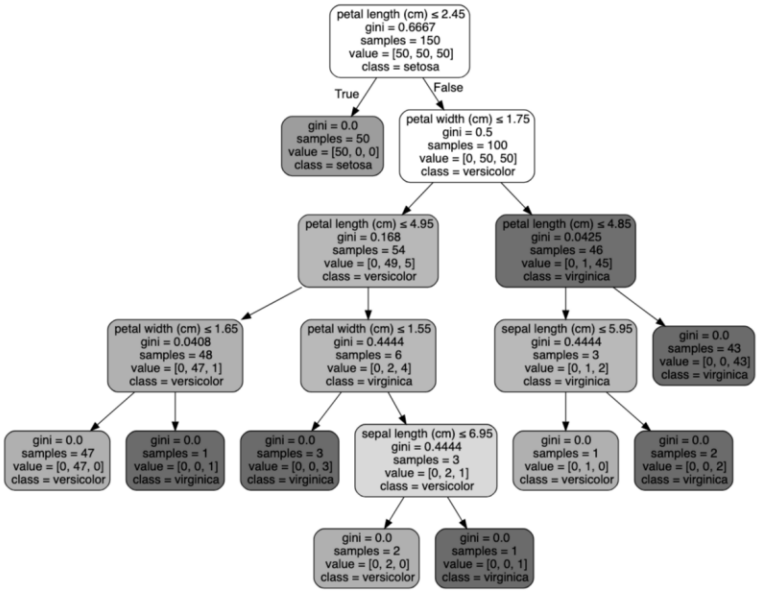

Figure 1. Exemplo de árvore de decisão
ScikitLearn

2.1.4. Regressão Logística

O modelo de Regressão Logística é uma ferramenta de aprendizado de máquina que é implementada a partir de um modelo linear de regressão. Ele surge a partir da necessidade de modelar a posteriori probabilidades de K classes via funções lineares em x, enquanto ao mesmo tempo garantem que essas probabilidades somam um e permanecem no intervalo [0, 1] [Hastie et al. 2001]. Dessa forma, ao receber os dados de entrada o algoritmo as combina de maneira linear para prever o desfecho. Ele pode ser utilizado com três propósitos:

- Prever a probabilidade de o desfecho ser igual a 1;
- Categorizar o desfecho ou as predições;
- Verificar as probabilidades ou riscos associados aos preditores do modelo

Ele é bastante utilizado para quando o desfecho tem valores binários (0 ou 1) e segue uma distribuição de probabilidade de Bernoulli. Como a distribuição de Bernoulli é um subconjunto da distribuição binomial mais geral, a regressão logística é reconhecida como um membro da família binomial de modelos de regressão [Hilbe 2015].

2.1.5. XGBoost

Outro modelo de aprendizado de máquina é o XGBoost, também conhecido como *Gradient Boosting Machine* e implementado pelo *ScikitLearn* [Pedregosa et al. 2011]. Ele é um modelo robusto que implementa técnicas que objetivam fornecer alta performance e escalabilidade, que é o principal fator responsável pelo sucesso deste modelo. Essa escalabilidade se deve aos vários sistemas e otimizações algorítmicas utilizadas por ele, que incluem otimizações ao utilizar a árvore de decisão, criando um novo algoritmo de aprendizado para lidar com dados esparsos e a inclusão de um procedimento de aprendizado ponderado, permitindo lidar com pesos ao utilizar árvores de decisão. Além disso, também implementa técnicas de regularização L1 e L2, podas de árvore, que consiste na remoção dos nós com o objetivo de diminuir a complexidade das árvores utilizadas e reduzir o *overfitting*, entre outras. [Chen and Guestrin 2016]

2.1.6. Medidores de Performance: F1 e Acurácia

Para que seja possível medir o desempenho dos modelos e quantificar a qualidade das predições é necessário que existam métricas padrão para avaliar todos de maneira igual e poder comparar resultados. Com este fim, duas métricas serão as principais avaliadas neste trabalho: a acurácia e o f1.

A acurácia é a métrica que diz respeito à performance dos modelos, avaliando o quão corretas são as predições feitas por ele e medindo a proporção de erros e acertos de acordo com o numero total de entradas.

A Equação 1 indica como é feito o cálculo da acurácia para classificação binária, sendo *tp* os valores que foram previstos como positivos e realmente eram positivos (verdadeiros positivos), *tn* os valores previstos como negativos e eram negativos (verdadeiros negativos), *fp* e *fn* representando falsos positivos e falsos negativos, respectivamente.

$$acurácia = \frac{tp + tn}{tp + fp + tn + fn} \tag{1}$$

O F1 é uma métrica que depende de outras duas: precisão e *recall* [Sokolova and Lapalme 2009]. A precisão define a proporção de positivos verdadeiros dentre todos os verdadeiros indicados pelo modelo, quantificando o quão preciso o modelo foi em prever corretamente o desfecho, representado pela Equação 2. Já o *recall* representa a taxa de positivos verdadeiros, medindo o quão bom o modelo é em identificar os valores positivos do desfecho, cuja fórmula está representada na Equação 3.

$$precísao = \frac{tp}{tp + fp} \tag{2}$$

$$recall = \frac{tp}{tp + fn} \tag{3}$$

A partir do cálculo de precisão e *recall*, podemos calcular a métrica F1 com base em ambas as métricas de acordo a Equação 4:

$$F1 = \frac{2 * (precísao * recall)}{precísao + recall} \quad (4)$$

2.2. Trabalhos relacionados

A signature-based machine learning model for distinguishing bipolar disorder and borderline personality disorder [?]

Esse trabalho analizado utiliza dados provenientes de questionários com dados de pacientes saudáveis, com transtorno do humor bipolar ou transtorno de personalidade borderline. Ele tem por objetivo classificá-los de acordo com seu transtorno e prever seus humores subsequentes. Para isso, foram utilizados dados vindos de um questionário no qual os participantes classificavam diariamente seu humor de acordo com seis diferentes categorias: ansiedade, euforia, tristeza, raiva, irritabilidade e energia. Tanto para prever o diagnóstico do paciente quanto seu humor, o trabalho aplicou regressão utilizando random forest e os resultados apresentados mostram que 75% dos participantes foram categorizados de acordo com seu diagnóstico original e na previsão do humor dos pacientes foram obtidos 89-98% de acurácia em pacientes saudáveis, 82-90% de acurácia para pacientes com transtorno do humor bipolar e 70-78% para pacientes com transtorno de personalidade borderline

Mental disorder detection: Bipolar disorder scrutinization using machine learning [Jadhav et al. 2019]

Analisando o trabalho, foi identificado seu objetivo, que é identificar pacientes que possuem o THB para auxiliar em seus diagnósticos. Nele é ressaltada a importância do diagnóstico do transtorno do humor bipolar, visto que esse transtorno persiste ao longo da vida de 4-5% da população geral. Para atingir o objetivo, o trabalho utiliza um questionário chamado *Mood Disorder Questionnaire* (MDQ), que é aplicado como uma abordagem inicial para o diagnóstico da presença de sintomas do THB e composto por perguntas abordando situações não comuns, que geralmente não ocorrem com pessoas consideradas saudáveis. Cada resposta pode ser preenchida com *"SIM"* ou *"NÃO"* e, de acordo com os resultados, outras perguntas são ou não aplicadas ao paciente. O trabalho também menciona pesquisas demonstrando uma uma taxa de 70% de acerto quando aplicado o MDQ para o diagnóstico do transtorno. Para a construção do modelo, foi implementada uma estrutura de árvore de decisão utilizando a biblioteca *SKlearn* da linguagem Python, que internamente utiliza o algoritmo CART. Com isso, um *dataset* possuindo dados coletados de 983 testes de triagem aplicados, sendo 864 negativos para THB e 119 positivos, foi utilizado e dividido em duas partes: 60% dos dados foram usado para o treinamento do modelo e 40% para testá-lo. Os resultados apresentaram uma acurácia de 88.07% e identificou perda de acurácia nas situações em que menos de sete perguntas do MDQ eram respondidas.

BD-MDL: BIPOLAR DISORDER DETECTION USING MACHINE LEANRING AND DEEP LEARNING [R.Saranya 2022]

No trabalho são mencionadas as fases do transtorno do humor bipolar ressaltando manifestações de episódios de mania e depressão. O conjunto de dados utilizado contém dados relativos a idade, gênero e sentimentos que os pacientes possuem, como impaciência e melancolia, por exemplo. Após isso, o autor discorre sobre a importância

do pré processamento dos dados visto que os valores faltantes ou outros fatores podem inutilizar o *dataset*. Além disso, também é mencionado o aumento na qualidade e eficácia das descobertas por conta do pré processamento. Para realizá-lo foram necessárias duas etapas, sendo a primeira a normalização dos dados para que informações discrepantes fossem removidas e a segunda a utilização de *DecisionTreeRegressor* com *Standard scalar*. Após o pré processamento foi feita a seleção de features utilizando algoritmo de floresta aleatória e regressão logística. O modelo utilizado foi o de redes neurais convolucionais com memória de curto e longo prazo, que foi aprimorado com a otimização de Adam (Estimativa de Momento Adaptativo). Por fim, o modelo utilizado no artigo atingiu uma acurácia de 97.2%.

Learning the mental health impact of covid-19 in the united states with explainable artificial intelligence: Observational study [Jha et al. 2021]

O trabalho possui objetivo de identificar um conjunto de fatores que possam indicar uma predisposição ao desenvolvimento de um distúrbio mental durante a pandemia da COVID-19. Para prever esses fatores, foram utilizados dados de 17764 adultos coletados a partir de uma pesquisa sobre o impacto da COVID, que fornece dados dados socioeconômicos tais como cidade, estado, se ficou sem comida no período de entre os meses maio, abril e junho de 2020, se está empregado, idade, sexo, informação se o indivíduo se comunica com os amigos, entre outros dados que resultam de informações com cinco opções para o indivíduo selecionar a qual mais se identifica ou perguntas diretas com respostas de *"SIM"* ou *"NÃO"*. A partir de uma análise estatística inicial seguida pelo uso de redes bayesianas com abordagens clássicas de aprendizado de máquina para a construção do um modelo preditivo, foram identificados os principais fatores que afetam a saúde mental durante a pandemia da COVID e também prever quais participantes cujos indicadores de saúde mental os tornavam mais vulneráveis que os demais com uma acurácia de aproximadamente 80%. Também foi utilizado o teste de independência do qui-quadrado para verificar associações entre indicadores de saúde mental e demais variáveis.

No trabalho são mencionadas as fases do transtorno do humor bipolar ressaltando manifestações de episódios de mania e depressão. O conjunto de dados utilizado contém dados relativos a idade, gênero e sentimentos que os pacientes possuem, como impaciência e melancolia, por exemplo. Após isso, o autor discorre sobre a importância do pré processamento dos dados visto que os valores faltantes ou outros fatores podem inutilizar o *dataset*. Além disso, também é mencionado o aumento na qualidade e eficácia das descobertas por conta do pré processamento. Para realizá-lo foram necessárias duas etapas, sendo a primeira a normalização dos dados para que informações discrepantes fossem removidas e a segunda a utilização de *DecisionTreeRegressor* com *Standard scalar*. Após o pré processamento foi feita a seleção de features utilizando algoritmo de floresta aleatória e regressão logística. O modelo utilizado foi o de redes neurais convolucionais com memória de curto e longo prazo, que foi aprimorado com a otimização de Adam (Estimativa de Momento Adaptativo). Por fim, o modelo utilizado no artigo atingiu uma acurácia de 97.2%.

Behavioral modeling for mental health using machine learning algorithms [Srividya et al. 2018]

O objetivo deste trabalho é a criação de uma ferramenta para determinar o estado mental de um indivíduo e auxiliar profissionais na realização de testes e diagnóstico de transtornos mentais de seus pacientes e menciona a importância do diagnóstico precoce de transtornos mentais para que seja possível manter um equilíbrio de vida adequado. Além disso, também é ressaltado que, com o avanço da tecnologia, o papel dos profissionais da saúde pode ser suplementado por modelos que fazem o uso de inteligência artificial. Nesse trabalho, foi usado um questionário contendo 20 questões e um *dataset* com 656 indivíduos e suas respostas, que foi dividido em 80% para o conjunto de treinamento e 20% para o conjunto de testes. Para a construção do modelo, foram usados diferentes algoritmos de classificação: Regressão logística, *Naïve bayes*, Máquinas vetoriais de suporte, Árvore de decisão e K-vizinhos mais próximos. Além esses algoritmos, também foram construídos modelos com *Ensmble bagging* e *Tree ensemble* utilizando *random forest*. As acurácias dos diferentes algoritmos classificadores para prever o estado mental dos pacientes foram: 84% para Regressão logística; 73% para *Naïve bayes*; 89% para SVM; 81% para árvore de decisão; 89% para o K-vizinhos mais próximos; 90% para *Ensemble bagging* e 90% para *Tree Ensemble* utilizando *Random Forest*. Por fim, o trabalho concluiu que considerando os resultados obtidos, é possível utilizar seus modelos como mecanismos auxiliadores para realizar a modelagem comportamental de um grupo de indivíduos.

2.2.1. Principais achados

A partir dos trabalhos apresentados, percebe-se que o diagnóstico do transtorno do humor bipolar depende de muitos fatores comportamentais e que com o uso de questionários é possível identificá-los, evidenciando a importância da inteligência artificial como ferramenta de apoio. Em relação aos resultados, os algoritmos *Ensmble bagging* e *Tree ensemble* com *random forest* obtiveram uma acurácia de 90%, o que demonstra o quão promissoras podem ser suas aplicações. Além destes, os outros algoritmos mencionados devem ser considerados, visto que seus resultados também demonstraram potencial

3. Proposta de solução

Este trabalho é desenvolvido a partir de uma metodologia inspirada no processo conhecido como *CRoss Industry Standard Process for Data Mining* (CRISP-DM) [Wirth and Hipp 2000], que é utilizada para representar em forma de etapas um projeto que utiliza mineração de dados de maneira que ele seja visto como um ciclo. Essas etapas são: compreensão do problema; compreensão dos dados; preparação dos dados; modelagem; avaliação e implantação. A Fig.2 ilustra o fluxo planejado.

O grupo de pesquisadores do Programa de Pós-Graduação em Saúde e Comportamento (PPGSC) da Universidade Católica de Pelotas (UCPel) disponibilizou um conjunto de dados obtido a partir de um estudo de coorte transversal, uma modalidade de estudo observacional em ciências sociais e biológicas que analisa dados de um subconjunto representativo da população em um momento específico [Coggon et al. 1997]. Tais dados contém informações a respeito do perfil de pacientes que passaram por atendimento psicológico, tais como sexo, idade, etnia, episódio de depressão, sentimentos que o paciente possui, marcadores biológicos e se possui ou não o Transtorno do Humor Bipolar

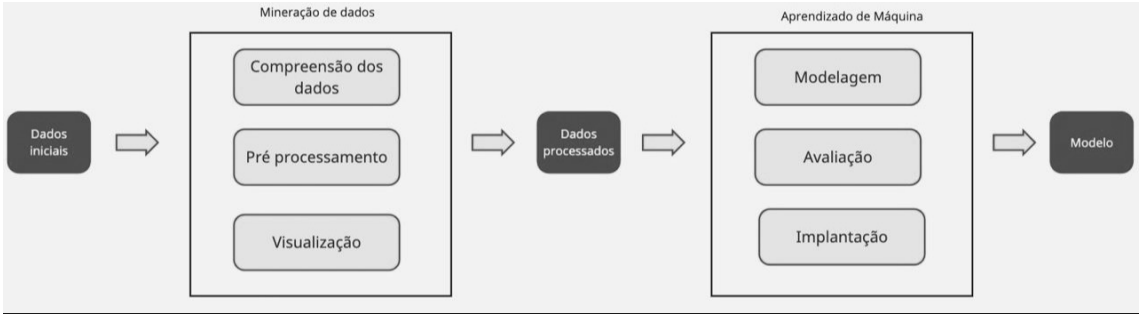

Figure 2. Etapas do projeto inspiradas na metodologia CRISP-DM
Elaborada pelo autor (2022).

(THB). A todo, inicialmente existiam 600 atributos e 808 registros, que passaram por uma seleção de atributos referentes ao humor, transtornos, marcadores biológicos e perfil dos pacientes, resultando em um total de 22 atributos selecionados para o trabaho.

Para a etapa pré processamento foi necessário compreender como os dados fornecidos estão organizados e categorizados, obtendo informações sobre o *dataset* como número de linhas, tabelas e colunas. Tais informações auxiliarão na parte de análise inicial dos dados para que seja feito um pré-processamento, identificando o número de pacientes, quais os estados de humor com os quais mais se identificam, idade, sexo, dados socioeconômicos, entre outras informações que possam agregar ao trabalho.

Além disso, também é necessário preparar os dados para que eles sejam utilizados na construção do modelo. Por ser um conjunto de dados provenientes de questionário, é esperado que existam falhas em seu preenchimento, sendo assim necessário que haja um tratamento para dados faltantes. Dentre as opções para tratar os dados faltantes estão: eliminar os objetos ou atributos faltantes, ignorar os valores faltantes durante a análise ou estimar [Tan et al. 2018]. Ao eliminar ou ignorar tais valores a perda de dados seria considerável, então a imputação dos dados faltantes a partir de uma estimativa se mostrou a melhor opção para evitar perda de registros ou dados. O banco de dados possui tanto dados numéricos quanto categóricos, sendo assim é necessário que o cálculo de estimativas seja diferente para os dois tipos de dados, portanto, será utilizado o cálculo de média para os dados numéricos e moda para os dados categóricos.

Por ser um conjunto de dados clínicos e pelo fato de o Transtorno do Humor bipolar estar presente apenas em 1% a 2% da população adulta [Fears and Reus 2015] é esperado que a quantidade de amostras com a coluna de THB marcada como ”sim” seja diferente e inferior das marcadas como ”não”. Portanto, será necessário fazer um balanceamento dos dados para que a incidência dos valores do atributo alvo não interfiram na acurácia do modelo. É importante ressaltar que o balanceamento é feito apenas nos dados de treinamento do modelo.

Já com as informações primordiais referentes ao *dataset*, é feito um estudo sobre os atributos obtidos e iniciado o processo de *Feature Engineering*. Esse processo é essencial para projetos de aprendizado de máquina pois é incerto o caminho percorrido entre os dados iniciais até a obtenção de respostas. Por conta disso, é importante que eles contenham informações de alta compatibilidade com o contexto da análise que será realizada. Portanto, nessa etapa será feita a padronização dos dados, a filtragem atributos relevantes

e que ajudarão a alcançar o objetivo sem que interfiram negativamente nos resultados e também serão criados atributos novos.

Com os dados já separados, inicia-se a etapa de modelagem, na qual serão usados três modelos diferentes com o objetivo de prever a ocorrência do Transtorno do Humor bipolar: *Regressão Logística, Árvore de Decisão e XGBoost*. Para seu treinamento, será dividido o *dataset* para que 20% dele seja usado para o conjunto de testes e os 80% restantes para o conjunto de treinamento. A partir dos modelos prontos, serão analisados os dados de acurácia e serão feitas relações com os modelos que foram utilizados nos trabalhos relacionados para identificar qual que melhor se adéqua ao objetivo do trabalho. Também será feita a otimização dos hiperparâmetros do modelo, buscando quais os melhores parâmetros para cada um deles e medindo o quão relevantes eles são para os resultados finais.

4. Análise de resultados

A partir dos padrões coletados pelo modelo com o objetivo de validar o que foi encontrado e verificar se o comportamento apresentado é o esperado, é feita uma avaliação em quatro etapas para entender o efeito de cada uma delas nos resultados do modelo. Para isso, foram medidos resultados para os dados sem nenhum ajuste, depois para os dados apenas com a etapa de pré processamento, em seguida para os dados com pré processamento e engenharia de atributos, e por último dados com pré processamento, engenharia de atributos e balanceamento. A Fig. 3 mostra os resultados obtidos para os modelos em cada uma das etapas avaliadas.

Dados sem nenhum ajuste

Score	Árvore de Decisão	Regressão Logística	Gradient Boosting Classifier
Acurácia	86,92%	86,27%	86,27%
F1	47%	32,25%	32,25%
Precisão	60%	62,5%	62,5%

Dados com pré processamento

Score	Árvore de Decisão	Regressão Logística	Gradient Boosting Classifier
Acurácia	80%	85%	85%
F1	33,33%	33,33%	36,84%
Precisão	32%	46,15%	46,66%

Dados com pré processamento e engenharia de atributos

Score	Árvore de Decisão	Regressão Logística	Gradient Boosting Classifier
Acurácia	81,25%	83,75%	86,87%
F1	28,57%	31,57%	40%
Precisão	33,33%	42,28%	63,63%

Dados com pré processamento, engenharia de atributos e balanceamento

Score	Árvore de Decisão	Regressão Logística	Gradient Boosting Classifier
Acurácia	80,72%	78,75%	83,12%
F1	32,40%	41,37%	47,05%
Precisão	34,34%	35,29%	44,44%

Figure 3. Resultados obtidos em cada uma das etapas
Elaborado pela autora (2023)

5. Conclusão e trabalhos futuros

Neste trabalho foram introduzidos os conceitos do Transtorno do Humor bipolar e das técnicas Mineração de dados e Aprendizado de Máquina. Também foi explicado seu objetivo: utilizar essas técnicas em um conjunto de dados fornecido pelo grupo de pesquisadores da Universidade Católica de Pelotas para prever a ocorrência do THB.

O estudo de literaturas existentes para prever o THB com o uso de questionários e técnicas de mineração de dados e aprendizado de máquinas demonstrou resultados satisfatórios e com uma boa acurácia a partir dos métodos utilizados, evidenciando a possibilidade de prever o THB a partir de questionários utilizando os algoritmos mencionados.

Com isso, a proposta de solução considerou a metodologia CRISP-DM para que o ciclo de suas etapas fosse seguido no desenvolvimento do projeto. Foram realizadas as etapas de entendimento de dados, pré processamento e engenharia de atributos, que se mostrou um pouco mais complexo por conta da dificuldade em relacionar os atributos já existentes para a criação de novos.

Por fim, com a avaliação dos modelos, o melhor F1 encontrado foi para o *Gradient Boosting Classifier* atingindo um valor de 47,05%, juntamente com uma acurácia de 83,12%. Apesar de todas as etapas terem sido seguidas e o melhor modelo encontrado ter passado por todas elas, ao analisar os resultados percebe-se que ainda há espaço para que melhorias nos processos sejam feitas, sugerindo uma melhoria no processo de engenharia de atributos principalmente.

Ao fazer uma comparação com os trabalhos relacionados de mesmo desfecho, em [Jadhav et al. 2019] não são apresentados resultados referentes à outras métricas que não sejam acurácia, limitando as comparações. Já em [R.Saranya 2022], o trabalho apresenta as mesmas métricas, sendo que seus resultados foram mais promissores, porém, existem diferenças entre o tipo de dado utilizado para diagnosticar o THB, visto que o presente trabalho usa o diagnóstico de sim ou não, o trabalho relacionado utiliza o tipo de episódio depressivo no qual o paciente se encontra, o que pode explicar a diferença entre os resultados.

Apesar de os resultados de acurácia apresentados no trabalho terem sido satisfatórios, o fato de os resultados de F1 e precisão estarem baixos indica que é necessária uma melhora no processo de engenharia de atributos, o que foi evidenciado ao testar o modelo com um conjunto de dados diferente. Essa foi a maior dificuldade encontrada no decorrer do desenvolvimento deste trabalho visto que, por lidar com dados de questionário que tratam de padrões comportamentais, transtornos, traumas e sentimentos, é necessário que um profissional especializado seja consultado para entender melhor como se dão as interações entre eles para que novos atributos relevantes sejam criados. Também foi observada a presença de *overfitting* nos resultados, que também pode ser amenizado com uma engenharia de atributos mais elaborada. Portanto, seria interessante trabalhos que se aprofundem na questão de como buscar uma melhora na engenharia de atributos para dados de questionários para prever o THB e verificar se com isso o *overfitting* pode ser evitado ou se seriam necessárias outras técnicas.

References

Aggarwal, C. C. et al. (2015). *Data mining: the textbook*, volume 1. Springer.

American Psychiatric Association, D., Association, A. P., et al. (2013). *Diagnostic and statistical manual of mental disorders - DSM-5*, volume 21.

Azar, G., Gloster, C., El-Bathy, N., Yu, S., Neela, R. H., and Alothman, I. (2015). Intelligent data mining and machine learning for mental health diagnosis using genetic algorithm. In *2015 IEEE International Conference on Electro/Information Technology (EIT)*, pages 201–206. IEEE.

Chen, T. and Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 785–794, New York, NY, USA. ACM.

Coggon, D., Barker, D., and Rose, G. (1997). Epidemiology for the uninitiated.

Fears, S. C. and Reus, V. I. (2015). Chapter 104 - bipolar disorder. In Rosenberg, R. N. and Pascual, J. M., editors, *Rosenberg's Molecular and Genetic Basis of Neurological and Psychiatric Disease (Fifth Edition)*, pages 1275–1291. Academic Press, Boston, fifth edition edition.

Grande, I., Berk, M., Birmaher, B., and Vieta, E. (2016). Bipolar disorder. *The Lancet*, 387(10027):1561–1572.

Hastie, T., Tibshirani, R., and Friedman, J. (2001). *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc., New York, NY, USA.

Hilbe, J. M. (2015). *Practical Guide to Logistic Regression*. Springer Series in Statistics. Taylor Francis Group, Arizona, USA.

Jadhav, R., Chellwani, V., Deshmukh, S., and Sachdev, H. (2019). Mental disorder detection: Bipolar disorder scrutinization using machine learning. In *2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, pages 304–308. IEEE.

Jha, I. P., Awasthi, R., Kumar, A., Kumar, V., and Sethi, T. (2021). Learning the mental health impact of covid-19 in the united states with explainable artificial intelligence: Observational study. *JMIR mental health*, 8(4):e25097.

MITCHELL, T. (1997). *Machine Learning. 1. ed.* McGraw-Hill.

Müller-Oerlinghausen, B., Berghöfer, A., and Bauer, M. (2002). Bipolar disorder. *The Lancet*, 359(9302):241–247.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Perez Arribas, I., Goodwin, G. M., Geddes, J. R., Lyons, T., and Saunders, K. E. (2018). A signature-based machine learning model for distinguishing bipolar disorder and borderline personality disorder. *Translational psychiatry*, 8(1):1–7.

R.Saranya, D. N. (2022). Bd-mdl: Bipolar disorder detection using machine leanring and deep learning. *Journal of Pharmaceutical Negative Results*, 13:5892–5905.

RUSSEL, Stuart J., N. (2010). *Artificial Intelligence: A Modern Approach.* Pearson Higher Education, New Jersey.

Sokolova, M. and Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, 45(4):427–437.

Srividya, M., Mohanavalli, S., and Bhalaji, N. (2018). Behavioral modeling for mental health using machine learning algorithms. *Journal of medical systems*, 42(5):1–12.

Stuart Russell, P. N. (1995). *Artificial Intelligence: A Modern Approach*. Prentice Hall, 1st edition.

Tan, P.-N., Steinbach, M., Karpatne, A., and Kumar, V. (2018). *Introduction to Data Mining*. Pearson.

Wirth, R. and Hipp, J. (2000). Crisp-dm: Towards a standard process model for data mining. In *Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining*, volume 1, pages 29–40. Manchester.

Zoabi, Y., Deri-Rozov, S., and Shomron, N. (2021). Machine learning-based prediction of covid-19 diagnosis based on symptoms. *npj digital medicine*, 4(1):1–5.