Universidade Federal do Rio de Janeiro
Escola Politécnica
MBA em Big Data, Business Intelligence e Business Analytics
(MB3B)

A IMPORTÂNCIA DE UTILIZAR AS MÉTRICAS ADEQUADAS DE AVALIAÇÃO DE PERFORMANCE DE MODELOS PREDITIVOS DE MACHINE LEARNING DE ACORDO COM O PROBLEMA A SER ANALISADO

Autor:

Electronically signed by:
Wagner Luiz Lobo Ferreira
Wagner Luiz Lobo Ferreira
wagner.lobobr@gmail.com
20/10/2023 11:37:27
ID: 8GUYDQOJQGI-GEZTINBXHE2A-PKEVJONCAXIJW

Wagner Luiz Lobo Ferreira

Orientador:

Electronically signed by:
Manoel Villas Boas Júnior, M.Sc. Vice-Coordenador de Pós-Graduação Latu-Senso e Extensão ITLAB - Escola Politécnica Universidade Federal do Rio de Janeiro SIAPE: 0036511736
Manoel Villas Boas Junior
mvbjunior@poli.ufrj.br
20/10/2023 11:40:21
ID: 7GUYDQOJQGI-GEZTINBXHE2A-RMNN2LZP5OYTY

Manoel Villas Boas Junior, M. Sc.

Examinador:

Electronically signed by:
Norberto Ribeiro Bellas
Norberto Ribeiro Bellas
norbertobellas@gmail.com
22/10/2023 10:31:01
ID: CGUYDQOJQGI-GEZTINBXHE2A-FDH4Y7CWTQN4E

Norberto Ribeiro Bellas, M. Sc.

Examinador:

Electronically signed by:
Vinicius Drumond Gonzaga
viniciusdgonzaga@gmail.com
22/10/2023 15:56:20
ID: 9GUYDQOJQGI-GEZTINBXHE2A-TXQU34EF544RA

Vinicius Drumond Gonzaga, M. Sc.

Examinador:

Electronically signed by:
Vinicius Teixeira do Nascimento
vinicius.tn@yahoo.com
22/10/2023 17:25:46
ID: 8GUYDQOJQGI-GEZTINBXHE2A-PKEVJONCAXIJW

Vinicius Teixeira do Nascimento, M. Sc.

Rio de Janeiro
Junho de 2023

Declaração de Autoria e de Direitos

Eu, **Wagner Luiz Lobo Ferreira**, inscrito no CPF/ME sob o nº 108.454.587-01, autor da monografia ***AUTOMATIZAÇÃO DO PROCESSO DE BLOQUEIO DE ENTREGAS EM COMPRAS ONLINE FRAUDULENTAS***, subscrevo para os devidos fins, as seguintes informações:

1. O autor declara que o trabalho apresentado na defesa da monografia do curso de Pós-Graduação, Especialização MBA em Big Data, Business Intelligence e Business Analytics da Escola Politécnica da UFRJ é de sua autoria, sendo original em forma e conteúdo.

2. Excetuam-se do item 1 eventuais transcrições de texto, figuras, tabelas, conceitos e ideias, que identifiquem claramente a fonte original, explicitando as autorizações obtidas dos respectivos proprietários, quando necessárias.

3. O autor permite que a UFRJ, por um prazo indeterminado, efetue em qualquer mídia de divulgação, a publicação do trabalho acadêmico em sua totalidade, ou em parte. Essa autorização não envolve ônus de qualquer natureza à UFRJ, ou aos seus representantes.

4. O autor declara, ainda, ter a capacidade jurídica para a prática do presente ato, assim como ter conhecimento do teor da presente Declaração, estando ciente das sanções e punições legais, no que tange a cópia parcial, ou total, de obra intelectual, o que se configura como violação do direito autoral previsto no Código Penal Brasileiro no art.184 e art.299, bem como na Lei 9.610.

5. O autor é o único responsável pelo conteúdo apresentado nos trabalhos acadêmicos publicados, não cabendo à UFRJ, aos seus representantes, ou ao(s) orientador(es), qualquer responsabilização/ indenização nesse sentido.

6. Por ser verdade, firmo a presente declaração.

Rio de Janeiro, 2023-10-20 14:37:27.

Electronically signed by:
Wagner Luiz Lobo Ferreira
Wagner Luiz Lobo Ferreira
wagner.lobobr@gmail.com
20/10/2023 11:37:27
ID: 8GUYDQOJQGI-GEZTINBXHE2A-PKEVJONCAXIJW

Wagner Luiz Lobo Ferreira

UNIVERSIDADE FEDERAL DO RIO DE JANEIRO

Av. Athos da Silveira, 149 - Centro de Tecnologia, Bloco H, sala - 212,
Cidade Universitária Rio de Janeiro – RJ - CEP 21949-900.

DEDICATÓRIA

Dedico esse TCC primeiramente à minha mãe, que sempre esteve ao meu lado apoiando as minhas escolhas. Depois aos professores, que me deram o suporte necessário e o arcabouço de conhecimento útil que permitiu com que eu seguisse adiante nessa que é mais uma etapa importante que precede o mestrado acadêmico que eu planejo realizar em breve.

RESUMO

A ciência de dados tem se destacado como uma ferramenta poderosa para análises estatísticas e identificação de padrões. No entanto, é fundamental ir além da criação de modelos preditivos e avaliá-los para garantir sua eficácia e assertividade nas previsões. Este projeto tem como foco a etapa de avaliação de desempenho de modelos preditivos, uma fase essencial para determinar se o modelo é capaz de fornecer insights relevantes com base em dados históricos. Nessa pesquisa, serão discutidas as métricas comumente utilizadas para avaliar e definir a qualidade de um modelo preditivo, tais como acurácia, precisão, *recall*, *F1 Score* e curva *AUC*. O leitor será informado sobre as situações em que o uso de cada métrica é recomendado e em quais contextos eles podem ser aplicados. Também serão apresentados vários modelos preditivos distintos - cada um com suas especificidades - abordando de forma prática as informações discutidas ao longo do projeto. É importante ressaltar que algumas etapas cruciais do processo de análise de dados, como coleta de dados e EDA (*Exploratory Data Analysis*), serão abordadas de maneira mais sucinta neste trabalho, devido ao enfoque na avaliação das métricas dos modelos preditivos. Espera-se como resultado a criação de três modelos preditivos, os quais permitirão ao leitor, durante a conclusão deste projeto, compreender experimentalmente a importância da etapa de avaliação em um projeto de ciência de dados. Através das análises realizadas, será possível constatar a relevância prática dessa etapa e como ela contribui para a obtenção de insights valiosos a partir dos dados analisados.

Palavras-chave: Aprendizado de Máquinas, Métricas de Classificação, Métricas de Regressão, Decisão Orientada a Dados

ABSTRACT

Data science has emerged as a powerful tool for statistical analysis and pattern identification. However, it is essential to go beyond the creation of predictive models and evaluate them to ensure their effectiveness and assertiveness in predictions. This project focuses on the performance evaluation stage of predictive models, an essential phase to determine whether the model is capable of providing relevant insights based on historical data. In this research, the metrics commonly used to evaluate and define the quality of a predictive model will be discussed, such as accuracy, precision, recall, F1 Score and AUC curve. The reader will be informed about the situations in which the use of each metric is recommended and in which contexts they can be applied. Several different predictive models will also be presented - each with its specificities - approaching in a practical way the information discussed throughout the project. It is important to point out that some crucial stages of the data analysis process, such as data collection and EDA (Exploratory Data Analysis), will be addressed more succinctly in this work, due to the focus on evaluating the metrics of predictive models. It is expected as a result the creation of three predictive models, which will allow the reader, during the conclusion of this project, to experimentally understand the importance of the evaluation stage in a data science project. Through the analyzes carried out, it will be possible to verify the practical relevance of this step and how it contributes to obtaining valuable insights from the analyzed data.

Keywords: Machine Learning, Ranking Metrics, Regression Metrics, Data Driven Decision

SIGLAS

AUC	Area Under Curve
EDA	Exploratory Data Analysis
IA	Inteligência Artificial
PCA	Principal Component Analysis
ROC	Receiver Operating Characteristic
UFRJ	Universidade Federal do Rio de Janeiro

Lista de Figuras

Lista de Tabelas

Lista de Equações

Sumário

CAPÍTULO 1

Introdução

Desde que as atividades humanas passaram a ser registradas pela humanidade, dados começaram a ser gerados e estes eram utilizados como insumos em análises estatísticas diversas, desde as mais simples até as mais sofisticadas. Essas análises, que serviam a múltiplos propósitos, impulsionaram e guiaram o crescimento de diversos setores importantes da sociedade: na agricultura, era possível otimizar a produção de alimentos de acordo com a demanda local, na medicina era viável detectar doenças e produzir remédios e vacinas contra a mesma. Entre muitas outras aplicações.

Conforme dito acima, apesar de já ser conhecido o quão importante é a análise de dados em diversas áreas, sendo uma estratégia frutífera desde muito tempo, existiam muitos problemas em relação à capacidade de armazenar e processar esses dados. Como mostrado na figura 1.1, a capacidade de armazenamento de dados era bastante limitada e era feito localmente em governos e empresas. Afinal, o espaço de armazenamento era um grande problema, pois era algo muito custoso. Não apenas o armazenamento de dados, mas o processamento desses dados era algo inconcebível a algumas décadas atrás. Entretanto, é válido ressaltar que no passado, a geração de dados não era tão grande como nos dias atuais, devido às limitações tecnológicas da época.

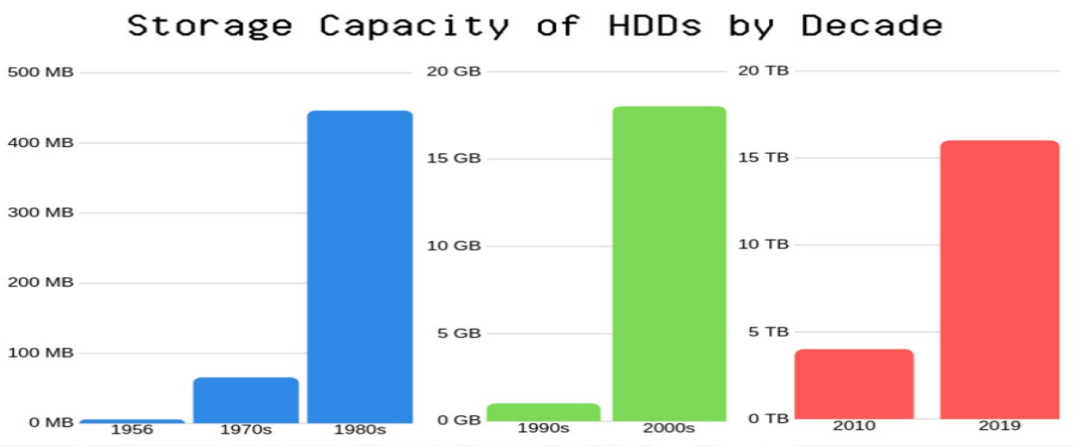

Figura 1.1 - Evolução da capacidade de armazenamento de dados
Fonte: SECURE DATA RECOVERY, 2023

Hoje, a Estatística continua sendo utilizada para os mesmos propósitos - dentre muitos outros - mas a proporção de geração de dados é completamente diferente.

A geração de dados está crescendo de forma exponencial nos últimos tempos. Segundo a revista Exame (https://exame.com/carreira/dados-uso-favor/), em 2020 foram produzidos mais de 40 trilhões de *gigabytes* de dados no mundo. E as fontes de dados são as mais diversas possíveis: São fotos, vídeos, áudios, textos, e estes são compartilhados e transmitidos de maneira contínua entre usuários comuns ou cientistas, analistas de dados e afins. Como desde muito tempo já está sendo gerado muito mais dados do que se é possível tratar e processar, a utilização de recursos computacionais tornou-se uma ferramenta imprescindível em análises estatísticas – e científicas de uma maneira geral. E com a constante evolução da tecnologia e, consequentemente, dos computadores utilizados no dia a dia (*desktops, notebooks*, celulares, *tablets*), além de cenários favoráveis como a Lei de Moore (mostrado na figura 1.2), surgiram novas ferramentas capazes de analisar todos esses dados de maneira mais eficiente e mais rápida, tanto por empresas, governos e mais recentemente, por usuários comuns. Uma das ferramentas mais utilizadas atualmente para análises estatísticas é o *Machine Learning*.

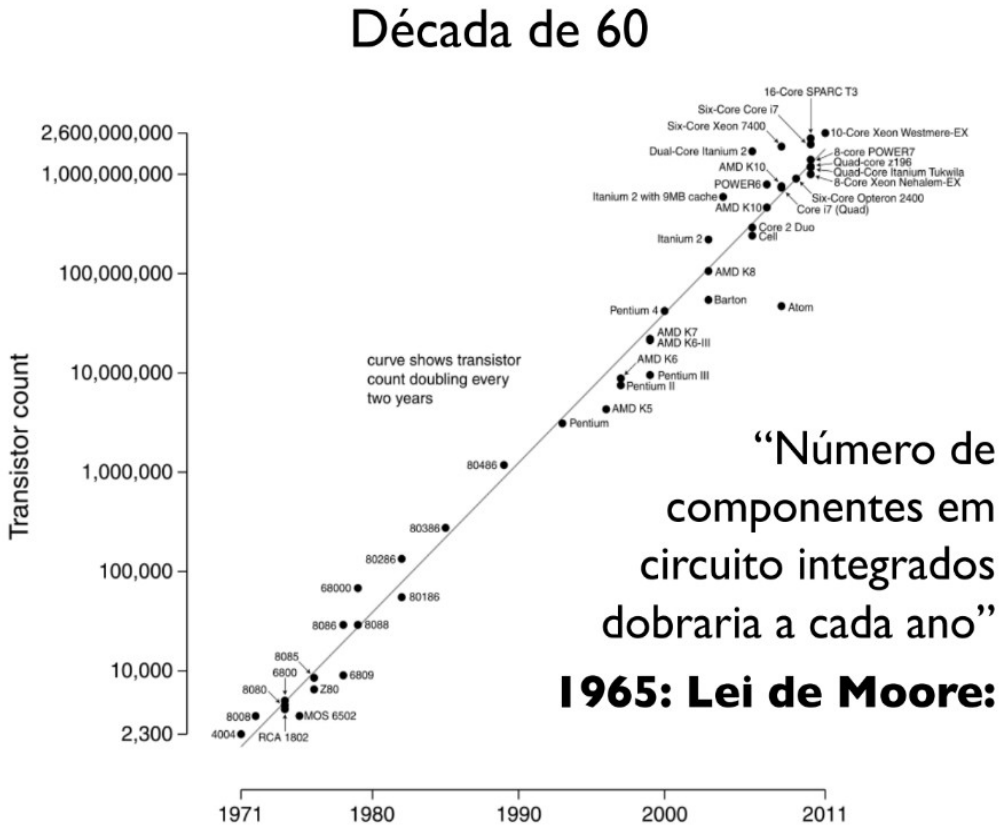

Figura 1.2 - Evolução do poder computacional segundo a Lei de *Moore*
Fonte: Edson Borin, 2015

Durante muito tempo, extrair informações valiosas de uma quantidade colossal de dados era uma árdua tarefa porque, como dito anteriormente, a quantidade de dados gerados crescia, mas a uma taxa muito maior do que era possível processá-los na mesma velocidade para encontrar alguma informação relevante. Modelos estatísticos bastante conhecidos *como naive bayes*, algoritmos de árvore, regressão linear entre outros já haviam sido publicados em *papers* e apresentados à comunidade científica ou já eram conhecidos pela comunidade estatística, mas a aplicação desses modelos em alguns tipos de problemas era tecnologicamente inviável. Foi nesse contexto que - utilizando-se desses mesmos modelos estatísticos e aliados à crescente capacidade de poder computacional - *Machine Learning* surgiu. Entretanto, como a figura 1.3 sugere, Machine Learning é basicamente uma Estatística Aplicada com recursos computacionais adequados.

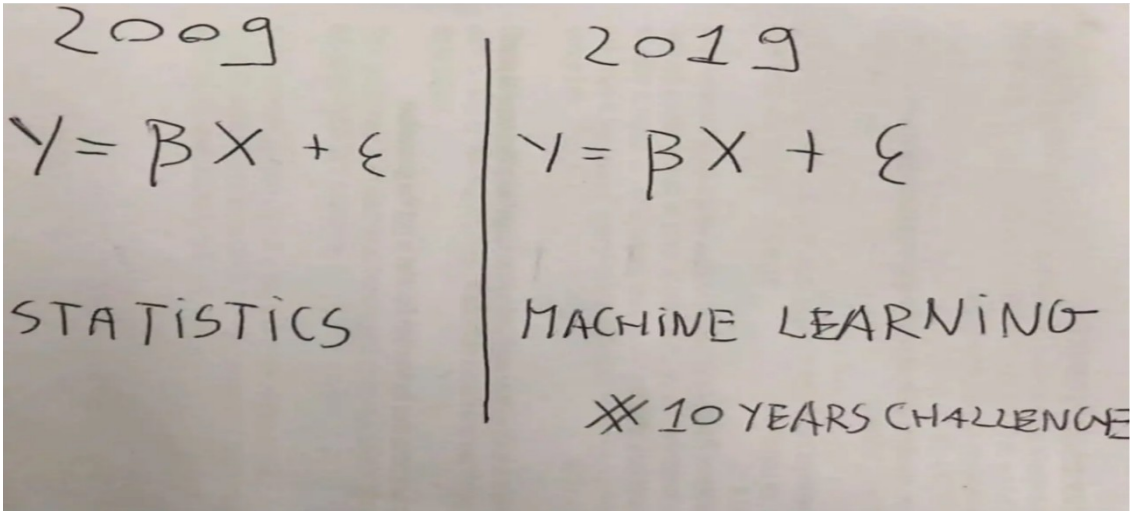

Figura 1.3 - *Machine Learning* como Estatística aplicada
Fonte: Matthew Stewart, 2019

Machine Learning nasceu na década de 60 e, como mostrado na figura 1.4, é um braço da Inteligência Artificial. Apesar do nome pomposo e da fama de ser capaz de prever o futuro, a estratégia nada mais é do que uma técnica de reconhecimento de padrões, que cresceu bastante em popularidade nos últimos tempos. A capacidade de detectar inferências e encontrar correlações em múltiplas variáveis (de maneira relativamente rápida) é uma solução bastante apreciada em inúmeras atividades e ramos de atuação, especialmente no mercado de trabalho. A área financeira, de saúde, engenharia, comércio, educação, entre muitas outras, são beneficiadas pelo crescimento do poder analítico que *Machine Learning*, através da utilização de modelos estatísticos, é capaz de prover.

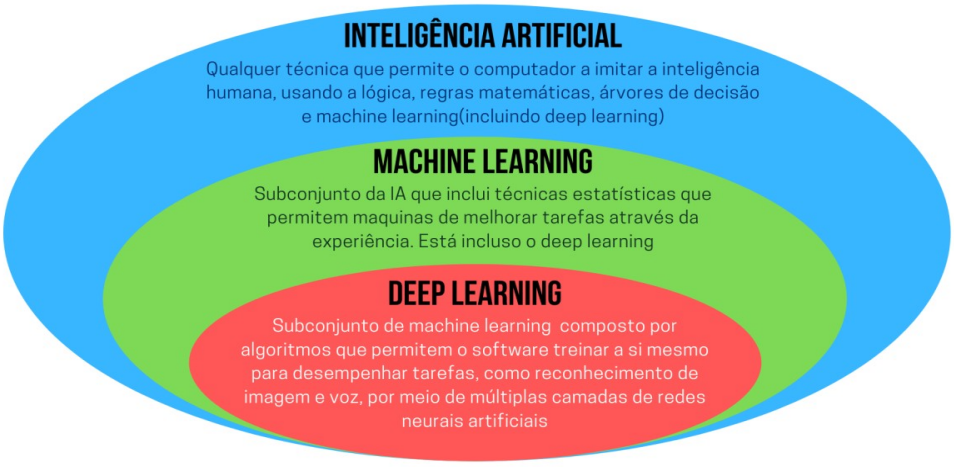

Figura 1.4 - *Machine Learning, Deep Learning*
Fonte: Alberis Castro, 2020

1.1 - Objetivo Geral

Esse presente trabalho tem como objetivo geral comparar as diferentes maneiras de se avaliar a performance de um modelo preditivo de algoritmos supervisionados.

1.2 - Objetivo específico

Muitos iniciantes na área de dados começam seus estudos com modelos preditivos de classificação e, em virtude disso, muitos acreditam que calcular a acurácia – que é a métrica que define o quão assertivo é o modelo em detectar padrões – é o suficiente. Porém, um cientista de dados com mais experiência sabe que somente isso nem sempre é o suficiente. O estudo presente neste trabalho tem como objetivo principal mostrar outras diferentes maneiras de se medir a eficiência de um modelo preditivo de *Machine Learning*, bem como mostrar diferentes situações em que um modelo preditivo possa ser utilizado e aprimorado caso o mesmo performe mal na etapa de testes.

1.3 – Delimitação

Esse projeto foca nas métricas de avaliação de modelos Supervisionados. Além do mais, toda a parte de análise exploratória dos dados, visualização dos gráficos e afins, que são as etapas iniciais de um projeto de Ciência de Dados, serão deixados a cargo do leitor.

1.4 – Metodologia

Para a realização dessa pesquisa foi utilizado pesquisa bibliográfica em inúmeros livros da área de Ciência de Dados, bem como pontuações diversas de sites como referência para muitas explicações explicitadas ao longo do texto. Além do mais, a experiência do autor, que é um cientista de dados com alguns anos de experiência, foi levada em conta. Dado que inúmeros problemas aqui apresentados já foram vistos no passado. Além do mais, para efeitos de demonstração, foram escolhidos 3 tipos de problemas de aprendizado supervisionado, que embora sejam simples, exemplificam bem diferentes maneiras de se avaliar um modelo preditivo.

1.5 – Descrição

No capítulo 2 será feito uma fundamentação teórica onde será explicado o que são modelos supervisionados e não supervisionados, bem como a exemplificação dos

alguns dos modelos mais utilizados no mercado. No capítulo 3, o foco será os modelos supervisionados. Serão explicados de maneira detalhada, os principais métodos utilizados para validação dos algoritmos preditivos apresentados no capítulo 2. No capítulo 4, serão apresentadas algumas implementações técnicas de tudo que foi explicitado nos capítulos anteriores. O capítulo 5, será a conclusão da pesquisa feita e comentários sobre as métricas de avaliação discutidas até então.

CAPÍTULO 2

Conceituação Teórica

2.1 - Tipos de Aprendizagem de Máquinas

Nos dias atuais, os variados problemas que *Machine Learning* é capaz de resolver encaixam-se em categorias bem diferentes. Em alguns problemas o objetivo é apenas classificar uma determinada transação como fraudulenta ou não fraudulenta, em outro tipo de problema, o objetivo é prever uma variável contínua, como o preço de uma casa. Em alguns casos, é necessário apenas agrupar elementos semelhantes entre si. Entre os tipos de aprendizagem de máquina, existem: Aprendizado Supervisionado, Aprendizado Não Supervisionado e Aprendizado por Reforço.

2.1.1 - Aprendizado Supervisionado

Modelos do tipo de aprendizado supervisionado (figura 2.1) fazem parte de uma categoria de algoritmos preditivos onde a máquina aprende e detecta padrões a partir de base de dados rotuladas, ou seja, que possuem as classes (modelos de Classificação) ou os valores contínuos (modelos de Regressão) explicitamente mostrados na variável alvo durante o treinamento.

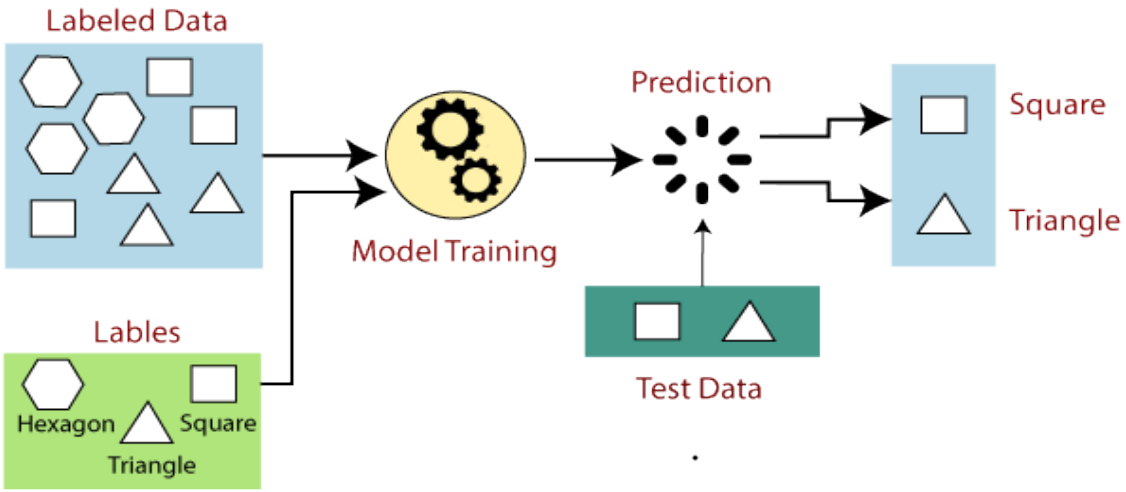

Figura 2.1 - Modelo Supervisionado
Fonte: Sonoo Jaiswal, 2021

Como dito anteriormente, durante o treinamento de modelos supervisionados é possível identificar cada rótulo presente na variável alvo da qual se quer aprender padrões. Durante o treinamento o modelo preditivo chega a conclusões baseados nas informações contidas nas variáveis independentes e compara essas informações com a variável dependente (variável alvo). Após esse treinamento, o modelo utiliza os padrões detectados durante o mesmo para tentar prever os rótulos do dataset de teste, sem olhar para a variável alvo do mesmo.

Durante a fase de teste e com dados nunca antes vistos pelo modelo, são realizadas comparações entre os rótulos previstos pelo modelo preditivo e os rótulos da variável alvo que tinha sido previamente separada do *dataset* de teste. É nessa fase de teste que será possível avaliar a performance do modelo preditivo criado e conferir o quão assertivo ele está, através de métricas que serão explicitadas em capítulos futuros.

Existem 2 tipos de algoritmos supervisionados: Classificação e Regressão.

2.1.1.1 - Algoritmos de Classificação

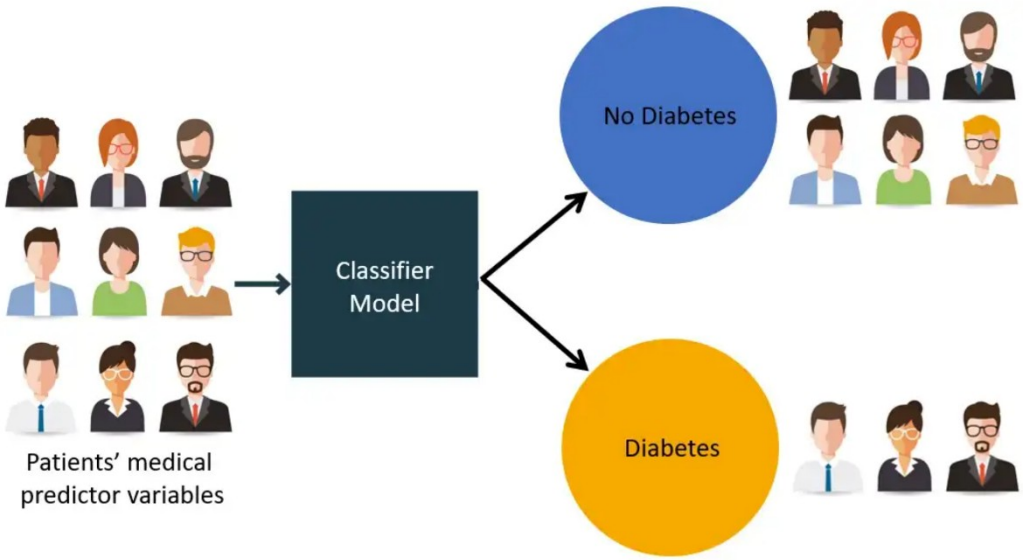

Figura 2.2 - Modelo de Classificação
Fonte: Iprathore, 2020

A premissa de um modelo de classificação é simples: um algoritmo classificador é usado para detectar padrões em classes da variável alvo discretas. O modelo visa categorizar ou classificar os rótulos de acordo com os parâmetros

encontrados nas variáveis independentes. Tal qual a imagem acima mostra, qualificando as pessoas entre pessoas com “diabetes” ou pessoas “sem diabetes” e essa qualificação é feita com base em registros médicos passados, assim como mostrado na figura 2.2. É importante frisar que essa categoria é sempre discreta, ou seja, não existe meia diabete ou meia fraude. É sempre um valor discreto, embora nem sempre binário.

Seguem abaixo alguns exemplos de algoritmos de Classificação populares. Existem muitos outros algoritmos de classificação, mas explicar todos eles fogem do escopo desta pesquisa.

2.1.1.1.1 - *Decision Tree*

Uma árvore de decisão é um tipo de algoritmo utilizado em modelos supervisionados de *Machine Learning* onde em cada nível da árvore, ela divide-se em dois outros ramos e uma decisão é feita, respondendo alguma questão. Essa decisão, combinado às outras respostas feitas nos níveis anteriores, levam a alguma conclusão que é mostrada nas folhas da árvore. Os motivos pelos quais cada opção em cada nível é feita e como os ramos são divididos são baseados em ganho de entropia a cada vez que a árvore se divide. Isso quer dizer em suma que, escolhas vão sendo feitas e a árvore continua a ser dividida até que não se tenha mais nenhum ganho de informação advindo de divisões futuras, ou seja, até que o ramo contenha apenas instâncias de uma única classe, conforme mostrado na figura 2.3.

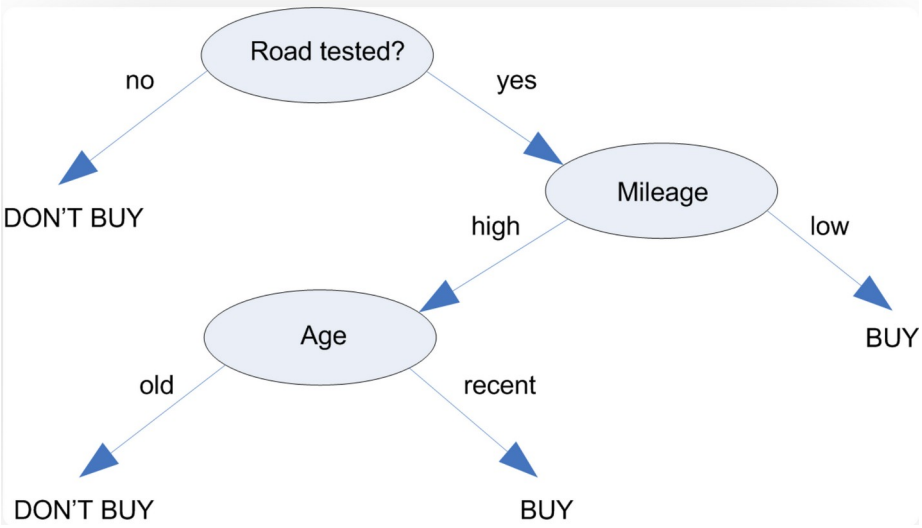

Figura 2.3 - Exemplo de árvore de classificação
FONTE: Anukrati Mehta (2019)

2.1.1.1.2 - *Random Forest*

Random Forest (figura 2.4) é um outro tipo de algoritmo de Classificação, mas que também pode ser usado em problemas de Regressão. Trata-se de um conjunto de árvores onde cada árvore responde a inúmeras questões, dividindo-se em vários ramos até chegarem a uma decisão final em suas folhas. A principal diferença é que cada árvore responde perguntas diferentes contidas no *dataset* de maneira aleatória. No final, a resposta mais presente (votada) entre as folhas de todas as árvores é a resposta fornecida pelo modelo final.

Para exemplificar como um modelo de *Random Forest* funciona, imagine que uma pessoa precisa decidir entre qual dos 3 restaurantes ela irá escolher: restaurante A, B ou C. Para decidir isso, ela pergunta a 100 pessoas diferentes, questões aleatórias sobre esse restaurante. Para algumas pessoas é perguntado sobre o sabor das comidas, para outras o preço. No final, cada pessoa que respondeu à pergunta, escolhe um restaurante dentre as 3 opções e no final, o restaurante mais votado é o escolhido.

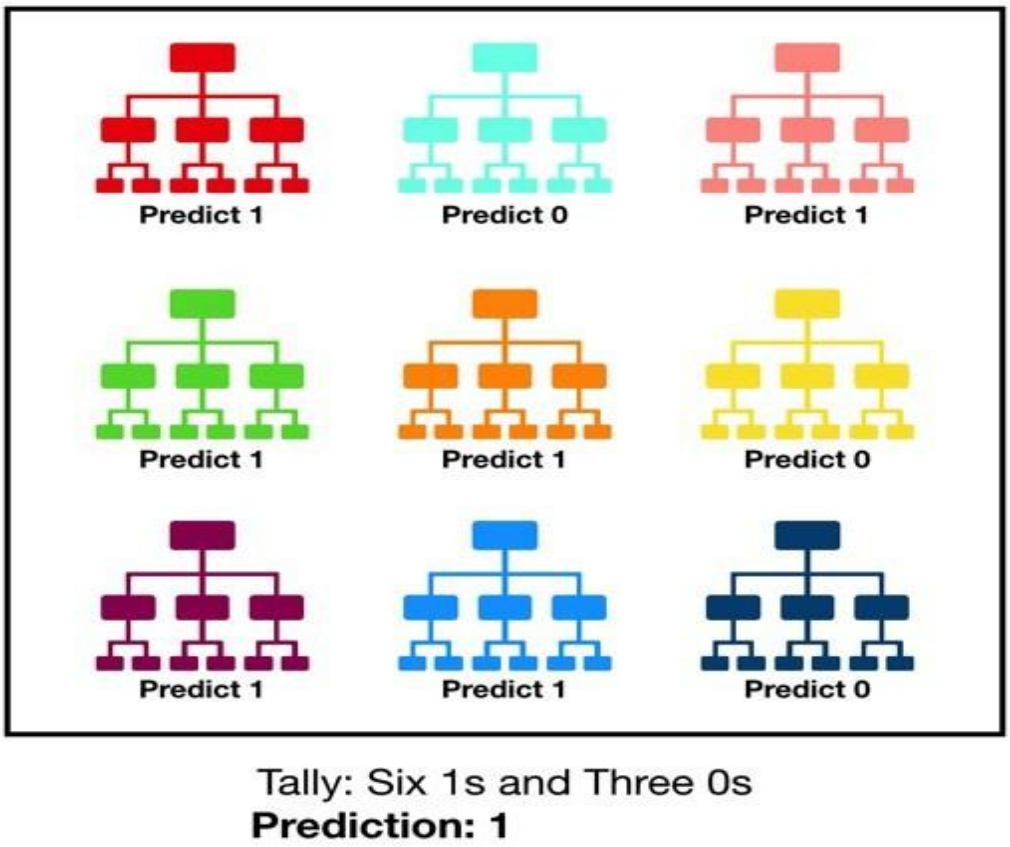

Figura 2.4 - *RandomForestClassifier*
FONTE: Tony Yiu (2019)

2.1.1.1.3 - KNN (K – Nearest Neighbor)

O algoritmo KNN ou K Vizinhos Mais Próximos, é um algoritmo que classifica as classes de acordo com as similaridades entre os dados novos e os dados já existentes. Similaridade no caso do KNN, significa a menor distância euclidiana (Equação 2.1) entre o novo dado a ser inserido e os K pontos existentes e já classificados no *dataset*. A figura 2.5 ilustra bem isso: para categorizar o ponto vermelho, é necessário considerar os k pontos mais próximos solicitados pelo algoritmo. K é um parâmetro escolhido no momento da aplicação do modelo. Quando escolhido K = 3, temos que o ponto vermelho é mais similar a classe B, quando é escolhido um K = 6, temos que esse ponto vermelho é mais similar a classe A.

Equação 2.1 *- Cálculo da distância Euclidiana*

Distância entre duas instâncias $\mathbf{p_i}$ e $\mathbf{p_j}$ definida como:

$$d = \sqrt{\sum_{k=1}^{n} (p_{ik} - p_{jk})^2}$$

$\mathbf{p_{ik}}$ e $\mathbf{p_{jk}}$ para k = 1, ..., **n** são os **n** atributos que descrevem as instâncias $\mathbf{p_i}$ e $\mathbf{p_j}$, respectivamente

FONTE: Italo José (2018)

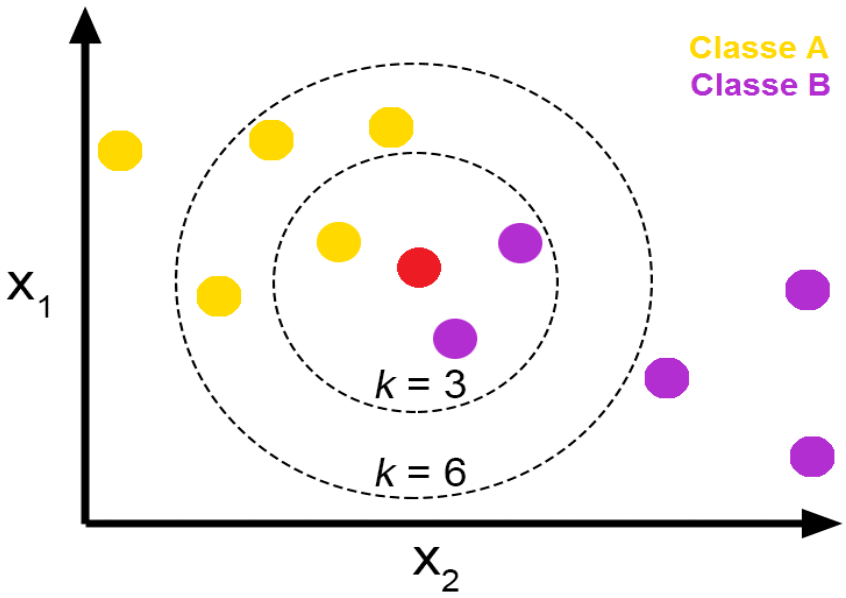

Figura 2.5 - Algoritmo *KNN*
FONTE: Italo José (2018)

2.1.1.2 - Algoritmos de Regressão

Os modelos de regressão têm como principal diferença em relação aos modelos de classificação, o fato de que as previsões são valores contínuos. Ou seja, o modelo não está tentando prever a que categoria pertence aquela variável alvo, mas sim qual o valor dela. A figura 9 ilustra bem isso, onde cada ponto representa um valor diferente. O modelo em questão tenta traçar uma linha entre todos os pontos de entrada, de maneira que a média da diferença entre os valores contínuos previstos e os reais, seja mínima.

Seguem abaixo alguns algoritmos de Regressão populares. Existem muitos outros algoritmos de regressão, mas explicar todos eles é uma tratativa que foge do escopo desta pesquisa.

2.1.1.2.1 - Regressão Linear

Esse é um dos principais algoritmos de regressão e consiste em atravessar uma linha chamada "Linha de Regressão" entre inúmeros pontos de entrada de dados, de maneira que essa linha reta (Linear) contemple o maior número de dados possíveis. Essa linha mede o relacionamento entre as variáveis.

Na figura 2.6, Y é a variável alvo que se quer prever e X é a variável independente. A ideia é que para cada valor de X, existe um valor de Y. No momento em que uma linha de regressão intercepta os pontos de entrada, o objetivo é garantir que a distância euclidiana entre os pontos e a linha de regressão seja a menor possível. Isso permitirá que para cada **novo** dado de X, seja possível definir o valor de Y de maneira assertiva, sendo Y e X valores contínuos. A métrica utilizada para avaliar a performance do modelo será explicada mais adiante.

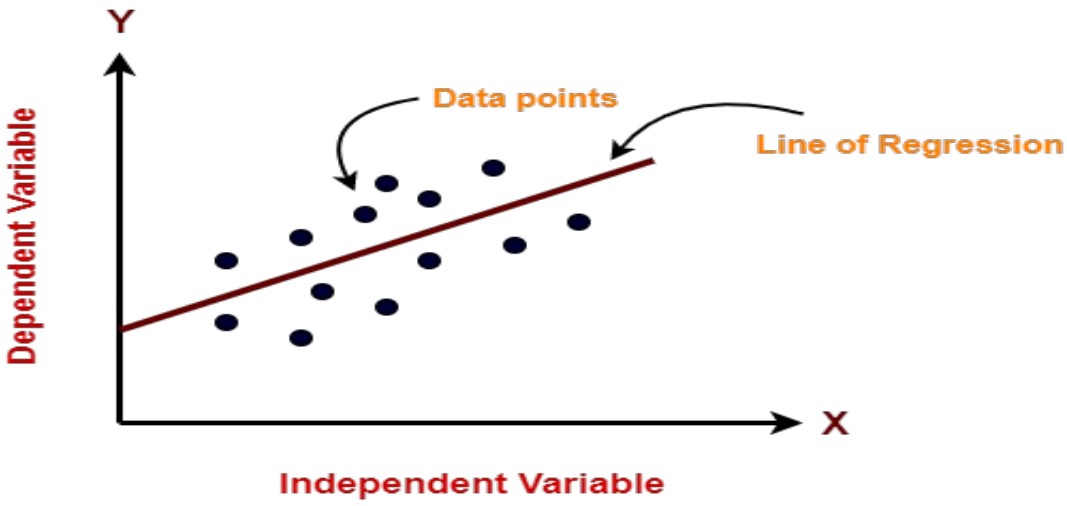

Figura 2.6 - Modelo de Regressão Linear
Fonte: Gaurav Sharma, 2022

2.1.2 - Aprendizado Não Supervisionado

Modelos de aprendizado de máquina Não Supervisionado, como o nome sugere, são criados a partir de bases de dados não rotuladas. Ou seja, todos os padrões existentes entre diferentes grupos da base de dados não podem ser identificados por treinamento e comparação com uma variável alvo e sim pela semelhança estatística entre esses grupos e identificação de relações entre as variáveis.

Figura 2.5 - Exemplo de clusterização de dados
Fonte: OpenCadd, 2023

Existem 3 tipos principais de algoritmos de Aprendizado Não Supervisionado: Regras de Associação, Clusterização (figura 2.7) e Redução de Dimensionalidade.

2.1.2.1 - Regras de Associação

São algoritmos do tipo não supervisionado que visam detectar padrões a partir da frequência com que diferentes pontos de dados aparecem juntos na base de dados, pois isso indica que existe uma certa relação entre esses pontos. Esse tipo de modelo é frequentemente utilizado em *E-commerce* no momento em que uma compra está sendo feita. O carrinho de compras digital irá ofertar outros produtos que são associados com o item que você está comprando e que são muitas vezes comprados juntos por outros clientes.

Um exemplo muito comum no mercado de algoritmo de Regras de Associação é o algoritmo *Apriori*.

O algoritmo *Apriori* é bastante popular entre os algoritmos de Regras de Associação e se baseia em uma ideia principal: A geração de suporte, onde define-se um limite mínimo para que um conjunto de produtos comprados juntos seja considerado frequente. A partir desse limite, novas associações são criadas até que não restem mais itens a ser associados.

2.1.2.2 - Clusterização

Os algoritmos de clusterização, nada mais são do que agrupamento de dados por semelhança. Como não existem rótulos para identificar a que classes pertencem os dados, eles (os dados) são agrupados em *clusters* de maneira que a semelhança entre os dados do mesmo cluster seja máxima e entre dados de clusters diferentes seja mínima.

Um exemplo muito comum no mercado de algoritmo de clusterização é o algoritmo K-Means.

O algoritmo *K-Means* é baseado na divisão dos dados em '*K' clusters*. O algoritmo divide os dados em K centróides, atribuindo cada *input* de dados ao *cluster* cujo centro seja mais próximo, conforme mostrado na figura 2.8.

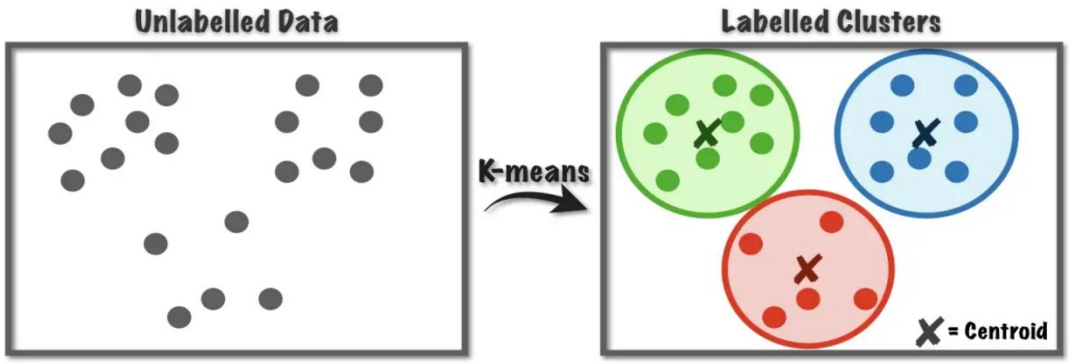

Figura 2.8 - *K-means* na prática
Fonte: Alan Jeffares, 2019

2.1.2.3 - Redução de Dimensionalidade

Os algoritmos de Redução de Dimensionalidade permitem diminuir a quantidade de variáveis (ou colunas) que são semelhantes entre si, sem perder informação relevante. Essa semelhança é identificada pela alta correlação entre variáveis independentes. Exemplo: Se na mesma base de dados, existem informações do período do dia (dia, tarde, noite) e da hora do dia, um desses dados pode ser eliminado porque estão provendo a mesma informação.

O motivo pelo qual a Redução de Dimensionalidade é usado é para diminuir a complexidade de modelos de aprendizado de máquina que não performam bem com uma quantidade muito grande de variáveis independentes correlacionadas. Modelos altamente complexos são mais imprecisos e mais lentos para rodar do que modelos menos complexos.

Um exemplo muito comum no mercado de algoritmo de Regras de Dimensionalidade é o algoritmo PCA.

PCA ou Análise de Componentes Principais utiliza as características geométricas da base de dados com n variáveis (ou n dimensões) de maneira que o algoritmo preserva os componentes principais - as características (variáveis) que explicam a maior parte da variação dos dados - e eliminam dados menos relevantes, combinando os mesmos em uma única variável.

2.1.3 - Aprendizado Por Reforço

Aprendizagem por Reforço é uma técnica em que um agente autômato precisa cumprir uma determinada tarefa, que a princípio, aparentemente não possui regras, através da estratégia de tentativa e erro. O agente autômato pode ser representado por uma Inteligência Artificial em jogos que aprende o estilo de jogo do jogador e se adapta a ele, recomendador de produtos comercializados no *E-commerce* de acordo com as características de compra do indivíduo.

O agente atua no ambiente com um método parecido com o da "força bruta", onde uma determinada solução é encontrada a partir de experiências de tentativas passadas. Nesse específico contexto, se parece um pouco com aprendizado supervisionado.

A grande diferença em relação ao aprendizado supervisionado é que o aprendizado por reforço, não tem um parâmetro de comparação ou uma resposta rotulada que indica o caminho que se quer chegar. O agente da ação recebe um estímulo (que pode ser negativo ou positivo) a cada novo passo que ele dá iterativamente. Esses estímulos guiam o agente da ação para a direção que possui um resultado mais vantajoso (onde ele recebe mais estímulos positivos), ao passo que o afasta de resultados prejudiciais (quando ele recebe estímulos negativos). Esse processo geralmente funciona a partir de um modelo matemático chamado Cadeia de Markov.

2.1.3.1 - Cadeia de Markov

Cadeia de Markov é um modelo matemático que explica um processo de tomada de decisão. Nele existe a representação de cada estado possível e da probabilidade do agente de transicionar para novos estados, dado o estímulo adequado.

Na figura 2.9, um agente que se encontra no estado “sorvete”, tem 10% de chances de continuar no mesmo estado e 70% de chances de mudar para o estado “correr”.

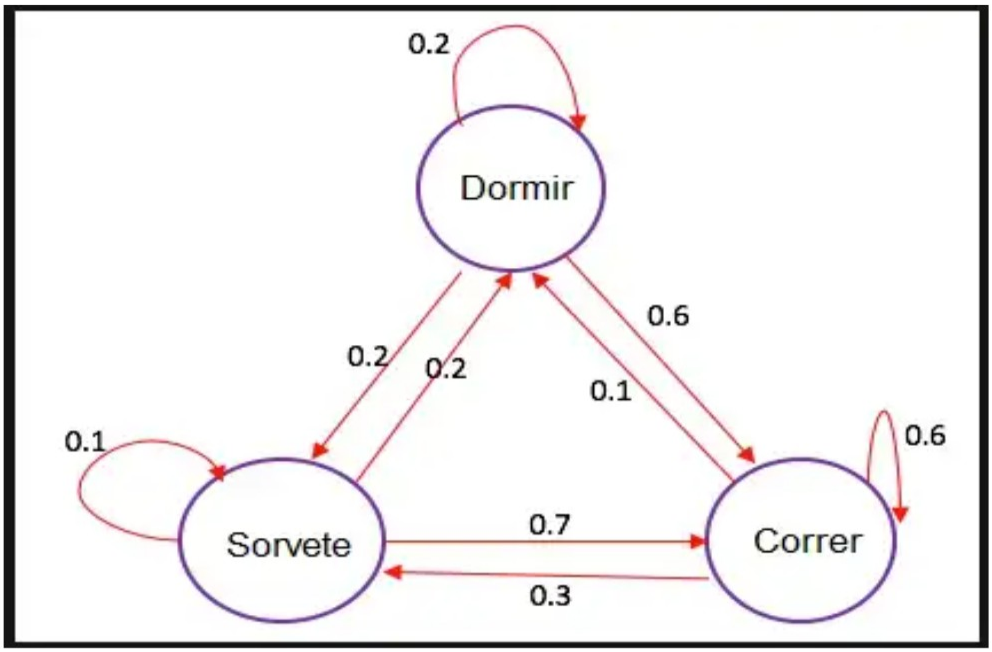

Figura 2.9 - Cadeia de *Markov* para 3 estados diferentes
Fonte: Enzo Neves, 2020

2.1.3.2 - Fases de Aprendizagem por Reforço

Existem tipicamente duas fases ao se aplicar essa estratégia de aprendizado: **Exploração** e **Aplicação**: A fase de exploração é onde o agente levanta informações sobre os processos de perdas e ganhos para todos os estados em que ele transiciona. Na fase de aplicação, o agente de fato aplica esse conhecimento de maneira que ele possa transicionar para os estados que lhes forneçam os maiores estímulos positivos possíveis.

CAPÍTULO 3

Definição das Métricas

Todas as métricas analisadas e discutidas a partir de agora serão de modelos de Aprendizado de Máquina Supervisionado. Isso porque ele é o tipo de modelo mais comumente utilizado no mercado, haja vista que os profissionais de dados possuem uma maior facilidade de explicar esses modelos e principalmente, aplicá-los.

Métricas de avaliação servem para validar a performance dos modelos preditivos. Depois que o modelo é treinado, é necessário saber se ele exercerá sua função de maneira adequada e, principalmente, se ele será efetivo. As métricas mais comuns serão explicitadas no decorrer dessa pesquisa.

Embora a métrica de validação mais utilizada em inúmeros modelos preditivos seja a acurácia, ela é apenas uma dentre os variados tipos de métricas existentes. Existem muitos outros tipos. O tipo de métrica a ser utilizado depende do problema em questão e do objetivo do modelo em si. Além do mais, as métricas de validação de performance entre modelos de classificação e regressão são diferentes. E também precisam ser interpretadas de maneiras diferentes.

As métricas de avaliação de performance que serão analisadas serão as seguintes: Acurácia, *Precision, Recall, F1 Score*, Matriz de Confusão, *AUC ROC* (curva *AUC*) e RMSE (Raiz Quadrada do Erro Médio).

3.1 - Acurácia

Acurácia (Equação 3.1) é uma das métricas mais utilizadas para a validação de um modelo preditivo principalmente por ser a mais simples de se entender. Ele é utilizado para mensurar a performance de algoritmos supervisionados, mais especificamente, algoritmos de classificação. Acurácia é o cálculo do racional entre a quantidade de previsões corretas divididas pelo total de previsões feitas.

Equação 3.1 - Fórmula simples da Acurácia

$$Acurácia = \frac{Previsões\,Corretas}{Total\,de\,Previsões\,Feitas}$$

Fonte: Mateus Pádua, 2020

Uma tarefa onde o cientista de dados precisa prever se amanhã fará sol ou não, se o cliente irá deixar de pagar suas parcelas ou não, se uma determinada transação é fraudulenta ou não, são todos exemplos de aplicações de modelos preditivos onde a acurácia pode ser utilizada. Muitas vezes, a classe alvo a qual precisa ser feita a previsão, sequer é binária. Sabendo-se que a os valores de acurácia variam de 0 a 100% (ou de 0.0 a 1.0), normalmente, o entendimento que os cientistas de dados mais inexperientes têm é que quanto maior a acurácia do modelo, melhor. Entretanto, será mostrado a seguir que nem sempre é assim.

Quando um modelo de classificação apresenta uma acurácia muito alta sem o devido tratamento de dados, um alerta precisa ser ligado na cabeça do cientista de dados que está realizando as análises preditivas. Uma acurácia muito alta, nem sempre significa uma performance confiável do modelo. Em muitos casos, esse tipo de situação pode ser causado por *overfitting*. Abaixo seguem os tipos de condições:

A. *Overfitting* é uma condição onde o modelo treinado, não consegue detectar nenhum padrão nos dados. Tudo que ele é capaz de fazer é decorar o modelo treinado. Isso significa que ele tem uma performance muito boa na fase de treinamento, mas se o mesmo modelo for utilizado em qualquer outra massa de dados, ele não será capaz de realizar previsões de forma acurada.

   Um modelo que utiliza uma base de dados muito complexa, ou seja, com uma quantidade muito grande de parâmetros, precisa tomar cuidado com a alta variância nos dados treinados. Se a base de dados possui muitos ruídos durante o treinamento, eles tendem a não ser muito flexíveis durante a fase de testes, ocasionando assim o *Overfitting*.

   Uma maneira de evitar o *Overfitting* é a redução da variância entre os dados. Além da diminuição dos ruídos (*outliers*). Existem inúmeras maneiras de se fazer isso, que incluem regularização L1 e L2 e que são capazes de detectar os melhores parâmetros durante um processo de aprendizado de máquina.

B. *Underfitting* é um tipo de problema que ocorre onde o modelo sequer consegue generalizar a massa de dados durante a fase de treinamento. Ou seja, o modelo performa mal mesmo durante as fases de treinamento e não consegue encontrar as relações entre os parâmetros. Na fase de teste, o modelo ficou ainda pior. Esse tipo de problema ocorre em modelos muito simples (poucos parâmetros) e/ou que não foram treinados adequadamente e apresentam um alto *Bias* (suposição errada da massa de dados).

   Existem muitas maneiras de evitar o *Underfitting* e a maioria desses métodos consistem em aumentar a quantidade de parâmetros na base de dados e melhorar os métodos de detecção de padrões dos modelos preditivos, diminuindo assim o *Bias*.

Em suma, um modelo não pode ser nem muito complexo e nem muito simples. É preciso encontrar um ponto ótimo que permita gerar um modelo preditivo competente e assertivo. Para isso, muitos testes e experimentos precisam ser realizados com as bases de dados. Conforme mostrado na figura 3.1, o ponto ótimo não é nem tão complexo e nem tão simples. Mas encontrar esse ponto ótimo é uma tarefa árdua para todos os cientistas de dados.

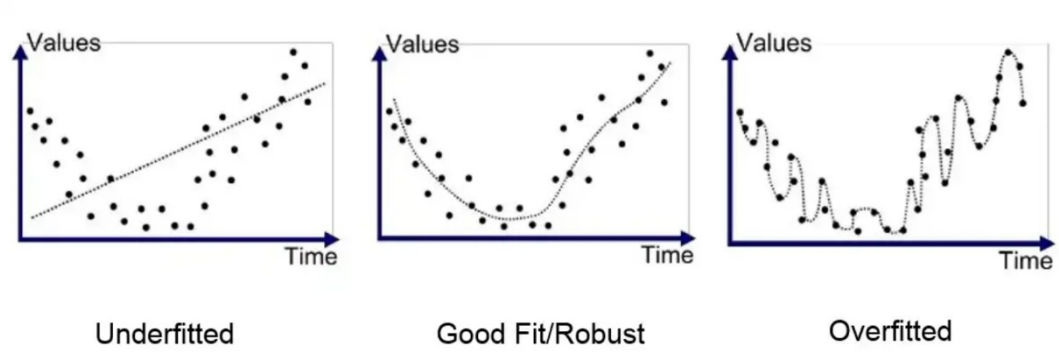

Figura 3.1 - *Overfitting, Underfitting* e um bom modelo
Fonte: Anup Bhande, 2018

3.1.1 - O problema das classes desbalanceadas

A validação de um modelo de classificação utilizando acurácia, só funciona quando a distribuição de classes da variável alvo apresenta o mesmo número de amostras para cada classe. Ou seja, se existe um *dataset* com 2 classes, isso significa que durante a modelagem, ele deve conter a mesma quantidade de registros para cada uma dessas classes (ou no mínimo, valores muito próximos).

Exemplo: Se uma amostra com 500 registros tem 2 classes diferentes na variável alvo, o ideal é ter 250 registros representando cada classe. Uma amostra com 600 registros e 3 classes da variável alvo precisa de 200 registros para cada uma das 3 classes. E assim por diante. A figura 3.2 mostra um exemplo de uma base balanceada e outra desbalanceada.

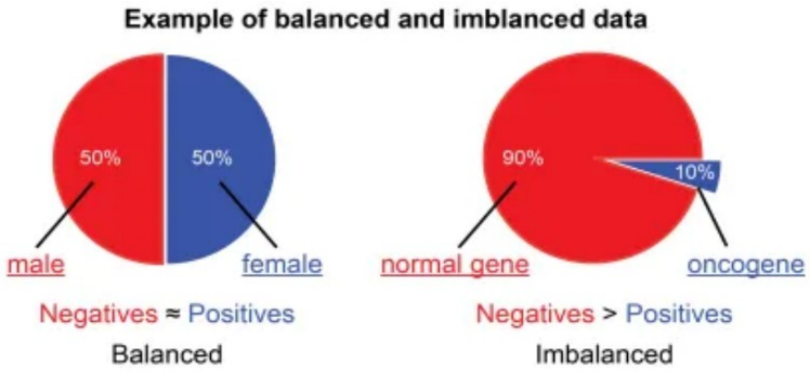

Figura 3.2 - Classes balanceadas e desbalanceadas
Fonte: Himanshu Tripathi, 2019

A necessidade de se trabalhar com bases balanceadas quando se está usando como métrica de validação a acurácia surge do fato de que o modelo pode ficar enviesado caso ele seja feito tendo como base um *dataset* desbalanceado.

Exemplo: Ao validar um modelo de classificação utilizando a acurácia em uma base cuja variável alvo seja binária e cuja distribuição está entre 90% dos registros para classe A e 10% dos registros para classe B, se esse problema não for endereçado durante a modelagem e se um modelo muito simples for feito em cima desse *dataset* desbalanceado, durante a fase de previsão, esse modelo iria "prever" que a maioria dos resultados pertencem à classe A. E mesmo que o modelo criado fosse tão simples que, durante a previsão, ele simplesmente chutasse que todos os resultados fossem pertencentes à classe A, o modelo iria acertar com uma acurácia de 90%.

Existem muitas maneiras de se resolver esse problema de desbalanceamento de bases. As técnicas de recalibração de registros (*Oversampling* e *Undersampling* conforme mostrado na figura 3.3**),** onde a distribuição de classes torna-se sinteticamente igualitária, são as estratégias mais comuns, mas também existem outras métricas de avaliação de performance muito mais robustas em relação a acurácia para esses tipos de problemas. Essas outras métricas serão explicitadas logo adiante.

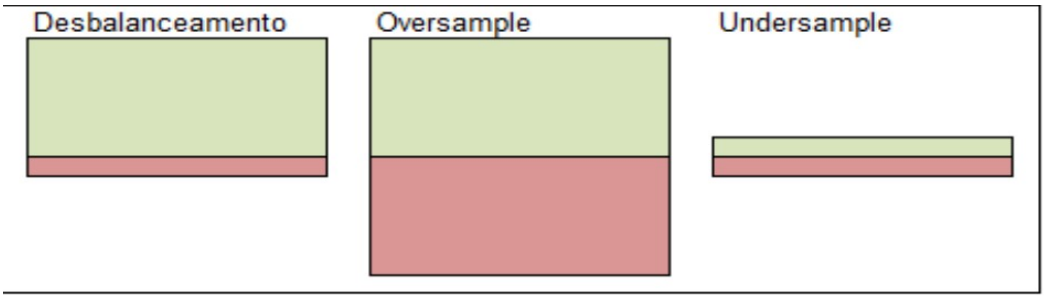

Figura 3.3 - *Oversampling e Undersampling*
Fonte: Bastos Stoll, 2020

3.2 - Termos importantes utilizados em Modelos Supervisionados

Para melhor entendimento das próximas métricas que serão demonstradas ao longo desse trabalho, primeiro precisa-se explicar alguns termos muito importantes: Verdadeiro Positivo, Verdadeiro Negativo, Falso Positivo, Falso Negativo, que são usados na validação de modelos de classificação binários.

3.2.1 - Verdadeiro Positivo

Verdadeiro Positivo é um termo que identifica uma previsão positiva (transação é fraude, cliente vai comprar, ação irá subir), como de fato positiva, de maneira acertada.

3.2.2 - Verdadeiro Negativo

Verdadeiro Negativo é um termo que identifica uma previsão negativa (transação não é fraude, cliente não vai comprar, ação irá cair), como de fato negativa, de maneira acertada.

3.2.3 - Falso Positivo

Falso Positivo, também conhecido como erro do Tipo I, é um termo que identifica uma previsão negativa (transação não é fraude, cliente não vai comprar, ação irá cair), como se fosse positivo, mesmo sem ser.

3.2.4 - Falso Negativo

Falso Negativo, também conhecido como erro do Tipo II, é um termo que identifica uma previsão positiva (transação é fraude, cliente vai comprar, ação irá subir), como se fosse negativa, mesmo sem ser;

3.3 - *Precision*

Em muitos problemas da vida real, não basta apenas que os modelos preditivos tenham uma alta acurácia. Porque como já foi dito anteriormente, isso nem sempre significa que o modelo seja assertivo. Formalmente, *Precision* ou Precisão (Equação 3.2) é uma métrica de avaliação de algoritmos supervisionados que é dada pela fórmula:

Equação 3.2 - Equação de Precisão

$$\frac{VP}{(VP+FP)}$$

Fonte: Mateus Pádua, 2020

, onde VP significa **Verdadeiro Positivo** e FP significa **Falso Positivo.**

Resumindo em outras palavras, Precisão representa a medida em que, dentre todas as previsões positivas que o algoritmo fez, acertando ou errando essas previsões, quantas estão de fato corretas? Dependendo do tipo de problema a ser analisado, a Precisão é muito mais importante que a acurácia como métrica de avaliação.

Por exemplo:

> Em um cenário de grandes contas de uma empresa, onde um Cientista de Dados precisa detectar fraudes sem perder os clientes grandes, é crucial que ao detectar que um desses clientes esteja cometendo fraudes, o profissional seja o mais preciso possível nessa acusação. Ou seja, de todas as empresas que ele previu que estavam cometendo fraudes, o que se quer é diminuir o máximo possível o número de falsos positivos.

3.4 - *Recall*

Recall (Equação 3.3) é uma outra métrica de validação de algoritmos supervisionados que é complementar a métrica Precisão. Ela é representada pela fórmula:

Equação 3.3 - Equação de Recall

$$\frac{VP}{(VP+FN)}$$

Fonte: Mateus Pádua, 2020

, onde VP significa **Verdadeiro Positivo** e FN significa **Falso Negativo.**

Resumindo em outras palavras, *Recall* representa à medida que, dentre todas as previsões positivas feitas, quantas de fato estão corretas? É importante em casos onde o falso negativo é mais prejudicial do que de costume.

Por exemplo:

Em um cenário onde seja necessário verificar a eficácia de exames médicos feitos para certas doenças altamente contagiosas, um modelo de *Machine Learning* de Classificação pode ser criado para prever se os exames conseguirão detectar se os pacientes estão ou não doentes. Se o resultado for um alto número de casos de falso negativos, os clientes que de fato estão doentes não irão procurar ajuda, pois eles receberam um diagnóstico de que estavam saudáveis quando na verdade não estavam. Consequentemente, essas pessoas irão infectar outras pessoas e assim por diante. Esse é o típico cenário onde se quer diminuir ao máximo a quantidade de falso negativos.

3.5 - *F1 Score*

F1 Score (Equação 3.4) é uma outra maneira de validação de modelos de Classificação binário. Como definição, essa métrica representa a média harmônica entre as métricas de Precisão e de *Recall*. Média harmônica é um tipo de média utilizada quando precisa-se representar valores que se comportam de maneira inversamente proporcional, como no caso do Precisão e *Recall*, em um único número.

Equação 3.4 - Equação de F1 Score

$$F1\ Score = 2 * \frac{Precisão * Recall}{Precisão + Recall}$$

Fonte: Mateus Pádua, 2020

Em outras palavras, a métrica *F1 Score* é um valor numérico que varia entre 0 e 1 (quanto maior, melhor) que agrega informações sobre "Precisão" e "*Recall*" em cenários onde garantir um bom desempenho em ambas as duas métricas seja importante. Ou seja, utiliza-se o *F1 Score*:

- Quando a amostra é desbalanceada;

- Quando o problema a ser resolvido, envolve um cenário em que tanto as previsões falso positivas, quanto as previsões falso negativas são muito prejudiciais e o objetivo é contemplar esses 2 problemas ao mesmo tempo.

3.6 - Matriz de Confusão

A matriz de confusão, nada mais é do que uma tabela contendo uma comparação entre os valores previstos e os valores reais da variável alvo. Comumente é uma matriz 2x2 onde ficam sumarizadas os valores previstos pelo modelo, cujo os resultados podem ser Falso Positivo, Falso Negativo, Verdadeiro Positivo ou Verdadeiro Negativo, que já foram explicitados anteriormente. Um exemplo de uma matriz de confusão é mostrado na figura 3.4.

Uma matriz de confusão com dimensões maiores do que 2 é possível, muito embora seu entendimento fique mais confuso conforme as dimensões forem aumentando.

Figura 3.4 - Matriz de Confusão
Fonte: Mateus Pádua, 2020

A partir dessa visualização resumida é possível calcular a Precisão, Recall, F1 Score, entre outros.

3.7 - Curva *ROC* e *AUC*

Para entender a curva *ROC* e *AUC*, alguns conceitos precisam ser cobertos primeiro. Esses conceitos são: Taxa de Verdadeiro Positivo e Taxa de Falso Positivo.

3.7.1 - Taxa de Verdadeiro Positivo

Taxa de verdadeiro positivo (Equação 3.5) é a relação entre a quantidade de previsões que são verdadeiras positivas, divididos pelo total de registros que de fato são positivos (TP + FN). Ou seja, trata-se do *Recall*, como já explicado anteriormente.

Equação 3.5 - Taxa de Verdadeiro Positivo

$$\frac{VP}{(VP+FN)}$$

Fonte: Mateus Pádua, 2020

3.7.2 - Taxa de Falso Positivo

Taxa de falso positivo (Equação 3.6) é a relação entre a quantidade de previsões falso positivas, divididos pela quantidade de registros que de fato são negativos (FP + TN).

Equação 3.6 - Taxa de Falso Positivo

$$\frac{FP}{(FP+TN)}$$

Fonte: Mateus Pádua, 2020

3.7.3 - Definição da Curva *ROC*

A definição mais simples da Curva *ROC* é que se trata de um gráfico que representa a relação entre a taxa de previsões verdadeiro positivas e a taxa de previsões falso positivas de um modelo preditivo de Classificação binário. Um bom modelo, tem uma taxa de verdadeiro positivos maior do que a taxa de falso positivo.

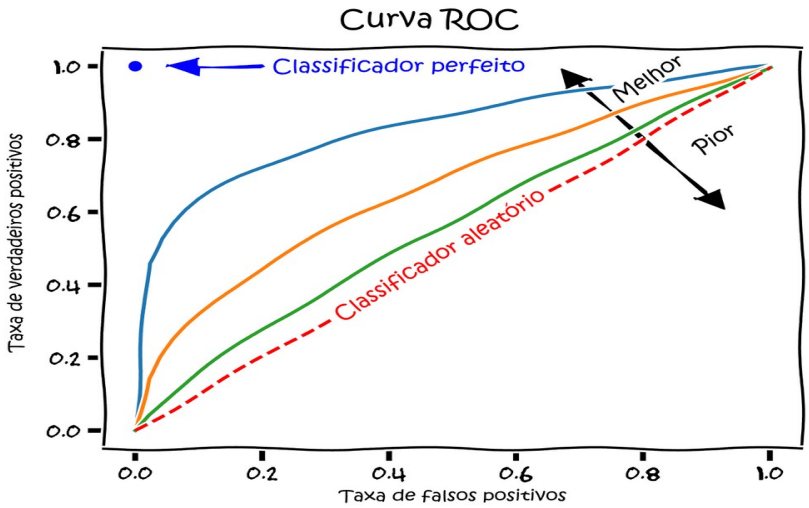

Figura 3.5 - Curva ROC detalhada
Fonte: Martin Thoma, 2021

Como mostrado na figura 3.5, existe uma linha na diagonal chamada "Classificador Aleatório". Isso ocorre porque, como o gráfico está mensurando um modelo de classificação binário, se a linha tracejada for exatamente na diagonal, significa que existem chances iguais de as duas previsões possíveis ocorrerem, sendo assim, um modelo de classificação aleatório (e que não consegue prever nada).

A imagem também mostra vários modelos preditivos diferentes, representados pelas linhas desenhadas no gráfico. Quanto maior a taxa de verdadeiro positivo, ou seja, quanto mais acima da linha diagonal for a linha referente ao modelo preditivo (e menor for a taxa de falso positivo), melhor será a capacidade do mesmo de separar – e distinguir - de maneira correta as classes da variável alvo binária.

3.7.4 - *AUC* (*Area Under Curve*)

AUC é um valor numérico que representa a área embaixo da curva e indica o grau de separabilidade entre as classes binárias. Ele varia entre 0 e 1 e quanto maior esse valor, melhor será o modelo.

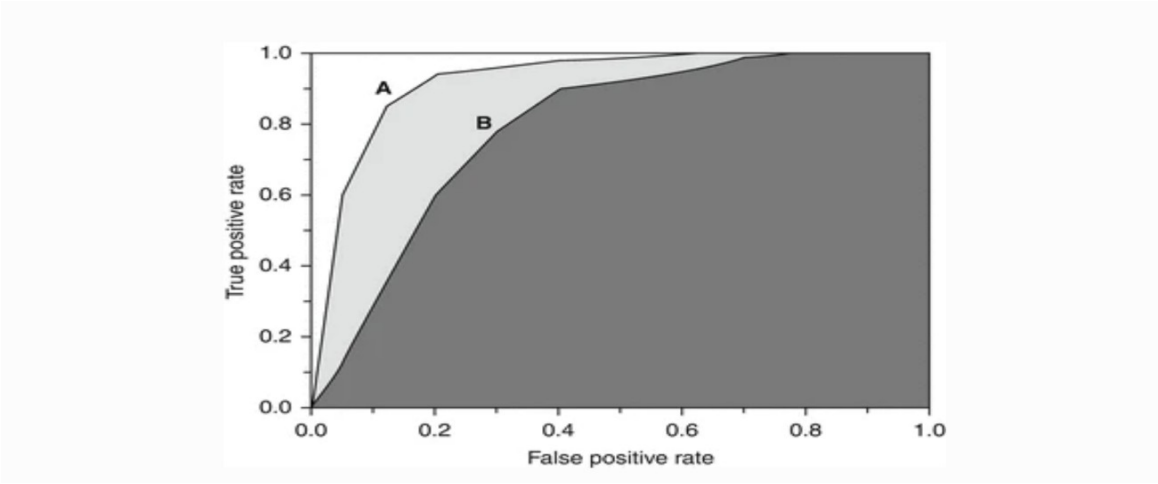

Figura 3.6 - Relação entre a Taxa de Verdadeiro Positivo e Falso Positivo entre o modelo A e o modelo B
Fonte: *Springer Link*, 2023

Conforme mostrado na figura 3.6, a área do modelo A é maior do que a área do modelo B. Ou seja, o modelo A é o melhor classificador binário, nesse caso.

3.8 - *RMSE* (Raiz Quadrada do Erro Médio)

Todas as métricas explicitadas até agora, são utilizadas para avaliar modelos supervisionados de Classificação. A Raiz Quadrada do Erro Médio (*RMSE* em inglês) é uma métrica que utiliza o erro médio visando avaliar modelos de Regressão. Como já dito anteriormente, os modelos de Regressão visam prever um número: o valor do preço de casas, a quantidade de material usado na confecção de produtos na indústria e etc. Então é importante que seja possível realizar uma previsão com o menor erro possível em relação ao valor real.

3.8.1 - MAE (Erro Médio Absoluto)

A primeira métrica de avaliação de modelos de Regressão é o Erro Absoluto Médio (*MAE*).

Equação 3.7 - Equação do Erro Absoluto Médio

$$MAE\,(y, \hat{y}) = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i|$$

Fonte: Clebio Junior, 2021

Conforme mostrado na Equação 3.7, o Erro Absoluto Médio calcula a diferença em módulo entre o valor real yi e o valor previsto ŷi, dividido pelo tamanho da amostra. Quanto menor o erro absoluto, melhor é o modelo. O problema em utilizar-se dessa métrica está quando temos valores fora da curva na amostra (*outliers*). Os valores discrepantes ficarão mascarados utilizando-se dessa métrica e em muitas análises, isso não é desejado.

3.8.2 - *MSE* (Erro Quadrado Médio)

O cálculo do erro quadrado médio é bastante parecido com o erro absoluto médio, conforme mostrado na Equação 3.8. Só que ao invés do módulo da diferença, é calculado o quadrado da diferença.

Equação 3.8 - Equação do Erro Quadrado Médio

$$MSE\,(y, \hat{y}) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

Fonte: Clebio Junior, 2021

Nesse caso, existe uma penalização maior por *outliers*. Ou seja, resultados muito diferentes entre o valor previsto e o valor real impactam o modelo de maneira mais incisiva. E o esforço para reduzir o valor dessa diferença precisa ser maior, fazendo esse modelo mais assertivo nesses casos discrepantes.

3.8.3 - *RMSE* (Raiz Quadrada do Erro Médio)

Essa métrica é uma evolução da métrica anterior (*MSE*) e, portanto, também penaliza outliers. Entretanto, ao elevar os valores ao quadrado como em *MSE*, perde-se a escala original da diferença entre os valores previstos e o real. Para mitigar esse problema, a raiz quadrada é utilizada no cálculo final, voltando à escala original, mas ainda penalizando *outliers*. Isso é mostrado na Equação 3.9.

Equação 3.9 - *Equação da Raiz Quadrada do Erro Quadrado Médio*

$$RMSE\left(y, \hat{y}\right) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2}$$

Fonte: Clebio Junior, 2021

CAPÍTULO 4

Métricas de Avaliação na Prática

Depois de explicitadas as principais métricas de avaliação de modelos preditivos supervisionados nos capítulos anteriores, neste capítulo será mostrado na prática algumas situações onde essas métricas podem ser usadas.

4.1 - Acurácia e o balanceamento de dados

Como dito anteriormente, a acurácia é a métrica mais conhecida em termos de avaliação de modelos preditivos. Mas nem sempre é recomendável sua utilização. Um dos critérios importantes para o seu uso ao avaliar modelos é uma base de dados balanceada. Segue abaixo, na tabela 4.1, um exemplo da base de dados Iris.

- Base de Dados Iris

Tabela 4.1 - *Base de dados da flor Iris*

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa

Fonte: Autor, 2023

A base de dados Iris é uma das bases de dados mais conhecidas para quem está começando a praticar Ciência de Dados. O problema a ser resolvido com essa base de dados é identificar qual das 3 espécies é a flor Iris (variável alvo "*species*") baseado em 4 parâmetros que identificam aspectos físicos da planta. As 3 espécies possíveis são "Iris-setosa", "Iris-versicolor" e "Iris-virginica".

Uma das características que fazem dessa base de dados um bom ponto de partida para iniciantes é o fato dela ser uma base comportada, ou seja, sem dados nulos, sem *outliers*, balanceada e etc. Além do mais, é uma base pequena. Existem 150

registros na base de dados ao total e 50 registros para cada um dos 3 valores possíveis da variável alvo, conforme mostrado na figura 4.1.

```
dados.shape
```

```
(150, 5)
```

```
dados.groupby('species').size()
```

```
species
Iris-setosa        50
Iris-versicolor    50
Iris-virginica     50
dtype: int64
```

```
dados.isnull().sum()
```

```
sepal_length    0
sepal_width     0
petal_length    0
petal_width     0
species         0
dtype: int64
```

Figura 4.1 - Base de dados pequena, balanceada e sem valores nulos
Fonte: Autor, 2023

Em uma base comportada como essa, é esperado que a acurácia seja bastante assertiva. A figura 4.2 mostra o cálculo da acurácia depois de treinar modelos de classificação em 5 algoritmos distintos, todos com acurácia acima de 90%:

```
DECISION TREE CLASSIFIER =>
Dataframe Original => Accuracy: 0.9466666666666667

LOGISTIC REGRESSION CLASSIFIER =>
Dataframe Original => Accuracy: 0.9733333333333334

RANDOM FOREST =>
Dataframe Original => Accuracy: 0.9600000000000002

KNN =>
Dataframe Original => Accuracy: 0.9733333333333334

XGBOOST CLASSIFIER =>
Dataframe Original => Accuracy: 0.9466666666666667
```

Figura 4.6 - Acurácia de 5 modelos distintos
Fonte: Autor, 2023

- Base de dados de transações em cartões de crédito

Esse é um outro tipo de base de dados bastante conhecido. Existem várias bases de dados como essa que tentam responder a mesma pergunta: baseado em características de transações anteriores, quais transações futuras serão fraudulentas ou não? Entretanto, ao contrário da análise feita sobre a base de dados Iris, essas bases de dados sobre fraudes costumam não ser muito comportadas.

A base de dados utilizada para demonstrar métricas de avaliação para esse tipo de problema está na Tabela 4.2 abaixo. Trata-se de uma base de dados de transações de cartões de crédito cuja variável alvo chama-se "TARGET". TARGET == 0 significa uma transação legítima e TARGET == 1 é uma transação fraudulenta.

Tabela 4.2 - Base de dados de transações de cartões de crédito

```
df.head()
```

	SAFRA	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12	CEP	TARGET
0	201901	NaN	8.1	9.99	1968	0.0	0	15.15	0.0	0.0	0	SP	São Paulo	8412006	0
1	201910	0.0	4.4	35.00	1369	0.0	0	63.98	1.0	0.0	0	RJ	Rio de Janeiro	23580304	0
2	201906	0.0	0.7	52.99	1228	0.0	0	98.84	0.0	0.0	0	MG	Belo Horizonte	30421310	0
3	201910	0.0	63.3	810.00	0	0.0	1	9237.21	0.0	0.0	0	SP	São Paulo	8253410	0
4	201902	0.0	4.1	17.50	0	0.0	1	27.70	1.0	0.0	0	ES	Vitória	29017186	0

Fonte: Autor, 2023

Quase todas as vezes esse tipo de problema apresenta uma base de dados desbalanceada, até pela própria natureza do problema: a quantidade de transações fraudulentas é muito menor do que a quantidade de transações legítimas. Do contrário, as empresas de crédito iriam à falência. Conforme mostrado na figura 4.3, esse problema é bem evidente nessa base de dados e ele precisa ser endereçado.

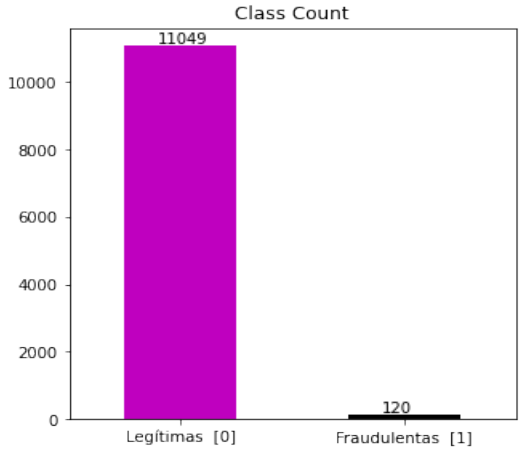

Figura 4.3 - Diferença de número de transações legítimas e fraudulentas
Fonte: Autor, 2023

Por apresentar muitos problemas, especialmente em relação ao balanceamento, essa é uma base de dados onde a utilização da acurácia como uma métrica de avaliação não é recomendada. Além disso, também existem algumas variáveis nulas no *dataframe* original, conforme mostrada na figura 4.4:

	Valores Nulos (%)
SAFRA	0.0%
V1	6.55%
V2	2.03%
V3	0.0%
V4	0.0%
V5	8.11%
V6	0.0%
V7	1.44%
V8	3.12%
V9	1.0%
V10	0.0%
V11	0.0%
V12	0.0%
CEP	0.0%
TARGET	0.0%

Figura 4.4 - Porcentagem de variáveis nulas de cada variável
Fonte: Autor, 2023

Existem algumas maneiras de se lidar com bases que possuem os mais diversos problemas. Eis a listagem de alguns desses problemas encontrados nessa base de dados e como eles foram resolvidos:

- Valores nulos:
    - Uma maneira muito comum de resolver esse tipo de problema é preencher os valores nulos categóricos com a moda (valores mais frequentes) e valores numéricos com a média.
- Tipos de dados diferentes:
    - Existem muitos problemas que podem ocorrer com o *dataframe* caso ele não possua o tipo de dado adequado. Por exemplo, datas podem ser impactadas se o seu tipo de dado for uma *String* (*Object*). Existem muitas maneiras de se converter os dados para o tipo apropriado e mitigar esse tipo de problema.
- Dados desbalanceados:
    - Existem muitas maneiras de lidar com dados desbalanceados. As duas maneiras mais comuns são as seguintes:

        ***Undersampling*:** Trata-se de uma categoria de técnicas que visa diminuir aleatoriamente a quantidade de dados da classe majoritária da variável alvo. No caso trabalhado, essa técnica visa diminuir a quantidade de transações legítimas de maneira que ela se iguale a quantidade de dados fraudulentos.

        ***Oversampling*:** É o contrário de *Undersampling*. Esse conjunto de técnicas visa aumentar artificialmente e aleatoriamente, a quantidade de dados da classe minoritária da variável alvo. No caso trabalhado, essa técnica visa aumentar a quantidade de registros de transações fraudulentas de maneira que ela se iguale a quantidade de registros de transações legítimas.

Em uma base sem tratamento, ao realizar a avaliação da performance do poder de previsão do modelo treinado, a acurácia pode apresentar resultados que levam a conclusões erradas. Ou seja, o modelo pode performar muito bem na base de testes (acurácia alta), mas na hora de testar em dados nunca antes vistos, sua verdadeira performance pode ser bastante degradada em virtude das características problemáticas

da base. Portanto, é importante avaliar o modelo preditivo utilizando-se também de outras métricas como Precisão*, Recall, F1 Score* e Curva *AUC*.

```
DECISION TREE CLASSIFIER =>
Dataframe Original => Accuracy: 97.69%
Dataframe Original => F1 Score: 15.65%
Dataframe Original => ROC AUC Score: 57.04%

Dataframe Tratado => Accuracy: 98.88%
Dataframe Tratado => F1 Score: 98.86%
Dataframe Tratado => ROC AUC Score: 98.83%
```

Figura 4.5 - *Decision Tree* em bases tratadas e não tratadas
Fonte: Autor, 2023

A figura 4.5 mostra a performance de um modelo criado com o algoritmo *Decision Tree* em dois *dataframes* de testes diferentes: um *dataframe* onde seus problemas foram tratados e outro *dataframe* original, sem tratamento. Para cada um desses *dataframes*, foi criado um modelo que foi avaliado utilizando-se de 3 métricas diferentes: Acurácia, *F1 Score* e Curva *AUC* (todas previamente explicadas). É visível que a acurácia muda muito pouco em relação ao *dataframe* problemático e o *dataframe* tratado. Isso porque a acurácia já era muito alta mesmo com o *dataframe* original. Já as métricas *F1 Score* e Curva *AUC* mudam completamente quando o *dataframe* é tratado. Isso é um ponto positivo dessas métricas porque mostra de maneira mais clara os efeitos que um tratamento de dados adequado pode ter na criação de um modelo preditivo. E como o tratamento de dados é uma das partes mais importantes em um projeto de dados, é importante mostrar o seu impacto de maneira bem evidente na criação de modelos preditivos.

4.2 - Métrica de Regressão

Como dito anteriormente, ao se criar modelos supervisionados de Regressão, o objetivo é realizar previsões numéricas, onde a variável alvo é um valor contínuo. E um modelo preditivo é tão bom quanto a sua capacidade de realizar as previsões que apresentem uma diferença pequena entre o valor previsto e o valor real.

A tabela 4.3 mostra uma base de dados que será usada como referência para explicitar as métricas de avaliação de modelos de Regressão. É uma base de dados onde cada linha representa a venda de uma casa, onde existem várias colunas

contemplando as características da casa e o seu preço final. O objetivo é identificar qual será o valor final da casa, baseado em inúmeras qualidades contidas nessas colunas. A variável alvo chama-se "*SalePrice*"

Tabela 4.3 - *Base de dados dos registros de vendas de casas*

```
train = pd.read_csv('train_house_prediction.csv')
train.head()
```

otArea	Street	Alley	LotShape	LandContour	Utilities	...	PoolArea	PoolQC	Fence	MiscFeature	MiscVal	MoSold	YrSold	SaleType	SaleCondition	SalePrice
8450	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	NaN	0	2	2008	WD	Normal	208500
9600	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	NaN	0	5	2007	WD	Normal	181500
11250	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	NaN	0	9	2008	WD	Normal	223500
9550	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	NaN	0	2	2006	WD	Abnorml	140000
14260	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	NaN	0	12	2008	WD	Normal	250000

Fonte: Autor, 2023

Essa base contém mais 80 colunas, e eventualmente precisará ser resumida de maneira que um modelo muito complexo não seja criado.

Dimensões do DataFrame:
Linhas: 1460
Colunas: 81

Os problemas preditivos de Classificação necessitam de uma base de dados tratada adequadamente para que os resultados sejam coerentes. O mesmo pode ser dito dos problemas de Regressão. Serão apresentados a seguir alguns dos problemas encontrados na base de dados bem como algumas das técnicas utilizadas para resolver esses problemas.

A base utilizada para avaliar as métricas de Regressão possui alguns problemas comuns a esse tipo de problema:

- Valores nulos;
- Valores Categóricos;
- Distribuição enviesada para a Direita (não normal);
- Uma quantidade muito grande de *features* (colunas);

Para endereçar todos esses problemas algumas técnicas *de feature engineering* foram empregadas. Seguem algumas estratégias utilizadas:

4.3. –Tratamento de Valores Nulos

```
train.isnull().sum()
```

```
Id                0
MSSubClass        0
MSZoning          0
LotFrontage     259
LotArea           0
Street            0
Alley          1369
LotShape          0
LandContour       0
Utilities         0
LotConfig         0
LandSlope         0
```

Figura 4.6 - Valores nulos

Fonte: Autor, 2023

Para os casos de colunas com muitos valores nulos (Figura 4.6), essas foram deletadas do *dataframe*, pois não representam dado significativo ao modelo. Caso permanecessem na base, iria atrapalhar o modelo preditivo criado por torná-lo muito complexo com dados que são irrelevantes do ponto de vista estatístico.

Para alguns casos numéricos, a média foi utilizada como forma de preencher esses valores nulos. Em alguns casos, foi necessário utilizar a média agrupada por localidade, de maneira que dados enviesados não fossem imputados à base. Para todos os outros casos, com uma quantidade pequena de valores nulos, essas linhas comprometidas foram deletadas e não a coluna inteira.

4.4 – Tratamento de Valores Categóricos

Valores categóricos são um grande problema para os algoritmos preditivos. Especialmente nessa base de dados, onde existem muitas colunas categóricas, conforme mostrado na Tabela 4.4. Para resolver esses problemas, as estratégias *OneHot Encoder* e *Label Encoder* foram usadas.

Tabela 4.4 - *Colunas Categóricas*

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	...	ScreenPorch	PoolArea	PoolQC	Fence
0	1461	20	RH	80.0	11622	Pave	NaN	Reg	Lvl	AllPub	...	120	0	NaN	MnPrv
1	1462	20	RL	81.0	14267	Pave	NaN	IR1	Lvl	AllPub	...	0	0	NaN	NaN
2	1463	60	RL	74.0	13830	Pave	NaN	IR1	Lvl	AllPub	...	0	0	NaN	MnPrv
3	1464	60	RL	78.0	9978	Pave	NaN	IR1	Lvl	AllPub	...	0	0	NaN	NaN
4	1465	120	RL	43.0	5005	Pave	NaN	IR1	HLS	AllPub	...	144	0	NaN	NaN

5 rows × 80 columns

Fonte: Autor, 2023

Primeiramente, foi tentado utilizar a técnica do *OneHot Encoder*. Entretanto, como ele binariza os valores das classes categóricas e transforma cada valor em uma coluna diferente. O resultado final foi um *dataframe* cuja complexidade foi aumentada consideravelmente em virtude do aumento expressivo do número de colunas. Mais de 200 colunas diferentes foram criadas com essa técnica. Esse aumento iria resultar em um modelo preditivo muito complexo. Em virtude disso, a técnica escolhida foi a *Label Encoder*, que apenas transforma cada valor em um número diferente e isso é feito para todas as colunas.

4.5 - Distribuição Enviesada para a Direita

Com a distribuição Enviesada para a Direita, a maioria dos dados caem para a direita, do lado positivo do gráfico, conforme mostrado na figura 4.7. Essa característica dificulta na atribuição de um valor típico (ou normal, comum) a variável, pois não há um ponto central claro da distribuição desses valores. Além do mais, os valores extremos na cauda longa podem ter um significativo impacto no resultado final, tornando a capacidade de realizar previsões do modelo prejudicada.

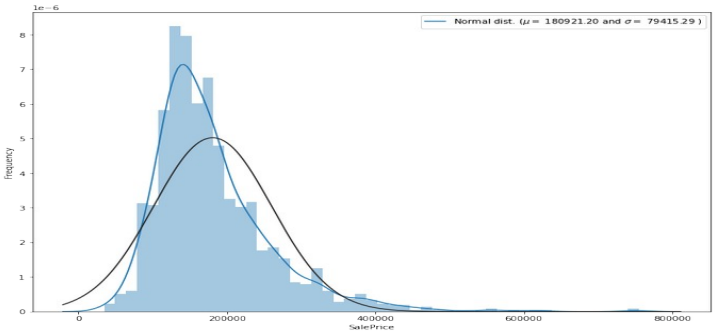

Figura 4.7 - Distribuição Enviesada para Direita
Fonte: Autor, 2023

Para amenizar esse problema, uma técnica bastante utilizada é a transformação dos dados para uma distribuição mais simétrica. Nesse caso, a transformação utilizada foi a logarítmica. Entretanto, ao utilizar essa transformação para criar o modelo preditivo, é

importante voltar os dados para sua distribuição inicial para que os valores previstos apresentados sejam os corretos.

4.6 - Quantidade muito alta de *features* (colunas)

Como já dito anteriormente, um *dataframe* com uma quantidade muito grande de dimensões, pode torná-lo impraticável, tanto para treiná-lo quanto para testá-lo. Isso porque, conforme o número de colunas cresce, a quantidade de combinações que os algoritmos preditivos precisam analisar, cresce exponencialmente.

Para lidar com o problema da alta dimensionalidade, o primeiro passo é encontrar a quantidade ótima de *features* para esse modelo. Isso foi feito de maneira iterativa, diminuindo-se a quantidade de *features*, um a um, e recalculando a performance do modelo na previsão da variável alvo. Existem algumas bibliotecas prontas em *Python* que realizam essa tarefa, como o *RFE*. No caso dessa base de dados, a quantidade ótima de *features* que era de mais de 80, diminuiu para 22.

Após descobrir o número de colunas ótimo para a base de dados, as colunas mais importantes selecionadas são utilizadas para se criar um modelo preditivo menos complexo e mais relevante.

Depois que a base de dados é tratada, ela está pronta para ser utilizada na criação de um modelo preditivo de Regressão. Independente da métrica de avaliação de performance utilizada (Erro Médio Absoluto, Erro Quadrado Médio, Raiz do Erro Quadrado Médio), o objetivo final é diminuir a diferença entre os valores previstos e o valor real.

Como já foi dito, os modelos preditivos de Regressão trabalham com variáveis alvos de valores contínuos. Isso significa que esses modelos têm como output um valor aproximado ao valor real daquilo que está sendo previsto. Entretanto, essa diferença dificilmente será zero pois entre um número contínuo e outro, existem infinitos valores. Além do mais, o objetivo é que o modelo seja capaz de generalizar para várias outras entradas de dados, portanto esse comportamento não só é improvável, como é indesejado.

A escolha do modelo de Regressão foi feita da mesma forma que os modelos de Classificação: foram testados vários algoritmos, testando métricas de avaliação diferentes. O algoritmo previsor que se saiu melhor, foi o escolhido. E nesse caso, foi o algoritmo chamado ***Gradient Boost***. Pois ele apresentou a menor diferença entre os valores previstos e o real.

Mesmo com o tratamento de dados, existem ainda diferenças consideráveis na previsão dos valores contínuos. Isso significa que ainda existem várias melhorias que podem

ser implementadas no modelo preditivo, como o *grid Search* na busca de parâmetros adequados para o algoritmo de previsão.

Ao analisar a métrica *MSE*, nota-se uma alta discrepância em relação à previsão dos valores esperados. Isso porque, os valores estão elevados ao quadrado e não representam os valores das previsões reais. Para não deixar a análise enviesada, é feito a raiz quadrada dessa métrica, dando origem a uma outra métrica de performance que é a mais usada para modelos de Regressão: *RMSE*.

O erro médio absoluto (*MAE*) possui a menor diferença encontrada entre os valores previstos. Embora o *MAE* seja uma métrica adequada para avaliar o desempenho de um modelo de Regressão nesse caso específico, em muitos outros casos ele não é o mais adequado. Por exemplo, quando temos uma variável alvo com valores negativos ou com muitos outliers, a melhor alternativa é elevar os erros ao quadrado para eliminar esses valores negativos – criando assim a métrica de erro quadrado médio *MSE*. Além do mais, ao fazer isso, um modelo mais sensível a *outliers* será criado, facilitando assim a identificação dos mesmos e posteriormente, eles podem ser tratados da maneira adequada. Cada uma das métricas de Regressão utilizadas para avaliar o modelo preditivo encontra-se na figura 4.8.

```
MAE  => (Erro Médio Absoluto):  48.54609968264512
MSE  => (Erro Quadrático Médio):  3999.469714109244
RMSE => (Erro médio quadrático da raiz):  63.24136078634965
```

Figura 4.8 - Métricas de performance de modelos de Regressão
Fonte: Autor, 2023

4.7 - Resultados Esperados

Ao decorrer do texto foram explicitadas inúmeras maneiras de avaliar a performance de um modelo preditivo. Tanto de Classificação quanto de Regressão. Ao mesmo tempo, foram analisadas 3 bases de dados diferentes, com seus problemas variados, que tiveram que ser tratados para que pudessem ser utilizados na criação de modelos preditivos, utilizando-se os algoritmos de previsão adequados. A escolha das bases foi feita de maneira didática e o objetivo principal, não era criar um modelo preditivo que fosse 100% a prova de falhas. E sim mostrar os caminhos e as linhas de pensamento que devem ser seguidos quando se está trabalhando em um projeto de *Data Science* e principalmente, como avaliar um modelo de *Machine Learning* da maneira mais pertinente possível.

4.8 - Resultados obtidos

O objetivo principal desse projeto era apresentar diferentes maneiras de se avaliar a performance de um modelo preditivo. E esse objetivo foi concluído. Vários conceitos de exploração e tratamento de dados foram discutidos, bem como maneiras adequadas de se utilizar métricas de performance de acordo com o problema apresentado.

Para esse projeto, as principais bibliotecas de Python de Ciência de Dados foram utilizadas como o Pandas e o *Numpy*. Além do mais, foram testados vários algoritmos para a escolha e criação de modelos preditivos, tanto de Classificação, quanto de Regressão, como o *Decision Tree* e o *Random Forest.*

CAPÍTULO 5
Conclusão e Trabalhos Futuros

5.1 - Conclusão

A última etapa de um projeto de dados, costuma ser uma etapa negligenciada, especialmente quando o modelo está em produção. Principalmente no que tange a escolha da métrica adequada que irá balizar o quão efetivo está o modelo preditivo. Essa pesquisa mostrou através de análises empíricas (e bibliografia teórica) que nem sempre é uma boa ideia utilizar a acurácia como métrica avaliadora, mesmo que ela seja a mais utilizada em projetos no mercado. Mostrou também que um bom tratamento nos dados se faz necessário quando se precisa apresentar uma melhor performance em modelos preditivos. Bases desbalanceadas, muitos outliers e dados categóricos custam caro na pontuação positiva desses modelos. Conclui-se portanto, que uma boa limpeza de dados aliado a escolha adequada de métricas de avaliação são partes essenciais de qualquer projeto de Ciência de Dados.

5.2 – Trabalhos Futuros

Para trabalhos futuros, será considerado um projeto completo de Ciência de Dados. Dessa vez, abrangendo inclusive, as etapas iniciais que foram preteridas nesse trabalho em função das etapas finais, que é a avaliação de métricas de modelos preditivos supervisionados.

REFERÊNCIAS: