Aprendizagem de Máquina: O Algoritmo de Naive Bayes, a Classificação e a comparação com outros algoritmos probabilísticos

Felipe Fernandes
Instituto de Ciência e Tecnologia
Universidade Federal de São Paulo
São José dos Campos, Brasil
fernandes.felipe@unifesp.br

Henrique Dedini
Instituto de Ciência e Tecnologia
Universidade Federal de São Paulo
São José dos Campos, Brasil
henrique.carvalho@unifesp.br

***Resumo*—O presente artigo apresenta uma análise abrangente do algoritmo de Naive Bayes, uma técnica popular de aprendizagem de máquina supervisionada aplicada em problemas de classificação, que se baseia no Teorema de Bayes e na suposição simplificada de independência condicional entre atributos. São descritos os conceitos que envolvem esta técnica, o algoritmo escolhido, suas variações, a implementação em diversas bases de dados e a comparação com outros algoritmos de classificação probabilísticos.**

***Palavras-chaves*—aprendizado, algoritmos probabilísticos, técnica, classificação, supervisão**

I. INTRODUÇÃO

Nas últimas décadas, os avanços tecnológicos permitiram muitas facilidades, tanto do ponto de vista de *software* quanto de *hardware*. A capacidade de processamento de dados, por exemplo, têm auxiliado diversas pessoas de várias áreas, uma vez que cálculos impossíveis de serem feitos de forma manual são realizados em questão de segundos por dispositivos computacionais.

Isso fez com que tarefas de reconhecimento de padrões e predição de resultados se tornassem viáveis graças ao alto desempenho computacional que esses dispositivos possuem (FAWCETT; PROVOST, 2018).

Comumente falada e cada vez mais em evidência, a Inteligência Artificial e sua área de Aprendizagem de Máquina compreendem diversos conceitos e técnicas para a constante evolução das suas aplicações no mundo real. A sua principal ideia se concentra no desenvolvimento de algoritmos e modelos capazes de aprender padrões e tomar decisões a partir de dados.

Podemos afirmar que a base de dados é a matéria prima para qualquer atividade dentro desse contexto, dado que, através disso, torna-se possível extrair padrões, singularidades e outras informações a partir de um conjunto de características.

Baseado no exposto, torna-se possível discorrer sobre a aplicação em problemas concretos, detalhando pontos importantes e evidenciando como esta técnica pode auxiliar o mundo real.

II. FUNDAMENTAÇÃO TEÓRICA

A. Conceitos Fundamentais

Um dos principais conceitos a serem entendidos nessa área é a classificação, que consiste no processo de identificar um modelo ou função que descreve diferentes classes de dados, rotulando novas instâncias conforme sua base de informações, baseando-se no valor dos atributos dos mesmos (Han e Kamber, 2006).

Nesse contexto, o algoritmo de Naive Bayes e outros modelos relacionados surgem como técnicas amplamente utilizadas e eficazes para problemas desse tipo**.**

O primeiro, **Naive Bayes**, é um algoritmo de aprendizagem supervisionada que se baseia no Teorema de Bayes e na suposição de independência condicional entre os atributos. Essa suposição simplificada é chamada de "ingênua" (naive), pois assume que a presença ou ausência de um determinado atributo não está relacionada à presença ou ausência de outros atributos. Apesar dessa simplificação, o Naive Bayes tem se

mostrado surpreendentemente eficiente em várias aplicações práticas.

Seu funcionamento é relativamente simples e pode ser descrito em alguns passos:

- Preparação dos dados: os dados de treinamento são organizados em um conjunto de exemplos rotulados, onde cada exemplo possui um conjunto de atributos e uma classe associada.
- Estimação das probabilidades a priori: o algoritmo calcula as probabilidades a priori de cada classe com base na frequência com que cada classe ocorre no conjunto de treinamento.
- Estimação das probabilidades condicionais: para cada atributo, o algoritmo estima as probabilidades condicionais de cada classe. Isso envolve calcular a probabilidade de um determinado valor do atributo ocorrer, dado que a classe é conhecida.
- Classificação de novos exemplos: quando um novo exemplo não rotulado é apresentado ao algoritmo, ele utiliza as probabilidades a priori e as probabilidades condicionais para calcular a probabilidade de pertencer a cada classe. O exemplo é então atribuído à classe com a maior probabilidade.

É importante destacar que existem diversas variantes do algoritmo de Naive Bayes, que se diferenciam principalmente pela forma como tratam os atributos ou assumem diferentes distribuições. Algumas das variantes mais comuns são:

- **Naive Bayes Gaussiano:** adequado para atributos numéricos contínuos que seguem uma distribuição gaussiana (normal).
- **Naive Bayes Multinomial:** usado quando os atributos são representados por contagens ou frequências, sendo comumente aplicado em problemas de classificação de texto.
- **Naive Bayes Bernoulli:** semelhante ao Multinomial Naive Bayes, mas é adequado para atributos binários, onde ocorre a presença ou ausência.
- **Naive Bayes Complementar:** lida com o problema de classes desbalanceadas, ajustando as probabilidades de cada classe de forma complementar.
- **Naive Bayes Averaged One-Dependence Estimators (AODE):** considera dependências entre os atributos, embora mantenha a suposição de independência condicional.

Essas variantes adaptam o algoritmo para diferentes tipos de dados e suposições específicas, permitindo que ele seja aplicado em uma variedade de cenários de classificação. Cada variante tem suas próprias características e é selecionada com base nas características do conjunto de dados e do problema em questão.

Outro algoritmo abordado, a **Regressão Logística**, é uma técnica de análise de dados que também faz uso da matemática para encontrar as relações entre dois fatores de dados. Consiste em prever uma variável binária ou dicotômica com base em um conjunto de variáveis independentes.

Ao contrário da regressão linear, que é usada para prever valores contínuos, a regressão logística é usada para modelar a probabilidade de ocorrência de um evento, atribuindo uma probabilidade a cada possível resultado binário. Em outras palavras, estima a probabilidade de um evento pertencer a uma das duas categorias possíveis. A previsão, portanto, geralmente tem um número finito de resultados.

A regressão logística utiliza-se da função logística (também conhecida como função sigmoide) para modelar a relação entre as variáveis independentes e a variável dependente. Além disso, mapeia qualquer valor real para um valor entre 0 e 1, representando a probabilidade de um evento ocorrer. Assim, calcula uma combinação linear das variáveis independentes e aplica a função logística para gerar a probabilidade predita.

Durante o treinamento do modelo de regressão logística, os coeficientes das variáveis independentes são ajustados de forma a maximizar a verossimilhança dos dados observados. Isso é geralmente feito utilizando métodos de otimização, como a maximização da verossimilhança condicional ou o método dos mínimos quadrados.

Após o treinamento, o modelo pode ser usado para fazer previsões, atribuindo uma classe (0 ou 1) com base na probabilidade estimada.

Há três abordagens para análise de regressão logística com base nos resultados da variável dependente:

- **Regressão Logística Binária:** tipo mais comum de regressão logística. A variável dependente pode ter apenas dois valores, como sim e não ou 0 e 1. Embora a função logística calcule um intervalo de valores entre 0 e 1, o modelo de regressão binária arredonda a resposta para os valores mais próximos. Geralmente, respostas abaixo de 0,5 são arredondadas para 0, e respostas acima de 0,5 são arredondadas para 1, para que a função logística retorne um resultado binário.
- **Regressão Logística Multinomial:** a variável dependente possui mais de duas categorias mutuamente exclusivas. Pode

analisar problemas que tenham vários resultados possíveis, desde que o número de resultados seja finito. Por exemplo, ela é capaz de prever se os preços das casas aumentarão em 25%, 50%, 75% ou 100% com base em dados populacionais, mas não pode prever o valor exato de uma casa. Funciona mapeando os valores dos resultados para diferentes valores entre 0 e 1. Como a função logística pode retornar um intervalo de dados contínuos, como 0,1, 0,11, 0,12 e assim por diante, a regressão multinomial também agrupa a saída para os valores mais próximos possíveis.

- **Regressão Logística Ordinal:** utilizada quando a variável dependente é ordinal, ou seja, possui categorias ordenadas, mas sem uma distância quantitativa definida entre elas. Por exemplo, a regressão ordinal seria usada para prever a resposta a uma pergunta de pesquisa que pede para os clientes classificarem seu serviço como ruim, regular, bom ou excelente com base em um valor numérico, como o número de itens que eles compram de você ao longo do ano.

É importante escolher o tipo de regressão logística adequado ao problema em questão, levando em consideração a natureza dos dados e os objetivos da análise.

O próximo modelo abordado é a **Máquina de Vetores de Suporte (SVM)**, outro algoritmo de aprendizado de máquina supervisionado utilizado para problemas de classificação que visa construir um hiperplano ou conjunto de hiperplanos em um espaço de alta dimensionalidade que possa separar ou classificar os pontos de dados em diferentes categorias. Logo, podemos definir o objetivo principal da SVM como sendo encontrar o hiperplano que maximize a margem entre as classes de dados. Essa margem é definida como a distância entre o hiperplano de separação e os pontos de dados mais próximos de cada classe, conhecidos como vetores de suporte. Esses vetores de suporte são os pontos de dados mais críticos para determinar o hiperplano ótimo.

Em outras palavras, considerando um problema binário, o objetivo da SVM é separar as instâncias das duas classes através de uma função que será obtida a partir dos exemplos conhecidos na fase de treinamento. O intuito é produzir um classificador que funcione de forma adequada com exemplos não conhecidos, ou seja, exemplos que não foram aplicados durante o treinamento, adquirindo assim a capacidade de predizer as saídas de futuras novas entradas.

Embora tenha uma teoria um pouco mais complexa e rebuscada se comparada aos demais algoritmos, pode ser usado de várias formas, diferenciando diversos fatores e trabalhando com dados de alta dimensionalidade, tornando-se um modelo muito robusto e eficiente.

A SVM possui diferentes formas de lidar com problemas de classificação, dependendo da natureza dos dados e da separabilidade das classes. As principais variantes da SVM são:

- **SVM de Margem Rígida:** utilizado quando os dados são linearmente separáveis, ou seja, é possível traçar um hiperplano que separe completamente as classes sem erros. Nesse caso, a SVM busca encontrar o hiperplano com a maior margem possível.
- **SVM de Margem Suave:** aplicado quando os dados não são linearmente separáveis. Nesse caso, a SVM permite que certos pontos de dados sejam classificados erroneamente para obter uma separação melhor. Esses pontos incorretamente classificados são conhecidos como violações de margem.
- **SVM com Kernel:** uma extensão da SVM que permite lidar com problemas de classificação não lineares, onde os dados não podem ser separados por um hiperplano linear. A SVM com kernel mapeia os dados para um espaço dimensional mais alto, onde eles podem se tornar linearmente separáveis. Alguns dos kernels comumente usados são o kernel linear, o kernel polinomial e o kernel de função de base radial (RBF).

Outro ponto que torna esse modelo tão importante é o fato de poder se estender para problemas de regressão, conhecido como SVR.

Por fim, outro modelo estudado é a **Floresta Aleatória**, que através de outro conceito de aprendizado supervisionado (Árvores de Decisão), consegue executar tarefas de classificação.

Esse algoritmo combina múltiplas árvores de decisão, compondo um conjunto de árvores individuais, onde cada árvore é treinada com uma amostra aleatória do conjunto de dados de treinamento.

Além disso, durante o treinamento de cada árvore, em cada nó, é realizada uma seleção aleatória de um subconjunto de características (variáveis independentes) para determinar a melhor divisão. Esses dois mecanismos de aleatoriedade, amostragem de dados e seleção de características, são responsáveis pelo nome "Floresta Aleatória".

Durante o processo de treinamento, cada árvore é construída usando um algoritmo de árvore de decisão, como o algoritmo CART (Classificação e Regressão por Árvore). Cada árvore é treinada para realizar uma tarefa de classificação (quando se trata

de classificação) ou predição de valores (quando se trata de regressão). No caso de classificação, a Floresta Aleatória realiza uma votação entre as árvores para determinar a classe final prevista, enquanto na regressão, é calculada a média ou mediana das previsões das árvores.

O algoritmo de Floresta Aleatória adiciona aleatoriedade extra ao modelo quando está criando as árvores. Ao invés de procurar pela melhor característica ao fazer a partição de nodos, ele busca a melhor característica em um subconjunto aleatório das características. Este processo cria uma grande diversidade, o que geralmente leva a geração de modelos melhores.

Portanto, quando criando uma árvore na Floresta Aleatória, apenas um subconjunto aleatório das características é considerado na partição de um nó. Porém, é possível fazer as árvores ficarem mais aleatórias utilizando limiares (thresholds) aleatórios para cada característica, ao invés de procurar pelo melhor limiar (como uma árvore de decisão geralmente faz).

Toda essa preparação e informações torna possível a comparação com outros modelos probabilísticos mencionados anteriormente.

Os algoritmos probabilísticos, assim como Naive Bayes, são baseados em princípios probabilísticos e estatísticos para fazer inferências e tomar decisões. A análise e interpretação desses tipos de abordagens permitem um maior conhecimento acerca do que está sendo tratado.

Outro conceito de grande relevância e que será utilizado neste trabalho diz respeito às medidas de avaliação de um modelo.

As medidas de avaliação de um modelo de aprendizado de máquina são métricas utilizadas para avaliar e quantificar o desempenho do modelo em relação aos dados de teste ou validação. Essas medidas fornecem informações sobre a qualidade das previsões ou classificações feitas pelo modelo e permitem comparar diferentes modelos ou ajustes de parâmetros.

Em um problema de classificação, há duas soluções possíveis: erro ou acerto. Alguns dos conceitos mais importantes para a classificação binária gira em torno das classes que os dados preditos podem receber a partir de verdadeiro ou falso: VP, VN, FP e FN.

- **Verdadeiro Positivo (VP):** quando o método diz que a classe é positiva e, ao verificar a resposta, vê-se que a classe era realmente positiva;
- **Verdadeiro Negativo (VN):** quando o método diz que a classe é negativa e, ao verificar a resposta, vê-se que a classe era realmente negativa;
- **Falso Positivo (FP):** quando o método diz que a classe é positiva, mas ao verificar a resposta, vê-se que a classe era negativa;
- **Falso Negativo (FN):** quando o método diz que a classe é negativa, mas ao verificar a resposta, vê-se que a classe era positiva.

Uma maneira mais simples de representar dados de um método de classificação é através da **Matriz de Confusão**. Esta indica a quantidade de ocorrências que o modelo teve para cada uma das quatro categorias mencionadas anteriormente.

Através disso, podemos obter os valores de acertos ou erros das predições com um maior embasamento.

Outra métrica simples porém importante é a **Acurácia**, que tem como intuito avaliar o percentual de acertos do modelo. Pode ser obtida facilmente através da razão entre a quantidade de acertos e o total de entrada:

$$acurácia = \frac{Total\ de\ acertos}{Total\ de\ entradas} \quad (1)$$

Utilizando os conceitos anteriormente apresentados, podemos obter também pela seguinte fórmula:

$$acurácia = \frac{VP + VN}{VP + FN + VN + FP} \quad (2)$$

Dentre outras métricas, mais duas são essenciais para este estudo: **Precisão** e **F-score**.

Diferenciando-se da acurácia, a precisão avalia a quantidade de verdadeiros positivos sobre a soma de todos os outros valores positivos, isto é:

$$precisão = \frac{VP}{VP + FP} \quad (3)$$

Por outro lado, a F-measure, F-score ou F1 é uma média harmônica calculada com base em outras duas métricas: a própria precisão e **revocação (recall)**, que indica das amostras positivas, quantas o modelo conseguiu classificar corretamente. A F-score pode ser obtida a partir da seguinte equação:

$$F1 = \frac{2 * precisão * revocação}{precisão + revocação} \quad (4)$$

Além dessas medidas, dependendo do problema e do contexto, podem ser utilizadas outras métricas específicas, como Área sob a Curva ROC (AUC-ROC), Log Loss, Mean Absolute Error (MAE) ou Root Mean Squared Error (RMSE), entre outras. Elas não serão usadas para o contexto deste trabalho mas é válido ressaltar sua existência.

A escolha das medidas de avaliação adequadas depende do tipo de problema, das classes envolvidas, do desbalanceamento de classes, das consequências dos erros de previsão e das metas específicas do projeto.

B. Trabalhos Relacionados

Um dos principais temas que envolvem esse trabalho é o Teorema de Bayes e as Redes Bayesianas, essenciais para o estudo da teoria de probabilidade e estatística.

Segundo Jaynes (2003), a visão Bayesiana de probabilidade diz que "a probabilidade é o que mede o grau de incerteza sobre alguma coisa ou acontecimento, ela representa também o quanto é a chance de um determinado evento ocorrer devido certas circunstâncias".

Ainda segundo Jaynes (2003), o teorema de Bayes se baseia em três pontos principais para encontrar as probabilidades dos acontecimentos:

- **Representação numérica:** o grau de certeza ou incerteza sobre o evento, representado numericamente por uma função matemática.
- **Correspondência qualitativa com o senso comum:** quando uma informação sobre ou a favor de um evento é obtida, a probabilidade do acontecimento é aumentada.
- **Consistência:** duas maneiras diferentes de se chegar em um resultado devem possuir o mesmo valor.

Outros conceitos bastante importantes para o assunto é o da probabilidade condicional, que consiste basicamente na probabilidade de um evento A ocorrer, quando sabemos que um outro evento B já tenha ocorrido (SWEENEY, 2014).

Menciona-se também as probabilidades a priori e posteriori. O primeiro representa a probabilidade de um fato ter ocorrido ou não, enquanto o segundo representa a probabilidade revista com base nas novas informações obtidas (ANDERSON, 2005).

O Teorema de Bayes (ou Teorema das Causas), então, pode ser entendido da seguinte maneira segundo Paulino, Turkman e Murteira (2003):

"Uma interpretação do teorema de Bayes consiste em considerar os eventos Ai como "causas" do evento B, sendo atribuído probabilidades deste evento atuar na ocorrência de B. Esta probabilidade é calculada antes da realização do experimento, sendo designada como a probabilidade a priori de Ai . Após a realização do experimento, é conhecido que o evento B ocorreu, então a probabilidade a priori é revista por meio da fórmula de Bayes e então passa a atribuir aos eventos Ai , i= 1,2,...n as probabilidades a posteriori P(Ai |B), i = 1,2,...n (CRAMÉR , 1955)".

Dessa forma, de acordo com "Redes Bayesianas" [25], consegue-se observar que esse Teorema é um dos poucos resultados matemáticos que se propõe a caracterizar a aprendizagem com a experiência, ou seja, a modificação de atitude inicial em relação às "causas" depois de ter a informação adicional de que certo acontecimento se realizou.

As Redes Bayesianas, portanto, podem ser entendidas como uma representação compacta de uma tabela de conjunção de probabilidades do universo do problema, matematicamente falando (MARQUES; DUTRA, 2008). Por outro lado, podem também ser entendidas como um modelo gráfico que representa de forma simples as relações de causalidade das variáveis de um sistema (STUART; NORVIG, 1995).

Uma Rede Bayesiana consiste do seguinte:

- Um conjunto de variáveis e um conjunto de arcos ligando as variáveis;
- Cada variável possui um conjunto limitado de estados mutuamente exclusivos;
- As variáveis e arcos formam um grafo dirigido sem ciclos (DAG);
- Para cada variável A que possui como pais B1, ..., Bn, existe uma tabela P(A| B1, ..., Bn).

Essa rede pode ser vista como um modelo que utiliza Teoria dos Grafos, condições de Markov e distribuição de probabilidades para representar uma situação, suas variáveis e estados e a partir disso realizar inferências. A estrutura do grafo captura as relações causais entre as variáveis, enquanto as tabelas de probabilidade condicional especificam as probabilidades de cada variável, condicionadas às suas variáveis pai no grafo.

Em "Segmentação da Iris em imagens com ruído" [1], os autores sacrificaram a precisão para aumentar a velocidade e eficiência da segmentação das imagens das flores. Gabriel Barreto Moura [2] trabalha com NB em sistemas *fuzzy,* propondo uma técnica de relação entre o algoritmo probabilístico e as redes bayesianas. em "SOFTWARE CLASSIFICADOR DE ESPÉCIE DE PLANTAS UTILIZANDO O MODELO PROBABILÍSTICO BAYESIANO" [3], o autor faz uso do NB para responder a pergunta chave de sua pesquisa, "*Existe a possibilidade de realizar predições de espécies de plantas com base em suas características utilizando modelos probabilísticos?*". Os autores de "*Uma implementação do algoritmo Naïve Bayes para classificação de texto*" [4] fazem uso do algoritmo NB para criar uma ferramenta de classificação de texto. Os livros "*Data Mining: Concepts and Techniques*" [8] e "*Data Science para negócios: O que você precisa saber sobre mineração de dados e pensamento analítico de dados*" [9] apresentam uma abordagem didática mais completa sobre o contexto geral de Data Mining e algoritmos probabilísticos, fornecendo uma base de conhecimento geral sobre o tema. Em "*A New Model for Iris Classification Based on Naïve Bayes*

Grid Parameters Optimization" [10] os autores fazem uso do algoritmo Naive Bayes para classificar um dataset contendo informações sobre a flor de Íris, a mesma ideia desse trabalho, porém utilizam de uma técnica baseada em uma grade de busca, para melhorar a eficiência. Os autores de "*Heart diseases detection using Naive Bayes algorithm*" [11] abordam o tema de utilização de algoritmos probabilísticos no campo da medicina, com sua utilização focada em predição de doenças, no caso, problemas cardíacos.

Já Masud Karim e Rashedur M Rahman [12] fizeram um trabalho de comparação entre NB e a árvore de decisão C4.5 quando utilizados em problemas de análise comportamental de clientes de bancos. Seguindo a linha de comparação entre algoritmos, em "*Comparing Naive Bayes, Decision Trees, and SVM with AUC and Accuracy*" [13] vemos a comparação entre o NB com outros algoritmos, o mais importante para este trabalho sendo o SVM.

C. *Objetivo*

A partir da ideia de classificação e da perspectiva de Design Science (WIERINGA, 2014), um paradigma de pesquisa com foco no desenvolvimento e validação de conhecimento prescritivo em ciência da informação, foi realizável criar um modelo que conseguia abordar a seguinte questão: *Existe a possibilidade de realizar predições de diferentes bases de dados com base em suas características utilizando Naive Bayes e comparando com outros modelos probabilísticos?*

III. METODOLOGIA EXPERIMENTAL

Revisando a literatura e baseando-se no processo de descoberta (VASCONCELOS; CARVALHO, 2018) e nos passos que permeiam a ideia geral de Naive Bayes exposta anteriormente, foram definidas sete fases para se atingir o objetivo:

1. Limpeza dos Dados
2. Integração dos Dados
3. Seleção dos Dados
4. Transformação dos Dados
5. Mineração dos Dados
6. Avaliação dos Dados
7. Visualização dos Dados

Com essas informações, podemos partir para o processo de validação do projeto, usando a linguagem *Python* e seus frameworks (tais como Pandas, Numpy, Seaborn, MatPlotLib, SciKit-Learn e afins).

Utilizaremos várias bases de dados do mundo real que consigam contemplar diversas áreas de aplicação.

Como exemplo simples da aplicação de NB Gaussiano, podemos mencionar e utilizar as bases que possuam informações sobre o comprimento e largura das sépalas e pétalas das plantas Iris Setosa, Iris Versicolor e Iris Virginica (THE PLANT LIST, 2010).

Essa base de dados reais utilizada neste estudo será obtida a partir do repositório digital UCI gerenciado pelo centro de aprendizado de máquina e sistemas inteligentes da Universidade da Califórnia Irvine (DHEERU; KARRA, 2017), que conta com cinquenta instâncias de cada espécie mencionada acima. Os recursos do modelo são as medidas de comprimento e largura das sépalas e pétalas, ou seja, todos os recursos são numéricos. Adaptações podem ser feitas para os demais algoritmos, caso preciso.

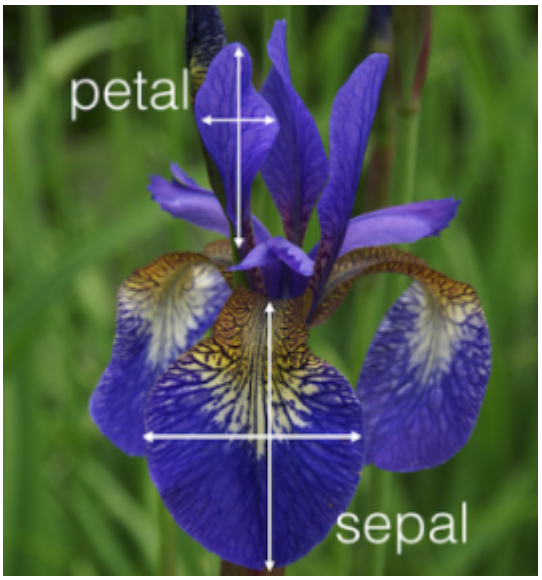

Figura 1: Pétalas e sépalas da Íris. Fonte.

Embora esse clássico exemplo nos dê uma boa visão de como o algoritmo em foco funciona, é necessário explorar outras bases e modelos a fim de ter uma visão ampla das possibilidades que podem ser trabalhadas.

Para esse fim, pretende-se utilizar bases relacionadas à identificação de doenças e/ou informações relacionadas usando algoritmos probabilísticos, tais como os citados acima: Regressão Logística, Máquina de Vetores de Suporte e Floresta Aleatória.

Baseando-se nas informações exibidas, torna-se possível a criação de um diagrama de blocos que descreve as etapas do Pipeline Experimental para a execução do trabalho:

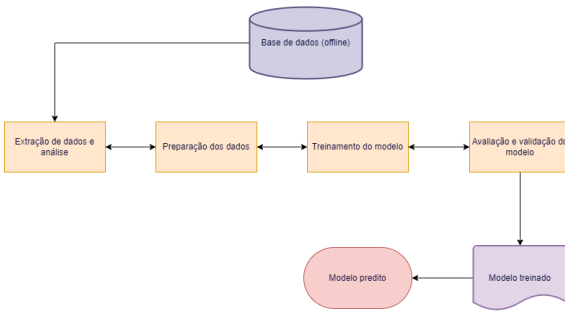

Figura 2: Diagrama de blocos. Fonte: Autor

As etapas estão separadas da seguinte forma:

- **Base de dados (offline):** necessário reunir os dados relevantes para o problema em questão. Isso envolve a busca por conjuntos de dados existentes, coleta de dados de sensores ou dispositivos, extração de informações de bancos de dados, entre outras fontes.
- **Extração de dados e análise:** necessário realizar uma análise exploratória para compreender melhor o conjunto de dados. Isso envolve a identificação e o entendimento das diferentes variáveis presentes, a análise de sua distribuição e características, a detecção de possíveis problemas, como valores ausentes ou discrepantes, e a identificação de correlações entre as variáveis.
- **Preparação dos dados:** os dados precisam ser preparados para o treinamento do modelo. Isso pode incluir a remoção de valores ausentes, a normalização ou padronização das variáveis, a codificação de variáveis categóricas, a seleção de características relevantes e a divisão dos dados em conjuntos de treinamento, validação e teste.
- **Treinamento do modelo:** fase em que o modelo é construído e treinado com base nos dados preparados. Isso envolve a escolha do algoritmo adequado para o problema em questão, a definição dos hiperparâmetros do modelo e a execução do processo de treinamento. Durante o treinamento, o modelo aprenderá a mapear os padrões nos dados e ajustar seus parâmetros internos para realizar previsões ou tomar decisões.
- **Avaliação e validação do modelo:** necessário avaliar o desempenho do modelo para verificar sua capacidade de generalização em dados não vistos anteriormente. Isso pode ser feito usando técnicas como validação cruzada, onde o modelo é avaliado em diferentes partições dos dados, ou dividindo os dados em conjunto de treinamento e conjunto de teste. Métricas adequadas são utilizadas para medir o desempenho do modelo, como acurácia, precisão, recall, F1-score, entre outras.
- **Modelo treinado:** após a etapa de treinamento e validação, o modelo é considerado treinado. Isso significa que ele foi ajustado aos dados disponíveis e está pronto para fazer previsões ou tomar decisões com base em novos dados de entrada. O modelo contém os parâmetros aprendidos durante o treinamento e se encontra em um estado em que pode ser usado para inferências.
- **Modelo predito:** quando o modelo treinado é alimentado com novos dados de entrada, ele usa o conhecimento adquirido durante o treinamento para fazer previsões ou classificações. Essa etapa envolve a passagem dos dados de entrada pelo modelo e a obtenção de uma resposta ou resultado predito. Dependendo do tipo de problema, o modelo pode gerar uma previsão numérica, uma classificação em categorias ou até mesmo uma probabilidade associada a cada classe. O resultado obtido é a resposta do modelo às informações fornecidas.

Descritos os passos, define-se as medidas de avaliação adotadas. Para o nosso contexto, será considerado: acurácia e F-score, tendo a precisão incluída neste último, além da matriz de confusão sendo exemplificada na aplicação do algoritmo de Naive Bayes na base de dados da planta Íris.

A. Base de Dados

A primeira das bases de dados usadas diz respeito à predição da Esclerose Múltipla, baseado em um estudo de coorte prospectivo realizado no México entre 2006 a 2010 em pacientes recém-diagnosticados com CIS (Síndrome Clínica Isolada, o primeiro episódio neurológico de uma pessoa). Foi obtido através da plataforma *Kaggle*.

A descrição das colunas da base é a seguinte:

- **ID:** identificador do paciente (int)
- **Age (idade):** idade do paciente (em anos)
- **Schooling (escolaridade):** tempo que o paciente passou na escola (em anos)
- **Gender (gênero):** 1=masculino, 2=feminino
- **Breastfeeding (amamentação):** 1: sim, 2: não, 3: desconhecido
- **Varicella (catapora):** 1: positivo, 2: negativo, 3: desconhecido

- **Initial_Symptoms (sintomas iniciais):** 1: visuais, 2: sensoriais, 3: motores, 4: outros, 5: visuais e sensoriais, 6: visuais e motores, 7: visuais e outros, 8: sensoriais e motores, 9: sensoriais e outros, 10: motora e outra, 11: visual, sensorial e motora, 12: visual, sensorial e outra, 13: visual, motora e outra, 14: sensorial, motora e outra, 15: visual, sensorial, motora e outra
- **Mono_or_Polysymptomatic (mono ou polissintomático):** 1: monossintomático, 2: polissintomático, 3: desconhecido
- **Oligoclonal_Bands (bandas oligoclonais):** 0: negativo, 1: positivo, 2: desconhecido
- **LLSSEP:** 0: negativo, 1: positivo
- **ULSSEP:** 0: negativo, 1: positivo
- **VEP:** 0: negativo, 1: positivo
- **BAEP:** 0: negativo, 1: positivo
- **Periventricular_MRI (ressonância magnética periventricular):** 0: negativo, 1: positivo
- **Cortical_MRI (ressonância magnética cortical):** 0: negativo, 1: positivo
- **Infratentorial_MRI (ressonância magnética infratentorial):** 0: negativo, 1: positivo
- **Spinal_Cord_MRI (ressonância magnética da medula espinhal):** 0: negativo, 1: positivo
- **initial_EDSS:**?
- **final_EDSS:**?
- **Group (grupo):** 1: CDMS, 2: não-CDMS

Onde as definições de alguns termos médicos/técnicos são os seguintes:

- **BAEP:** Potencial evocado auditivo de tronco encefálico (PEATE), considerado um método de avaliação objetivo, que permite avaliar desde o nervo auditivo até o tronco encefálico e possibilita a constatação de anormalidades auditivas;
- **VEP:** Potencial Evocado Visual (PEV), usado para diagnosticar déficit visual devido a algum tipo de lesão no nervo óptico;
- **Bandas oligoclonais:** bandas de imunoglobulinas que são observadas quando o soro sanguíneo de um paciente ou líquido cefalorraquidiano (LCR) é analisado. São usados no diagnóstico de várias doenças neurológicas e sanguíneas. Bandas oligoclonais estão presentes no LCR de mais de 95% dos pacientes com esclerose múltipla clinicamente definida;
- **SSEP:** Potenciais evocados somatossensoriais (SSEP), são registrados a partir do sistema nervoso central após a estimulação dos nervos periféricos. ULSSEP (SSEP dos membros superiores), LLSSEP (SSEP dos membros inferiores);
- **EDSS:** Escala Expandida do Estado de Incapacidade (EDSS), um método de quantificar a incapacidade na esclerose múltipla e monitorar as mudanças no nível de incapacidade ao longo do tempo. É amplamente utilizado em ensaios clínicos e na avaliação de pessoas com EM.

A segunda base de dados usada diz respeito à predição do Câncer de Mama, utilizando o Wisconsin Breast Cancer Database, contido de medições de células provenientes de biópsias de mulheres com nódulos mamários anormais.

A descrição das colunas da base é a seguinte:

- **ID:** número de identificação
- **Diagnóstico:** (M = maligno, B = benigno)

O restante das colunas contém características de valor real que são computadas para cada núcleo celular:

- raio (média das distâncias do centro aos pontos do perímetro)
- textura (desvio padrão dos valores da escala de cinza)
- perímetro
- área
- suavidade (variação local nos comprimentos dos raios)
- compacidade (perímetro^2/área - 1,0)
- concavidade (gravidade das porções côncavas do contorno)
- pontos côncavos (número de porções côncavas do contorno)
- simetria
- dimensão fractal ("aproximação da costa" - 1)

A terceira base de dados usada diz respeito à predição da Diabetes. É uma coleção de dados médicos e demográficos de pacientes, juntamente com seu status de diabetes (positivo ou negativo).

Os dados incluem características como:

- idade
- sexo
- índice de massa corporal (IMC)
- hipertensão
- doenças cardíacas
- histórico de tabagismo
- nível de HbA1c
- nível de glicose no sangue.

A quarta base de dados usada diz respeito à análise e predição de Ataques Cardíacos. Os dados incluem as seguintes informações:

- **idade do paciente**

- **sexo do paciente:** 1= masculino, 0= feminino
- **cp (tipo de dor no peito):** 0: angina típica, 1: angina atípica, 2: dor não anginosa, 3: assintomático
- **trtbprs:** pressão arterial em repouso (em mm Hg)
- **chol:** colesterol em mg/dl
- **fbs:** açúcar no sangue em jejum > 120 mg/dl (1: verdadeiro; 0: falso)
- **output:** resultado (0: menor chance de ataque cardíaco; 1: maior chance de ataque cardíaco)

A avaliação de todos os modelos será feita via *cross-validation* fornecida pela biblioteca *sklearn*.

A técnica de validação cruzada é a k-fold, onde o conjunto de dados é dividido em k subconjuntos (chamados de "folds").

Através da função *cross_val_score*, executa-se a validação cruzada de acordo com o número de folders (partições), passado via parâmetro. Na sequência, retorna uma métrica de performance para cada folder de teste. Por padrão, essa métrica é a acurácia, adicionando-se também a F-score.

Faz-se o uso também da função *cross_val_predict*, que permite obter as predições do modelo ao invés de apenas as métricas finais. Neste caso, cada predição será obtida para o conjunto de teste de cada uma das partições.

Em outras palavras, se cv=5, o modelo vai ser treinado para 4 partições e testado em 1, que gera as predições.

Ao final das execuções, os resultados são concatenados e retornados.

Outra classe utilizada é a *StratifiedKFold*, que retorna os índices de cada partição de maneira aleatória e estratifica os dados de acordo com as classes do problema.

Em outras palavras, balanceia a quantidade de amostras de cada classe entre as partições a fim de equilibrar a base de dados.

Outro método utilizado foi a de busca em grade (*grid search*), que visa otimizar os hiperparâmetros dos modelos. Isso envolve a definição de uma "grade" de valores possíveis para cada hiperparâmetro, consistindo em treinar e avaliar o modelo para todas as combinações possíveis de valores na grade, a fim de encontrar a combinação de hiperparâmetros que resulta no melhor desempenho do modelo.

Embora computacionalmente intensiva, a busca em grade auxilia no processo descrito anteriormente.

Após esses procedimentos, é possível obter a acurácia e F-score das partições para cada modelo utilizado.

IV. RESULTADOS

Nesta seção, apresenta-se os resultados de uma análise realizada com base em um estudo sobre diferentes patologias, tendo como objetivo explorar o desempenho de algoritmos de aprendizado de máquina na predição e classificação dessas condições (mencionadas na seção anterior). .Os resultados obtidos são apresentados a seguir:

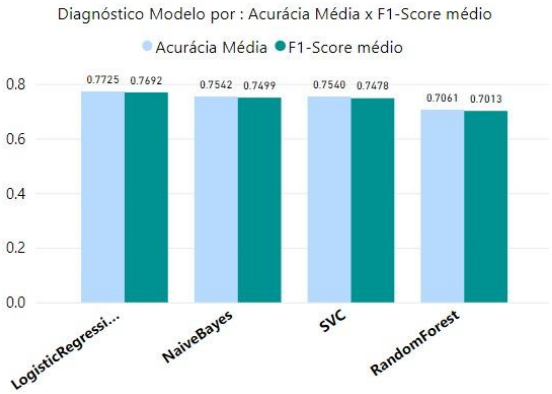

Figura 3: Diagnóstico do modelo a partir da acurácia e F1-Score. Fonte: Autor

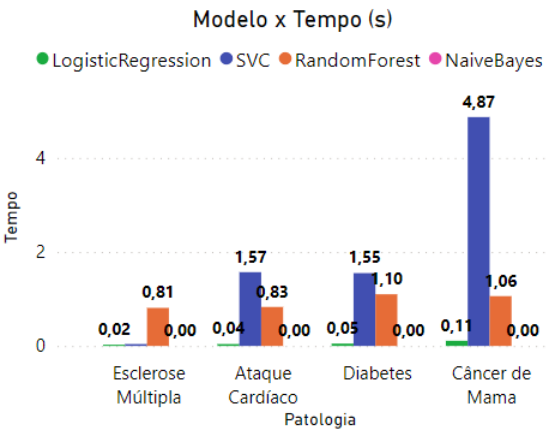

Figura 4: Diagnóstico do modelo a partir do tempo de treinamento. Fonte: Autor

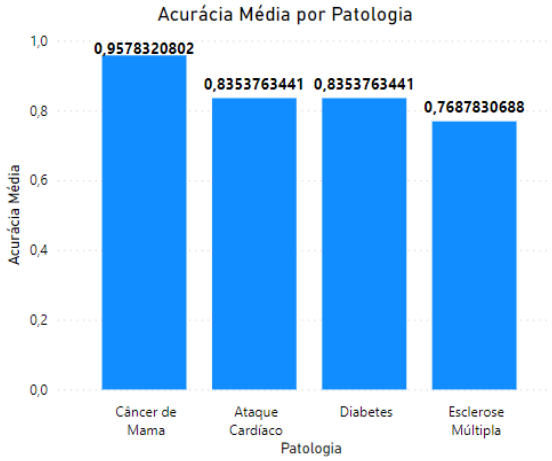

Figura 5: Acurácia média de acordo com a patologia. Fonte: Autor

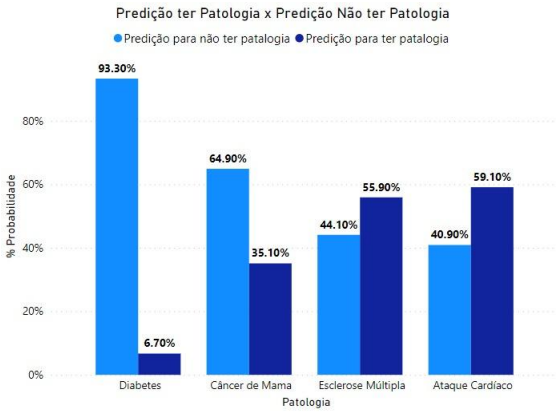

Figura 6: Gráfico da predição de ter ou não a patologia. Fonte: Autoria Própria.

Segundo os resultados obtidos a partir das tabelas fornecidas, utilizando uma base de dados específica e treinando os modelos de forma adequada, foram alcançados os seguintes resultados.

Utilizando o algoritmo de Regressão Logística, obteve-se uma acurácia média de 77,25% e um F1-Score médio de 76,92%.

O modelo SVM alcançou uma acurácia média de 75,40% e um F1-Score médio de 74,78%.

Já o modelo da Floresta Aleatória apresentou uma acurácia média de 70,61% e um F1-Score médio de 70,13%.

Por fim, o modelo Naive Bayes registrou uma acurácia média de 75,42% e um F1-Score médio de 74,99%.

Analisando o tempo de treinamento dos modelos para cada patologia, observou-se que, para a Esclerose Múltipla, a Regressão Logística levou 0,02 segundos, o SVM necessitou de 0,04 segundos, a Floresta Aleatória demandou 0,81 segundos e o Naive Bayes não teve tempo de treinamento registrado (tempo muito baixo e/ou desprezível).

No caso do Câncer de Mama, a Regressão Logística levou 0,11 segundos, o SVM precisou de 4,87 segundos, a Floresta Aleatória exigiu 1,06 segundos e o Naive Bayes novamente não teve tempo de treinamento registrado (tempo muito baixo e/ou desprezível).

Para o Ataque Cardíaco, a Regressão Logística demandou 0,04 segundos, o SVM necessitou de 1,57 segundos, a Floresta Aleatória precisou de 0,83 segundos e o Naive Bayes não teve tempo de treinamento registrado (tempo muito baixo e/ou desprezível).

Por fim, para a Diabetes, a Regressão Logística levou 0,05 segundos, o SVM exigiu 1,55 segundos, a Floresta Aleatória demandou 1,10 segundos e o Naive Bayes não teve tempo de treinamento registrado (tempo muito baixo e/ou desprezível).

Ao analisar a acurácia média de acordo com a patologia, constatou-se que a Esclerose Múltipla apresentou uma acurácia média de 76,88%.

Em relação ao Câncer de Mama registrou uma acurácia média de 95,78%.

Para o Ataque Cardíaco, a acurácia média foi de 83,54%.

Para a Diabetes, a acurácia média foi de 96,20%.

Por fim, considerando o gráfico da probabilidade de ter ou não a patologia, verificou-se que, para a Esclerose Múltipla, houve uma predição de 56% de chances de tê-la e 44% de chances de não ter.

Para o Câncer de Mama, a predição foi de 35% de chances de ter e 65% de chances de não ter.

No caso do Ataque Cardíaco, a predição foi de 59% de chances de ter e 41% de chances de não ter.

Quanto à Diabetes, houve uma predição de 7% de chances de ter e 93% de chances de não ter.

Através da função *VoteClassifier* foi possível obter o melhor resultado de cada partição/predição, trazendo ainda mais informação sobre o aprendizado dos modelos utilizados (figuras 5 e 6).

É válido ressaltar que, com base nos resultados apresentados, não é possível determinar com certeza qual é o melhor modelo, pois isso depende dos critérios específicos de avaliação e das necessidades do problema em questão.

O que pode-se destacar é que tanto o modelo de Regressão Logística quanto o SVM obteve uma acurácia média e um F1-Score médio razoavelmente bons, demonstrando consistência em sua performance. Além disso, o modelo Naive Bayes obteve resultados semelhantes, sugerindo que ambos são opções viáveis.

O que pode explicar o bom desempenho da Regressão Logística junto do SVM é o fato de esse algoritmo ser computacionalmente eficiente, além de ser também eficaz quando a relação entre as variáveis e a classe é aproximadamente linear. Isto é, quando os dados estavam bem representados em um hiperplano de separação linear, a Regressão teve uma boa performance.

Ainda em relação aos dados serem linearmente separáveis ou não, a busca em grade permitiu que fossem feitas combinações com dois tipos diferentes do SVM: linear e RBF (*Radial Basis Function*), possibilitando uma otimização ainda maior.

Outro ponto relevante em relação a ambos é que mesmo semelhantes em relação ao F1-Score e acurácia, o SVM conseguiu predizer melhor quem tem ou não as doenças na maioria dos casos.

No entanto, é importante considerar outros fatores além da acurácia, como o tempo de treinamento, as características específicas do problema e a interpretabilidade do modelo. Portanto, realizar uma análise mais aprofundada (ou fazer uma revisão), levando em conta esses aspectos antes de tomar uma decisão definitiva

sobre qual é o melhor modelo para a tarefa em questão, pode ser ideal para que se possa ter resultados ainda melhores.

V. FINALIZAÇÃO

O que será entregue no final: o algoritmo de Naive Bayes é uma técnica popular e poderosa de aprendizagem de máquina, amplamente utilizada para problemas de classificação. Uma das principais vantagens do algoritmo de Naive Bayes é sua simplicidade e eficiência computacional. Ele se baseia no Teorema de Bayes e na suposição de independência condicional entre os atributos, o que permite uma implementação direta e rápida. Além disso, o Naive Bayes lida bem com conjuntos de dados de alta dimensionalidade e é menos suscetível a overfitting em comparação com outros algoritmos mais complexos.

Ao utilizar o algoritmo de Naive Bayes para uma tarefa de classificação de plantas Íris, o resultado final tende a ser um modelo treinado capaz de prever a classe correta das plantas com base em seus atributos. Especificamente, ao aplicar o Naive Bayes às medidas dos atributos das plantas Íris, como comprimento e largura das sépalas e pétalas, o modelo aprenderá a estimar a probabilidade de pertencer a cada uma das classes de Íris: Setosa, Versicolor e Virginica.

Em comparação com outros modelos probabilísticos, o Naive Bayes oferece várias vantagens, como já mencionado acima. No entanto, é importante ressaltar que o Naive Bayes faz uma suposição forte de independência condicional entre os atributos, o que nem sempre é realista em muitos conjuntos de dados. Isso pode levar a uma diminuição da precisão do modelo quando a dependência entre os atributos é relevante. Portanto, é mais adequado para problemas em que a independência condicional é uma suposição razoável ou quando a precisão não é a principal preocupação.

Em contraste, outros modelos probabilísticos mais complexos, podem capturar dependências mais sofisticadas entre os atributos. As Redes Bayesianas, por exemplo, permitem modelar relações complexas entre as variáveis e incorporar conhecimentos prévios sobre o problema em questão. No entanto, o treinamento e a inferência em Redes Bayesianas podem ser computacionalmente mais intensivos e requerem mais dados para estimar os parâmetros.

Portanto, ao final da tarefa de classificação de plantas Íris utilizando o algoritmo de Naive Bayes, será entregue um modelo treinado capaz de prever a classe de novas plantas com base em seus atributos. Embora o Naive Bayes seja simples e eficiente, sua precisão pode ser limitada em casos de dependências complexas entre os atributos. Nesses casos, modelos probabilísticos mais avançados podem ser considerados para obter resultados mais precisos, mas com maior complexidade computacional e necessidade de mais dados.

É esse contraste que pretende-se evidenciar aplicando outras técnicas e modelos em novas bases de dados e comparando os resultados obtidos.

REFERÊNCIAS

[1] CATARINO, Francisco Manuel Inácio Ferreira. Segmentação da íris em imagens com ruído. 2009. Tese (Mestre em Engenharia Informática) - Universidade da Beira Interior, [S. l.], 2009. Disponível em: https://ubibliorum.ubi.pt/handle/10400.6/3731. Acesso em: 4 junho 2023.

[2] MOURA, Gabriel Barreto. REDES PROBABILÍSTICAS FUZZY NAÏVE BAYES. 2016. Dissertação (Pós-Graduação em Ciência da Computação) - Universidade Federal de Santa Catarina, [S. l.], 2016. Disponível em: https://repositorio.ufsc.br/xmlui/bitstream/handle/123456789/167777/341498.pdf?sequence=1&isAllowed=y. Acesso em: 4 junho 2023.

[3] SOUZA, Rafael Castro de; NETO, Francisco Milton Mendes. SOFTWARE CLASSIFICADOR DE ESPÉCIE DE PLANTAS UTILIZANDO O MODELO PROBABILÍSTICO BAYESIANO. 2022. Dissertação (Programa de Pós-Graduação em Ciência da Computação) - Universidade Federal do Piauí, [S. l.], 2022. Disponível em: https://www.editorainvivo.com/_files/ugd/08fcde_ead9d13c6e4c4127800cfec2596d1339.pdf#page=14. Acesso em: 4 junho 2023.

[4] Uma implementação do algoritmo Naïve Bayes para classificação de texto. Disponível em: <https://turing.pro.br/anais/ERBD-2013/artigos/aplicacoes/111388.pdf>. Acesso em: 4 junho 2023.

[5] NAIVE BAYES: COMO FUNCIONA ESSE ALGORITMO DE CLASSIFICAÇÃO. Disponível em: <https://blog.somostera.com/data-science/naive-bayes#:~:text=Quando%20usar%20Naive%20Bayes%3F,processamento%20de%20linguagem%20natural%2C%20portanto.>. Acesso em: 4 junho 2023

[6] Algoritmo de Classificação Naive Bayes Disponível em: <https://www.organicadigital.com/blog/algoritmo-de-classificacao-naive-bayes/#:~:text=O%20algoritmo%20de%20Naive%20Bayes,dado%20que%20tem%20a%20doen%C3%A7a%E2%80%9D.>. Acesso em: 4 junho 2023

[7] GNB_Iris. Disponível em:

<https://colab.research.google.com/drive/1pL2ZWTrXkiMWxAMosbVreHeGBWlvwalA?usp=sharing>

[8] Han, J. and Kamber, M. (2006). Data Mining: Concepts and Techniques. Morgan Kaufmann Publishers, 2nd ed.

[9] FAWCETT, T.; PROVOST, F. Data Science para negócios: O que você precisa saber sobre mineração de dados e pensamento analítico de dados. Alta Books Editora, 2018.

[10] Saba Abdul-baqi Salman, Al-Hakam Ayad Salih, Ahmed Hussein Ali, Mohammad Khamees Khaleel, Mostafa Abdulghfoor Mohammed. A NEW MODEL FOR IRIS CLASSIFICATION BASED ON NAIVE BAYES GRID PARAMETERS OPTIMIZATION. Disponível em: https://www.academia.edu/download/57172097/9229-27310-1-PB.pdf. Acesso em 6 de junho de 2023

[11] K.Vembandasamy, R. Sasipriya and E. Deepa. HEART DISEASES DETECTION USING NAIVE BAYES ALGORITHM. Disponivel em: https://scholar.archive.org/work/aeqqx3eltjg2vazgnqpfvm4x4u/access/wayback/http://www.ijiset.com/vol2/v2s9/IJISET_V2_I9_54.pdf. Acesso em 6 de junho de 2023.

[12] Masud Karim, Rashedur M Rahman. DECISION TREE AND NAIVE BAYES ALGORITHM FOR CLASSIFICATION AND GENERATION OF ACTIONABLE KNOWLEDGE FOR DIRECT MARKETING. Disponível em: https://www.scirp.org/html/6-9301587_30463.htm. Acesso em 6 de junho de 2023.

[13] Jin Huang, Jingjing Lu, Charles X. Ling. COMPARING NAIVE BAYES, DECISION TREES, AND SVM WITH AUC AND ACCURACY. Disponível em: https://www.computer.org/csdl/proceedings-article/icdm/2003/19780553/12OmNAkWvjU. Acesso em 6 de junho de 2023

[14] PACHECO, André. Introdução ao Scikit-learn - Parte 3: avaliando a qualidade do modelo via cross-validation. [S. l.], 3 jul. 2021. Disponível em: http://computacaointeligente.com.br/outros/intro-sklearn-part-3/. Acesso em: 11 jun. 2023.

[15] PACHECO, André. Avaliação de modelos, cross-validation e data leakage. [S. l.], 25 jun. 2021. Disponível em: http://computacaointeligente.com.br/conceitos/avaliando-performance-cross-validation/. Acesso em: 11 jun. 2023.

[16] LUCAS, Luiz Campos de Sá. Árvores, Florestas e sua função como preditores: uma aplicação na avaliação do grau de maturidade de empresas. Revista PMKT, [s. l.], p. 6-11, 17 mar. 2011.

[17] O que é regressão logística?. [S. l.], 2021. Disponível em: https://aws.amazon.com/pt/what-is/logistic-regression/#:~:text=A%20regress%C3%A3o%20log%C3%ADstica%20%C3%A9%20uma,resultados%2C%20como%20sim%20ou%20n%C3%A3o. Acesso em: 20 jun. 2023.

[18] Regressão logística. [S. l.], 2021. Disponível em: https://www.ibm.com/docs/pt-br/spss-statistics/saas?topic=regression-logistic. Acesso em: 20 jun. 2023.

[19] GONZALEZ, Leandro de Azevedo. Regressão Logística e suas Aplicações. 2018. Monografia (Bacharel em Ciência da Computação) - Universidade Federal do Maranhão, [S. l.], 2018. Disponível em: https://monografias.ufma.br/jspui/bitstream/123456789/3572/1/LEANDRO-GONZALEZ.pdf. Acesso em: 20 jun. 2023.

[20] GONÇALVES, André Ricardo. Máquina de Vetores Suporte. [S. l.], 200-. Disponível em: https://andreric.github.io/files/pdfs/svm.pdf. Acesso em: 20 jun. 2023.

[21] JUNIOR, Geraldo Braz. Máquinas de Vetores Suporte. [S. l.], 201-. Disponível em: https://nca.ufma.br/~geraldo/vc/12.SVM.pdf. Acesso em: 20 jun. 2023.

[22] JUNIOR, Gilson Medeiros de Oliveira. Máquina de Vetores Suporte: estudo e análise de parâmetros para otimização de resultado. 2010. Monografia (Bacharel em Ciência da Computação) - Universidade Federal de Pernambuco, [S. l.], 2010. Disponível em: https://www.cin.ufpe.br/~tg/2010-2/gmoj.pdf. Acesso em: 20 jun. 2023.

[23] SILVA, Josenildo Costa da. Aprendendo em uma Floresta Aleatória. [S. l.], 12 mar. 2018. Disponível em: https://medium.com/machina-sapiens/o-algoritmo-da-floresta-aleat%C3%B3ria-3545f6babdf8. Acesso em: 20 jun. 2023.

[24] SILVA, Fernando Trentino. Aplicação do Algoritmo Ensemble de Floresta Aleatória para a Classificação de Clientes Adimplentes e Inadimplentes. 2021. Trabalho de Conclusão de Curso (Bacharel em Engenharia de Computação) -

Universidade Federal de Mato Grosso, [S. l.], 2021. Disponível em: https://bdm.ufmt.br/bitstream/1/1969/1/TCC%20%20Fernanda%20Trentino%20Silva_organized.pdf. Acesso em: 20 jun. 2023.

[25] GONÇALVES, André Ricardo. Redes Bayesianas. [S. l.], 200-. Disponível em: https://andreric.github.io/files/pdfs/bayesianas.pdf. Acesso em: 20 jun. 2023.

[26] JAYNES, E.T; Probability Theory: The Logic of Science. Cambridge, Cambridge University Press, 2003.

[27] ANDERSON, D. R.; SWEENEY, D. J.; WILLIANS, T. A. Statistics for Business and Economics. Thompson, 2005.

[28] PAULINO, C. D.; TURKMAN, M. A. A.; MURTEIRA, B. Estatística Bayesiana. Lisboa:Fundação Calouste Gulbenkian, 2003.

[29] CRAMÉR, H. Elementos da Teoria da Probabilidade e algumas de suas aplicações. São Paulo:Mestre Jou, 1955

[30] MARQUES, R. L.; DUTRA, I. Redes Bayesianas: o que são, para que servem, algoritmos e exemplos de aplicações. Rio de Janeiro: [s.n.], 2008. Disponível em: <https://www.cos.ufrj.br~ines/courses/cos740/leila/cos740/Bayesianas.pdf>.

[31] Russel, J. Stuart & Norvig, Peter. “Artifical Intelligence: A modern Approach”.Prentice Hall 415-429, 436-457,1995.

[32] ROSSATTO, Felipe Copceski; DAMMANN, Julia; KAMPHORST, Eliane Miotto; KAMPHORST, Carmo Henrique; DONADEL, Ana Paula do Prado. Teorema de Bayes: estudo e aplicação. Salão do Conhecimento, [s. l.], 2016. Disponível em: https://publicacoeseventos.unijui.edu.br/index.php/salaoconhecimento/article/view/6858/5625. Acesso em: 20 jun. 2023.

[33] MONICO, João Francisco Galera; PÓZ, Aluir Porfírio Dal; GALO, Maurício; SANTOS, Marcelo Carvalho dos; OLIVEIRA, Leonardo Castro de. Acurácia e Precisão: revendo os conceitos de forma acurada. Boletim de Ciências Geodésicas, [S. l.], v. 15, n. 3, p. 469-483, 15 jul. 2009. Disponível em: https://www.redalyc.org/pdf/3939/393937709010.pdf. Acesso em: 20 jun. 2023.

[34] LEAL, Renato do Santos. Métricas Comuns em Machine Learning: como analisar a qualidade de chat bots inteligentes — métricas. [S. l.], 22 maio 2017. Disponível em: https://medium.com/as-m%C3%A1quinas-que-pensam/m%C3%A9tricas-comuns-em-machine-learning-como-analisar-a-qualidade-de-chat-bots-inteligentes-m%C3%A9tricas-1ba580d7cc96. Acesso em: 20 jun. 2023.

[35] MARIANO, Diego. Métricas de avaliação em machine learning. [S. l.], 25 abr. 2021. Disponível em: https://diegomariano.com/metricas-de-avaliacao-em-machine-learning/. Acesso em: 20 jun. 2023.

[36] BERTRAN, Erica. O que é Precisão e Revocação. [S. l.], 22 nov. 2020. Disponível em: https://medium.com/computando-arte/o-que-%C3%A9-precis%C3%A3o-e-revoca%C3%A7%C3%A3o-b0b991b67cde. Acesso em: 20 jun. 2023.