Universidade Federal de Uberlândia
Faculdade de Matemática

Bacharelado em Estatística

CLASSIFICAÇÃO DO PERFIL DOS CLIENTES ATRAVÉS DE TÉCNICAS DE MINERAÇÃO DE DADOS

Matheus Henrique dos Santos

Uberlândia-MG
2023

Matheus Henrique dos Santos

CLASSIFICAÇÃO DO PERFIL DOS CLIENTES ATRAVÉS DE TÉCNICAS DE MINERAÇÃO DE DADOS

Trabalho de conclusão de curso apresentado à Coordenação do Curso de Bacharelado em Estatística como requisito parcial para obtenção do grau de Bacharel em Estatística.

Orientador: Prof. Dr. José Waldemar da Silva

Uberlândia-MG

2023

Universidade Federal de Uberlândia
Faculdade de Matemática

Coordenação do Curso de Bacharelado em Estatística

A banca examinadora, conforme abaixo assinado, certifica a adequação deste trabalho de conclusão de curso para obtenção do grau de Bacharel em Estatística.

Uberlândia, ____ de _____ de 20____

BANCA EXAMINADORA

_____

Prof. Dr. José Waldemar da Silva

_____

Profa. Dra. Elaine Ribeiro de Faria Paiva

_____

Prof. Dr. Lucio Borges de Araujo

Uberlândia-MG

2023

Resumo

Atualmente volumes de dados cada vez maiores são gerados, porém nem sempre os mesmos são utilizados de forma "inteligente" para extração de informações. Nesse cenário, as ferramentas de mineração de dados são utilizadas para auxiliar a extrair informações que nem sempre são tão óbvias ou explícitas, o que faz com que ela seja uma das tecnologias mais promissoras da atualidade. O presente trabalho foi desenvolvido com a finalidade de criar um modelo de classificação, através de técnicas de mineração de dados, para a classificação do perfil dos clientes (Cliente Venda ou Cliente Não Venda) de uma empresa real. A estrutura deste Estudo de Caso utilizou a metodologia CRISP-DM, que é a mais usual em problemas que envolvam Mineração de Dados. Essa metodologia é segmentada em 06 etapas, que são: Entendimento do Negócio, Entendimento dos Dados, Preparação dos Dados, Modelagem, Avaliação e Utilização. Dois algoritmos de Classificação foram testados para desenvolvimento do modelo: Árvore de Decisão e Regressão Logística. Para aplicação nos algoritmos, a base de dados foi dividida em duas bases: Base de Treino (composta por 70% dos dados) e Base de Teste (Composta por 30% dos dados). Por se tratar de uma base com dados desbalanceados, devido a expressiva superioridade de clientes na classe Não Vendas, duas metodologias foram avaliadas para balanceamento dos dados: aplicação do SMOTE (*Synthetic Minority Oversampling Technique*) para técnica de sobreamostragem e ajuste dos pesos das classes minoritária e majoritária dentro das funções de custo dos algoritmos. Os modelos de Regressão Logística tanto para a aplicação do SMOTE quanto para ajuste dos pesos das classes apresentaram resultados semelhantes, e foram os que apresentaram melhores resultados. Com isso, o estudo permitiu concluir que o modelo de classificação encontrado, através da técnica de Regressão Logística, é de extrema relevância para a empresa, resultando em uma redução dos custos com a contactação dos clientes e um aumento expressivo do volume de retorno desses contatos (Vendas).

Palavras-chave: Aprendizado de Máquina, Regressão Logística, Árvore de Decisão, Perfil de Clientes.

Abstract

Currently, ever-increasing volumes of data are generated, but they are not always used in an "intelligent" way to extract information. In this scenario, data mining tools are used to help extract information that is not always so obvious or explicit, which makes it one of the most promising technologies today. The present work was developed with the purpose of creating a classification model, through data mining techniques, for the classification of the profile of the customers (Customer for Sale or Customer Not for Sale) of a real company. The structure of this Case Study used the CRISP-DM methodology, which is the most common in problems involving Data Mining. This methodology is segmented into 06 stages, which are: Business Understanding, Data Understanding, Data Preparation, Modeling, Evaluation and Use. Two Classification algorithms were tested for model development: Decision Tree and Logistic Regression. For application in the algorithms, the database was divided into two bases: Training Base (composed of 70% of the data) and Test Base (Comprised of 30% of the data). As this is a database with unbalanced data, due to the significant superiority of customers in the Non-Sales class, two methodologies were evaluated for data balancing: application of SMOTE (*Synthetic Minority Oversampling Technique*) for oversampling technique and adjustment of weights of the minority and majority classes within the cost functions of the algorithms. The Logistic Regression models both for the application of SMOTE and for adjusting the weights of the classes presented similar results, and were the ones that presented the best results. With this, the study concluded that the classification model found, through the Logistic Regression technique, is extremely relevant for the company, resulting in a reduction in the costs of contacting customers and a significant increase in the volume of return of these contacts (Sales).

Keywords: Machine Learning, Logistic Regression, Decision Tree, Customer Profile.

Sumário

Lista de Figuras

Lista de Tabelas

1. Introdução

Atualmente é extremamente comum deparar-se com volume de dados cada vez maiores, porém nem sempre os mesmo são utilizados de forma "inteligente" para extração de informações. Muitas vezes essas grandes bases de dados são utilizadas apenas para atendimentos dos requisitos dos sistemas da empresa, o que deixa de lado informações preciosas que podem ser muito úteis para traçar estratégias e tomar decisões [24]. Os grandes volumes de dados encontrados nas empresas trazem informações implícitas que podem ser de extrema valia para melhorar a relação com os clientes finais, além de otimizar a sua forma de trabalho, tornando-os mais assertivos [21].

O que auxilia uma empresa a sustentar-se e manter-se no mercado é a sua base de clientes. Uma boa e correta visão da sua base de clientes é de extrema importância para a formação de um diferencial competitivo e a possibilidade do oferecimento de um relacionamento vantajoso para ambos. Por isso, é possível notar-se que atualmente grandes investimentos estão sendo realizados em qualidade, call-centers e sistemas de atendimento ao cliente, buscando a fidelização de sua carteira, além da busca de novos clientes [20].

É notório a importância de as empresas se inovarem em relação às necessidades do mercado, sendo que o domínio tecnológico é um fator crítico desse processo, tornando a tecnologia e as informações fatores importantíssimos para a empresa [20]. Esse cenário torna o mercado cada vez mais competitivo e exige a utilização de tecnologias que estejam mais alinhados ao modelo de negócio das empresas [21].

A utilização de modelos inteligentes sobre as bases de dados podem acarretar na identificação de produtos e serviços mais adequados aos clientes e às suas necessidades. Isso é de extrema importância para as empresas, pois as tornam mais assertivas em sua atuação, fazendo que vendam para os clientes que realmente possuem uma maior possibilidade de compra, por exemplo. Isso cria uma maior retenção, além da criação de vínculos mais fortes e duradouros. [20].

Os dados são considerados extremamente valiosos no século atual, mas além de possuí-los é necessário saber como utilizá-los, como realmente transformá-los em algo valioso. Nesse âmbito que entra a Ciência dos Dados, um termo que está em alta e passou a ser muito utilizado com o surgimento de Big Data e o desenvolvimento do Aprendizado de Máquina [12].

Normalmente as bases de dados de algum sistema são modeladas apenas para fornecer ao seu usuário as informações "clássicas" que ele utilizará, como por exemplo informações consolidados de vendas e dados dos clientes. Essas informações são óbvias, e muitas vezes não agregam muito

nas tomadas de decisão. As ferramentas de mineração de dados são utilizadas para auxiliar a extrair informações que nem sempre são tão óbvias ou explícitas [24].

Essas ferramentas tem como objetivo a exploração de grandes volumes de dados, auxiliando a encontrar informações extremamente úteis e que nem sempre estão claras. Alguns métodos podem ser utilizados, como modelos de classificação, através de vários algoritmos existentes, modelos de agrupamento, análises preditivas, dentre outros. Dessa forma, as empresas que souberem utilizar os modelos de mineração de dados poderão usar suas bases de dados de forma mais lucrativa e assertiva, auxiliando a aumentar sua renda e diminuir seus gastos, além de ter uma maior produtividade [24].

Por esses e outros motivos, a mineração de dados é uma das tecnologias mais promissoras da atualidade. Um dos fatores que contribuem para esse cenário é o fato das empresas gastarem muito dinheiro com coletas de dados mas nem sempre conseguirem tirar informações úteis desses dados. Os modelos de mineração de dados, através das diversas ferramentas existentes, é algo que pode contribuir para a determinação dessas informações importantes [21].

Além da seção 2 onde são apresentados os objetivos, este trabalho é divido em outras quatro seções, Fundamentação Teórica (Seção 3), Metodologia (Seção 4), Resultados (Seção 5) e Conclusão (Seção 6). Por fim, serão apresentadas as Referências Bibliográficas utilizadas para o desenvolvimento do trabalho.

2. Objetivos

O objetivo principal desse trabalho é desenvolver um modelo de classificação, através de técnicas de mineração de dados, para a classificação do perfil dos clientes (Cliente Venda ou Cliente Não Venda) de uma empresa real.

Os objetivos específicos para o desenvolvimento do trabalho são:

- Revisar a literatura sobre mineração de dados;
- Obter e consolidar as bases de dados;
- Realizar o pré-processamento de dados;
- Implementar, validar e comparar os algoritmos de classificação;
- Demostrar que a abordagem de clientes pode ser otimizada a partir dos resultados obtidos com o algoritmo escolhido.

3. FUNDAMENTAÇÃO TEÓRICA

Nesta seção apresentaremos o embasamento teórico desse trabalho, abordando os principais conceitos dentro do aprendizado de máquina, mineração de dados e a descrição dos algoritmos a serem utilizados na metodologia.

3.1 BIG DATA

Atualmente, muito têm-se falado sobre o grande volume de dados que são gerados a cada fração de tempo. Esse grande volume de dados remetem ao famoso termo Big Data, que nada mais é do que uma forma utilizada para referir-se a grandes conjuntos de dados coletados de diversas fontes [5].

Com o passar do tempo, esse volume de dados disponíveis vem aumentando, principalmente devido ao aumento de ferramentas baseadas em dados e inteligências, além dos sistemas de Inteligência Artifical (IA) e dispositivos IoT (Internet of Things - Internet das Coisas) [5].

Conforme apresentado por Carter [5] em seu artigo, serão listadas a seguir algumas estatísticas referentes a esse grande volume de dados que são gerados:

- Cada ser humano criou cerca de 1.7 MB de dados por segundo em 2020;
- Em 2025, os especialistas indicam que mais de 463 exabytes de dados serão criados a cada dia, o equivalente a cerca de 212.765.957 DVDs;
- O mercado de análise de Big Data alcançará um valor de cerca de US $ 103 bilhões em 2027;
- 97.2% das organizações afirmam que agora estão investindo em IA e Big Data;
- 94% das empresas afirmam que os dados são essenciais para o crescimento dos negócios;
- 63% das empresas não conseguem obter insights de Big Data;
- 73% dos dados não são usados para fins analíticos;
- Empresas orientadas a dados têm 23 vezes mais probabilidade de adquirir clientes;
- 79% das empresas acham que o não uso de Big Data causará falência;
- As empresas que adotam o Big Data podem aumentar as margens operacionais em 60%.

Os números apresentados anteriormente corroboram com o aumento do volume de dados produzidos, e mostram que as empresas vem dando cada vez mais importância para essa circustância, além de mostrar que a utilização de Big Data proporciona um maior desenvolvimento para as mesmas. Porém, é preocupante perceber que um grande número de empresas ainda não conseguem obter insights e realizar análises com tantos dados disponíveis.

3.2 MINERAÇÃO DE DADOS

Com as informações apresentadas no tópico anterior uma pergunta pode surgir aos olhos de quem lê: Mas como fazer para extrair informação desse volume tão extenso de dados?

É inviável e extremamente oneroso a análise e tratativa de grandes quantidades de dados sem a utilização de ferramentas computacionais. Dessa maneira, é essencial a utilização de ferramentas específicas que auxiliem nas tarefas de análise e interpretação dos dados, além do auxílio na criação do relacionamento entre eles. Isso é fundamental para o desenvolvimento de estratégias em todas as áreas de aplicação [14].

Nesse sentido, surge uma área denominada Descoberta de Conhecimento em Bases de Dados (*Knowledge Discovery in Databases*– KDD). O termo Mineração de Dados (Data Mining) é uma das etapas dentro dessa área [14].

A Figura 3.1 apresenta uma representação visual do processo KDD:

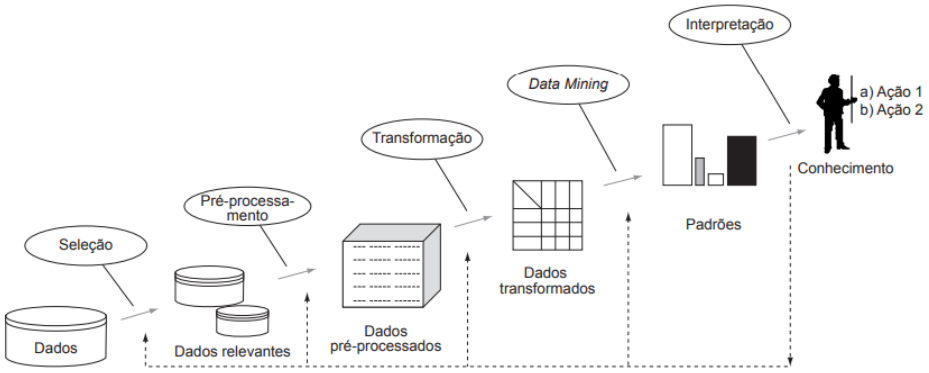


Figura 3.1: Etapas do processo KDD.

Fonte: [8] adaptado por [30].

O conceito do KDD surgiu em 1989 devido a grande necessidade de extrair informações e conhecimento de grandes bases de dados [8].

Não perceber e/ou interpretar de forma inadequada vários fatos observáveis durante o processo KDD, são fatos que o tornam mais complexo, além de conjugar dinamicamente tais interpretações de forma a decidir que ações devem ser realizadas em cada caso. Esses fatos torna imprescindível que um analista de dados, em conjunto com um especialista do negócio, orientem a execução desse processo [14].

A Mineração de Dados é o processo de análise de grandes conjuntos de dados com o objetivo de encontrar padrões interessantes e que representam informações úteis e verídica [20]. Além de padrões, esse processo auxilia também a identificar anomalias e correlações nesses conjuntos, contribuindo para previsibilidade de resultados. Através de diversas técnicas, essas informações podem ser utilizadas para aumentar a renda, diminuir custos, melhorar o relacionamento com os clientes e reduzir riscos [23].

Por definição, as ferramentas de Mineração de Dados devem trabalhar com grandes volumes de dados, e retornar novos conhecimentos, e que sejam relevantes. Porém, é de extrema

importância que sejam realizadas avaliações criteriosas sobre os resultados, principalmente pelas diversas relação existes e pelas equiações geradas, fatos que podem tornar impossível o processamento dos dados [20].

Pelo fato de ser na etapa de Mineração de Dados que ocorre a busca efetiva por conhecimentos novos e úteis, ela é considerada como a principal etapa do processo de KDD. Por este motivo, diversos autores tratam a Mineração de Dados e o KDD como um único processo, de maneira que fossem sinônimos [14].

O termo "mineração" só foi efetivado de fato nos anos 1990, mas sua base compreende três disciplinas científicas entrelaçadas que existem há tempos: estatística (o estudo numérico das relações entre dados), inteligência artificial (inteligência exibida por softwares e/ou máquinas, que se assemelha à humana) e aprendizado de máquina (algoritmos que podem aprender com dados para realizar previsões) (Figura 3.2). Além disso, o processo de Mineração de Dados continua em constante evolução, visando acompanhar o potencial ilimitado do Big Data e da computação de baixo custo [23].

Figura 3.2: Blocos de Formação da Mineração de Dados

Fonte: [23], adaptado pelo Autor, 2023.

Dentro desse referencial teórico não serão abordados conceitos e nem técnicas específicas para os demais pilares do processo KDD: Seleção, Pré-Processamento e Transformação dos dados, uma vez que na metodologia algumas técnicas serão mais bem descritas. Porém, é imprescindível ressaltar a importância dessas etapas, uma vez que elas garantirão que se obterá dados confiáveis e corretos, resultando em uma melhor execução do processo de Mineração de Dados, além da obtenção de resultados mais expressivos e confiáveis.

A seguir, será apresentado, de forma resumida, os principais conceitos dentro das três áreas do conhecimento listados na Figura 3.2, esses responsáveis pela formação da Mineração de Dados. Cabe ressaltar que esses conceitos não são aplicados apenas na etapa de Mineração de Dados, mas em todo o processo KDD.

3.2.1 ESTATÍSTICA

A mais antiga das linhagens das quais a Mineração de Dados descende é a estatística clássica. Sem ela não seria possível termos os processos de Mineração, uma vez que é a base da maioria das tecnologias a partir das quais esses processos são desenvolvidos [16].

A Estatística é a área responsável por envolver diversos conceitos, como distribuição probabilística, média, mediana, moda, variância, covariância, assimetria, curtose, correlação, análise de resíduos, análise de conjuntos e intervalos de confiança, todos utilizados nos estudos dos dados e nos seus relacionamentos. São nestes elementos que as análises mais avançadas se apoiam. Sendo assim, no interior das técnicas e atuais ferramentas de Mineração de Dados, é de fundamental importância o papel desempenhado pela análise estatística clássica [7].

3.2.2 Inteligência Artificial

A segunda linhagem da Mineração de Dados é a Inteligência Artificial (IA). Essa disciplina, que, ao contráro da estatística, é construída a partir dos fundamentos da heurística, tenta reproduzir a maneira como o ser humano pensa para resolver os problemas estatísticos. Em função dessa abordagem, ela requer um incrível poder de processamento, que até os anos 80 seria impraticável, que foi quando os computadores começaram a oferecer uma melhor capacidade de processamento a custos mais acessíveis [7].

A IA desenvolveu algumas aplicações para o alto escalão do governo/cientistas americanos, sendo que os custos elevados não permitiram que ela fosse acessível a todos. As consideráveis exceções foram alguns conceitos de IA adotados por produtos de ponta, como módulos de otimização de consultas para SGBD - Sistemas de Gerenciamento de Banco de Dados[7].

Mesmo assim, o custo/benefício das atuais aplicações faz com que essa metodologia se torne acessível para uma grande maioria das empresas, principalmente as Brasileiras. Atualmente, pequenas e médias empresas já dispõem de softwares de IA para realização de diversas análises, principalmente a respeito do descobrimento do perfil de clientes inseridos em base de dados e/ou sistemas de relacionamento [7].

3.2.3 APRENDIZADO DE MÁQUINA

A terceira linhagem da Mineração de Dados é o chamado Aprendizado de Máquina - AM (ou aprendizado de máquina - ML), que formado pela junção das melhores práticas de estatística e Inteligência Artificial [7].

As ténicas da IA foram sendo agregadas pelo Aprendizado de Máquina enquando ela não se tranformava em um sucesso comercial. O aprendizado de Máquina foi capaz de se valer das sempre crescentes taxas de custo/desempenho oferecidas pelos computadores nos anos 80 e 90, sendo capaz de ampliar a quantidade de aplicações devido às suas combinações entre a análise estatística e a heurística [7].

A base do Aprendizado de Máquina é fazer com que os programas de computador "aprendam" com os dados que são estudados, tal que esses programas tenham capacidade de tomar decisões diferentes baseadas nas características dos dados estudados. Espera-se atingir esse objetivo utilizando a estatística com seus conceitos fundamentais, além de adicionar mais heurística avançada da IA e algoritmos [7].

Desta maneira, pode-se definir que a mineração de dados é basicamente a adaptação das técnicas de Aprendizado de Máquina voltadas para os negócios. Com isso, é possível descrevê-la como a união dos históricos e dos recentes desenvolvimentos em estatística, em IA e Aprendizado de Máquina. O principal objetivo é utilizar essas ténicas, de forma conjunta, para analisar os dados e buscar encontrar tendências e padrões. Nos dias atuais, a mineração de dados tem experimentado uma crescente aceitação nas ciências e nos negócios, principalmente as que necessitam de análises com grandes volumes de dados, além da necessidade de encontrar tendências que não poderiam ser avaliadas de outra maneira [7].

Por existir uma infinidade de métodos para agrupá-los, não é uma tarefa tão fácil classificar os algoritmos de Aprendizado de Máquina. A forma mais básica, e mais utilizada normalmente, é classificá-los de acordo como eles aprendem. Utilizando esse formato, eles podem ser classificados em Aprendizado Supervisionado e Aprendizado Não Supervisionado [**?**].

1. **Aprendizado Supervisionado**

   A aprendizagem supervisionada é aquela onde as bases de dados utilizadas possuem informações sobre a classificação dos mesmos [15]. Essas classificações normalmente são chamadas de rótulos, e podem ser como por exemplo: 'sim' ou 'não, 'doente' ou 'saudável', 'venda' ou 'não venda', entre outras infinidades. Nesse tipo de análise, os dados utilizados como treino para o algorítmo estarão com os rótulos presente na base, juntamente com os demais atributos/caraterísticas. Durante o treino, o algoritmo procura padrões nos dados que se correlacionem com as saídas desejadas. Após essa etapa de treino, o algoritmo receberá os demais dados (dados de teste) e de acordo com suas análises atribuirá um rótulo a eles, baseando-se nos padrões dos dados de treino. O objetivo deste modelo de aprendizagem é prever a classificação correta para um novo conjunto de dados. Normalmente é utilizado quando possui-se dados históricos e tem como objetivo prever resultados futuros. Assim, é possível realizar a comparação dos valores atribuídos

pelo algorítmo com os rótulos reais [11].

Em algumas vezes, os padrões identificados no conjunto de treino podem não ser detectados quando avaliados no novo conjunto de dados. Nessas situações, o modelo está adequado para representar os padrões que existem apenas na base treino, ocasionando o que é chamado de overfitting. Isso quer dizer que o modelo está demasiadamente "afinado" para os dados de treino, não permitindo que seja aplicado a outros conjuntos desconhecidos [15].

Essa categoria de aprendizado engloba duas subcategorias: Classificação e Regressão.

(a) **Classificação:** Esses algoritmos buscam a identificação de uma categoria/rótulo através de uma observação dada. Neste caso, é estimado um "classificador" que gere como saída um resultado qualitativo, com base nos dados de entrada (que abrangem observações com classificações já definidas). Pode-se tomar como exemplo um modelo que utilize dados de um paciente e classifique-o como doente ou saudável [32].

(b) **Regressão:** de forma similar a classificação, esses algoritmos utilizam dados de entrada (preditores) já observados com o objetivo de prever uma resposta. A principal diferença é que, nestes modelos, procura-se estimar um valor/rótulo numérico e não uma classificação de uma observação. Pode-se tomar como exemplo um modelo que busque prever o salário de um indívido através de suas informações pessoais, como idade e anos de escolaridade [32].

3.3 SEPARAÇÃO DA BASE EM TREINO E TESTE NOS MODELOS DE CLASSIFICAÇÃO

Como garantia de que o modelo de classificação possua uma precisão aceitável, é necessário a realização de um teste adequado para saber se o modelo é capaz de prever corretamente o rótulo de novos dados. É de extrema importância que esses dados utilizados no teste do modelo não sejam os mesmos utilizados na "criação" (treinamento) do algoritmo, pois nessas situações ele poderia simplesmente replicar o rótulo já identificado anteriormente, prevendo sempre um resultado correto [10].

Para isso, é de praxe dividir o conjunto de dados iniciais em dois grupos: um grupo de treinamento e outro grupo de teste[10]. Essa divisão em treino e teste é realizada para verificar se um modelo apresenta bom desempenho não apenas nos dados utilizados inicialmente para seu ajuste (treinamento), mas também se possui capacidade de generalização para novas observações (teste). Normalmente, as divisões mais utilizadas são 60:40%, 70:30% ou 80:20%, o que varia de acordo com o tamanho do conjunto de dados. A maior proporção normalmente é utilizada no grupo de treino. Em geral, quanto mais observações, maior será a proporção utilizada para o grupo de treino [22].

De maneira geral, pode-se escolher aleatoriamente em torno de 20-30% dos dados como parte do grupo de teste e o restante ser utilizado como grupo de treinamento. Como esta divisão é feita de forma aleatória, as medidas de desempenho utilizadas para avaliar o algoritmo podem apresentar-se de maneira diferente dependendo de quais dados forem selecionados para cada grupo, especialmente se o conjunto de dados inicial for consideravalmente pequeno [10].

3.4 Medidas de Avaliação

Nos modelos de classificação não existe um algoritmo que seja ideal ou perfeito. Com isso, na construção do modelo, é necessário a avaliação de qual algoritmo e parâmetros são melhores para o preditor [19]. Em seguida, serão apresentadas formas de avaliar a qualidade de um modelo.

Denomina-se problema de classificação binário quando o modelo apresenta apenas duas possíveis classes. Um exemplo usual são os modelos para detecção de *spam*. Atualmente, os serviços de e-mail possuem avançados sistemas para detectar possíveis mensagens indeseajdas, que são chamadas de *spams*. Esse é um clássico problema de classificação binária. Com isso, cada mensagem pode ser classificada de duas formas: positivo (se for um *spam*) ou negativo (se não for um *spam*) [19].

Segundo [19], em problemas de classificação binária, as classes positivo e negativo podem, ainda, adotar quatro variações:

- **Verdadeiro positivo (VP):** quando o dado original corresponde a um item positivo e a predição foi realizada corretamente (por exemplo, uma mensagem de *spam* predita como *spam*);
- **Verdadeiro negativo (VN):** quando um item classificado como negativo é corretamente predito como negativo (por exemplo, uma mensagem que NÃO é *spam* predita como NÃO sendo *spam*);
- **Falso positivo (FP):** quando um item classificado como negativo é incorretamente predito como positivo (por exemplo, uma mensagem que NÃO é *spam* predita como *spam*);
- **Falso negativo (FN):** quando um item classificado como positivo é incorretamente predito como negativo (por exemplo, um *spam* predito como NÃO sendo *spam*).

Na Tabela 3.1 é possível ver a resresentção de uma matriz de confusão, que é uma das maneiras mais simples de se representar os resultados citados anteriormente, referente aos algoritmos de classificação [19].

Tabela 3.1: Representação de uma matriz de confusão.

Matriz de Confusão		**Classe Predita**	
		Positiva	**Negativa**
Classe Original	**Positiva**	VP	FN
	Negativa	FP	VN

Fonte: [19], adaptado pelo Autor, 2023.

A partir desta matriz, é possível calcular diversas métricas para a avaliação de modelos. Abaixo serão apresentadas as principais métricas utilizadas.

A **acurácia**, também conhecida como ACC, é mais utilizada métrica [19]. Seu cálculo é dado por:

$$acurcia = \frac{VP + VN}{VP + FN + VN + FP} \tag{3.1}$$

Uma outra forma de representar a acurácia é como a razão entre a quantidade de acertos e o total de elementos avaliados:

$$acurcia = \frac{total\ de\ acertos}{total\ de\ itens} \tag{3.2}$$

A **sensibilidade** ou ***recall***, calcula a probabilidade de se encontrar resultados positivos. Ela também pode ser chamada de recall ou revocação [19]:

$$sensibilidade = \frac{VP}{VP + FN} \tag{3.3}$$

Ao contrário da sensibilidade, a medida especificidade calcula a probabilidade de se obter resultados negativos [19], e é dada pela equação:

$$especificidade = \frac{VN}{VN + FP} \tag{3.4}$$

Outra medida existente é a **precisão**. Ela também foca na análise de resultados positivos. O que ela difere da fórmula da sensibilidade é apenas por usar falsos positivos no lugar dos falsos negativos [19]. Veja:

$$precisão = \frac{VP}{VP + FP} \tag{3.5}$$

Por último, tem-se a medida conhecida como F-measure ou F1-score. Ela é uma métrica que leva em consideração tanto a precisão quanto a sensibilidade. Ela é calculada pela **média harmônica** [19], e é dada por:

$$f1 = 2 \times \frac{precisão \times sensibilidade}{precisão + sensibilidade} \tag{3.6}$$

Por mais que o uso de tantas medidas para avaliação do desempenho de um modelo soe um pouco controverso, é importante ressaltar que sua utilidade vai depender do modelo adotado. Evidentemente não existe uma medida ideal para ser utilizada em todos os modelos. Por exemplo, a acurácia trará uma visão geral de quão bom o modelo é, mas, dependendo da situação, pode ser mais relevante realizar a análise de falsos negativos e falsos positivos [19].

A utilização da acurácia em modelos que trabalhem com dados desbalanceados não é apropriada, uma vez que grandes números de verdadeiros positivos poderão acobertar grandes números de falsos positivos e vice-versa, de modo um classificador que "acerta" muito bem amostras da classe com maior ocorrência e rotula de maneira mediana/ruim amostras da classe de menor ocorrência ainda assim apresentará um bom valor de acurácia [18].

A utilização de modelos criados a partir de dados desbalanceados exigem a utilização de medidas de avaliação que estejam mais atentas às classes com menor ocorrência. Dentre as

possíveis medidas a serem utilizadas, uma prática recomendada para estes cenários é a utilização de gráficos ROC (apresentado na seção 2), que apresentam a relação entre sensitividade e falsos positivos obtidos por um classificador quando vários limites para a sua função de decisão são investigados. Através das curvas projetadas neste gráfico, é possível encontrar para um determinado modelo de classificação quais são os custos-benefícios entre verdadeiros e falsos positivos obtidos, o que facilita encontrar a habilidade do classificador de determinar corretamente amostras de uma classe em relação às outras [18].

O cálculo da área abaixo da curva (AUC), resulta em um valor escalar no intervalo entre 0 e 1, e determina à probabilidade do classificador de classificar uma amostra positiva melhor do que uma amostra negativa, e este valor pode ser utilizado como medida de avaliação do modelo [18]. Quanto maior o valor de AUC encontrado para um modelo, melhor é considerado o seu desempenho.

3.5 TRATAMENTO PARA BASES DESBALANCEADAS

Grande parte dos modelos de classificação existentes, e que apresentam satisfatórios desempenhos de predição, são desenvolvidos para trabalhar com conjunto de dados balanceados. Pode-se definir como um conjunto de dados balanceado aqueles em que a proporção de ocorrência das diferentes classes do rótulo é igual em todas essas classes, ou com uma pequena diferença, de modo que todas as classes estejam representadas igualmente por suas distribuições [18].

Na Tabela 3.2, é possível notar que a base da esquerda é um exemplo de conjunto perfeitamente balanceado, uma vez que a classe A e a classe B ocorrem com a mesma frequência. Já a base da direita é um exemplo de conjunto de dados desbalanceado, uma vez que a classe A aparece em uma proporção consideravelmente maior que o da classe B [18].

Tabela 3.2: Exemplos de conjunto de dados balanceado esquerda e outro conjunto de dados desbalanceado direita.

Var1	Var2	Var3	Var4	Classe
8,05	1	15,15	97,85	A
12,12	1	30,21	96,84	A
36,1	2	32,68	95,14	A
24,89	1	18,13	98,27	A
9,36	3	12,79	97,39	B
15,42	2	31,43	99,1	B
11,05	3	39,24	96,28	B
10,74	2	10,54	95,99	B

Var1	Var2	Var3	Var4	Classe
8,05	1	15,15	97,85	A
12,12	1	30,21	96,84	A
36,1	2	32,68	95,14	A
24,89	1	18,13	98,27	A
9,36	3	12,79	97,39	B
15,42	2	31,43	99,1	B

Fonte: [18], adaptado pelo Autor, 2023.

Mesmo que a maioria dos modelos sejam desenvolvidos para lidar com esses dados balanceados, na prática o mais provável é que os dados disponíveis para utilização estejam desbalanceados [18].

Em muitos casos, a classe minoritária é a classe de interesse, o que torna o problema do desequilíbrio de classe ainda mais expressivo. Por exemplo, em um conjunto de dados de detecção de fraudes em transações financeiras, a classe minoritária pode ser a classe de transações fraudulentas. Nesse caso, é importante que o modelo seja capaz de identificar corretamente as transações fraudulentas, caso contrário, a detecção de fraudes pode ser ineficaz.

Dentro da fase de pré-processamento de dados algumas técnicas podem ser utilizadas com o objetivo de balancear a distribuição das classes no conjunto de dados de treino. Esses mecanismos podem ser a subamostragem da classe com maior ocorrência, sobreamostragem da classe com menor ocorrência ou uma combinação dessas duas técnicas [2].

Técnicas de subamostragem (*undersampling*) são as que removem os elementos da classe com maior ocorrência a fim de promover o balanceamento dos dados. Os exemplos a serem excluídos podem ser determinados de maneira aleatório (subamostragem aleatória) ou por meio de algum critério de seleção (subamostragem informativa) [2].

Na subamostragem aleatória, basicamente os elementos da classe majoritária são selecionados aleatoriamente e assim removidos do conjunto de treinamento. A quantidade de dados escolhidos pode variar, mas normalmente ela é suficientemente grande para que tenha-se no final a mesma quantidade de dados para todas as classes [2].

Como exemplo de subamostragem informativa têm-se a técnica OSS (*One-sided Selection*), que cria um novo conjunto de dados formado por todos os exemplos da classe com menor ocorrência e os exemplos mais relevantes da classe com maior ocorrência. Para definir os exemplos mais relevantes da classe com maior ocorrência, seleciona-se aleatoriamente apenas um elemento dessa classe. A partir desse único elemento, juntamente com todo o conjunto minoritário, tenta-se classificar todos os exemplos conhecidos. Os exemplos que forem classificados de forma correta serão considerados redundantes e então retirados do conjunto de treinamento. Dessa forma, o exemplo escolhido aleatoriamente, juntamente com os elementos classificados de forma incorreta, serão definidos como os mais relevantes da classe majoritária. Além disso, são utilizadas técnicas de limpeza dos dados com o intuito de remover os exemplos de borda ou ruidosos [2].

Já as técnicas de sobreamostragem (oversampling) são aquelas em que novos exemplos da classe com menor ocorrência são adicionados à base de dados. Essas técnicas podem repetir representantes dessa classe (sobreamostragem com repetição), de forma aleatória ou informativa, ou então gerar novos dados de forma artificial [2].

Para o processo de sobreamostragem aleatória, um subconjunto de elementos da classe de menor ocorrência é selecionado de maneira aleatória, ou esse conjunto inteiro é selecionado, e então ele é replicado para a base de treino. Esse processo é repetido até se obter a proporção de exemplos desejados, que normalmente é a quantidade que faça com a classe minotária fique com a mesma quantidade de observações da classe majoritária [2].

SMOTE (*Synthetic Minority Oversampling Technique*) é uma técnica existente que cria dados de maneira artificial através de interpolação. Essa técnica utiliza o algoritmo KNN para criar representantes entre exemplos e seus k vizinhos. Para isso, a cada iteração do algoritmo um exemplo da classe com menor ocorrência é escolhido, e então o vetor obtido pela diferença entre o elemento analisado e seu vizinho é multiplicado por um número aleatório entre 0 e 1. O vetor resultante desse processo é aplicado sobre o elemento analisado, selecionando um ponto no espaço, que fica na reta entre o elemento e seu vizinho. Neste ponto, o novo elemento da classe minoritária é criado [2].

Para exemplificar o processo de criação de amostras do SMOTE, [6] apresentou um breve exemplo. Considere uma amostra com dados observados $(6, 4)$. Dentre os k vizinhos mais próximos desta amostra foi selecionado de forma aleatória o par $(4, 3)$. Com isso, para cada elemento do vizinho escolhido será feita a subtração com o elemento, na mesma posição, da amostra observada. Para o exemplo adotado, o resultado da subtração será $(4, 3) - (6, 4) = (4 - 6, 3 - 4) = (-2, -1)$. O próximo passo é a escolha de um número aleatório entre 0 e 1. Por fim, a nova amostra será obtida realizando-se a soma da amostra observada com a multiplicação do número sorteado entre 0 e 1 com a subtração realizada, ou seja, denotando

este número aleatório por u, a nova observação será dada por $(6,4) + u \times (-2,-1)$. Esse resultado será inserido no conjunto de dados original adotando como rótulo a classe com menor número de observações.

Outra alternativa para lidar com os dados desbalanceados, em vez de técnicas de reamostragem, é levar em conta os custos dos erros de classificação no treinamento de um modelo de aprendizado de máquina. Por exemplo, pode-se adotar diferentes formas de penalização do modelo, dependendo se o erro de classificação incorreta for na classe de menor ocorrência ou na de maior ocorrência. Esses custos do erro são calculados pela função de erro, que leva em conta os resultados produzidos pelo modelo de classificação e os resultados reais do problema que se busca modelar.

Alguns algoritmos utilizados no treinamento dos modelos podem ser ajustados para levar em consideração o desbalanceamento das classes. A maneira de realizar esse processo é atribuir pesos diferentes para as classes majoritárias e minoritárias nas funções de custo. Essa diferença interferirá na classificação do modelo durante o treino. O objetivo desse ajuste é penalizar a classificação incorreta feita pela classe de menor ocorrência, estabelecendo um peso maior para essas classes, ao mesmo tempo que se reduz o peso da classe majoritária, fazendo com que o modelo possa se concentrar em reduzir os erros para a classe de menor ocorrência [27].

Grande parte das bibliotecas de modelagem do classificador *sklearn* no *Python* e até mesmo algumas bibliotecas baseadas em *boosting* como *LightGBM* e *catboost* possuem um parâmetro interno denominado "class_weight" que permite a otimização dos pesos para a classe de menor ocorrência [27].

Por padrão, o valor desse parâmetro class_weight já vem definido como None, ou seja, todas as classes recebem o mesmo peso. Porém, é possível definir esse parâmetro como "balanced" ou criar um dicionário que contenha pesos manuais específicos para cada uma das classes [27].

Quando class_weights recebe o valor "balanced", o algoritmo atribui de forma automática os pesos de classe de forma inversamente proporcional às suas frequências [27].

Para que fique mais claro, a fórmula para calcular é dada por [27] e é apresentada na expressão 3.7

$$w_j = n/(c \times c_j), \tag{3.7}$$

em que:

- w_j é o peso de cada classe j com $j = 1, 2, \cdots c$;
- n é o número total de amostras ou linhas no conjunto de dados;
- c é o número total de classes exclusivas;
- c_j é o número total de linhas da classe j.

No caso da regressão logística, um dos modelos utilizados nesse trabalho, a função perda logarítmica é normalmente utilizada como a função de custo. Não se utiliza o erro quadrático

médio como função de custo para esse modelo, uma vez que em vez de ajustar uma linha reta, a curva sigmóide que é utilizada como função de previsão. O quadrado da função sigmóide retornará uma curva que não é convexa, e com isso a função de custo poderá ter muitos mínimos locais e, com isso, a convergência para os mínimos globais se torna mais complicado. Porém, utilizando-se a perda logarítmica o resultado será uma função convexa, e com isso tem-se apenas um mínimo para convergir [27].

A função perda logarítmica [27] é apresentada na expressão 3.8,

$$LogLoss(y,p) = -\frac{1}{N}\sum_{i=1}^{N}(y_i log(p_i) + (1-y_i) log(1-p_i)), \tag{3.8}$$

onde:

- N é o número de amostras;
- y é o valor observado do rótulo (0 ou 1);
- p é a probabilidade prevista da classe alvo (entre 0 e 1).

Após acrescentar-se os pesos à função de custo, a função de perda logarítmica modificada é [27]:

$$LogLoss(y,p) = -\frac{1}{N}\sum_{i=1}^{N}(w_0(y_i log(p_i)) + w_1((1-y_i) log(1-p_i))), \tag{3.9}$$

onde:

- w_0 é o peso da classe para a classe 0;
- w_1 é o peso da classe para a classe 1.

Já para as árvores de decisão, que também é um dos algoritmos utilizados para desenvolvimento desse trabalho, em cada um dos nós ocorre uma divisão dos dados, buscando-se atingir grupos que sejam o mais puros possíveis (menos misturados) [17].

Nesse cenário, a pureza representa uma divisão limpa dos dados em grupos onde um grupo de dados possuindo apenas valores 0 ou apenas valores 1 (dados de apenas uma classe) é o mais puro possível, e uma mistura com 50% de cada classe é o cenário menos puro. Normalmente, a pureza é calculada atráves do Índice de Gini, embora também possa ser calculada usando a entropia [17]. Mais informações sobre o cálculos dessas medidas e das divisões dos nós na árvore de decisão é apresentado na subseção 3.6.1.

O cálculo de uma medida de pureza relaciona probabilidade de um exemplo de uma classe específica ser classificado erroneamente durante a divisão. Esses cálculos envolvem a soma do número de observações em cada uma das classe dentro de cada grupo.Visando o ponderamento dos dados desbalanceados, o critério utilizado para a divisão pode ser ajustado para não levar em consideração apenas a pureza da divisão, mas também o peso calculado para cada classe (mesma fórmula apresentada em 3.7) [17].

A forma de se obter esses valores ajustados é obtida substituindo-se a contagem de observações em cada grupo pela soma dos pesos representados por aquele grupo [17].

No caso da entropia, ao se calcular a probabilidade de uma tupla em S pertencer a classe C_i (Fórmula 3.7), ao invés de dividir o total de casos pertencentes à classe C_i em relação à S, deve-se dividir a soma dos pesos dos casos pertencentes à classe C_i pela soma dos pesos das observações de S [17].

Para exemplificar, considere um conjunto de dados composto por duas variáveis dependentes X0 e X1 e uma variável resposta Y (Rótulo).

Os valores dos índices de Gini e a construção da árvore de decisão para esse modelo é apresentado na Figura 3.3.

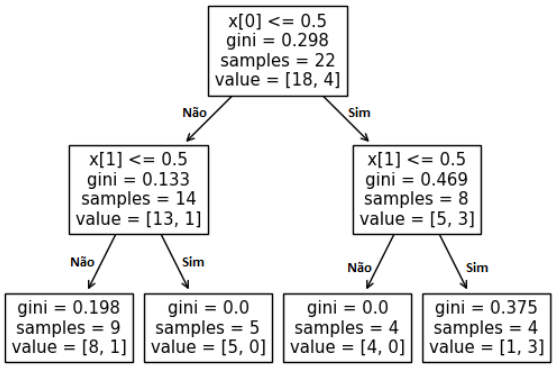


Figura 3.3: Árvore de Decisão para modelo sem ajuste do class_weight.
Fonte: Elaborado pelo Autor, 2023.

Adotando-se o parâmetro class_weight como "balanced", obtem-se uma nova árvore de decisão com mudança nos valores dos índices de Gini, conforme apresentado na Figura 3.4.

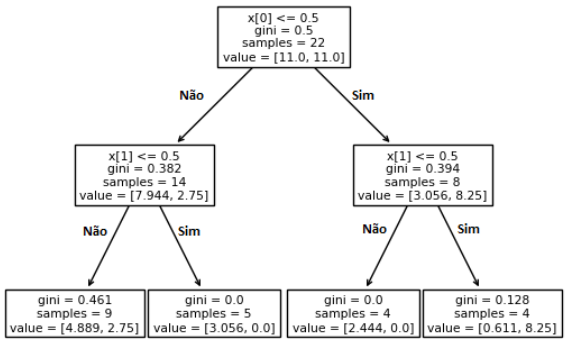


Figura 3.4: Árvore de Decisão para modelo com ajuste do class_weight.
Fonte: Elaborado pelo Autor, 2023.

3.6 Algoritmos para Modelos de Classificação

Em um projeto de mineração de dados, com foco definido em um poblema Classificação e Predição, utilizar diferentes abordagens e algoritmos é fundamental para um processo de avaliação. Serão avaliados não somente a precisão do modelo, mas também a velocidade de processamento, a robustez, a escalabilidade e a interpretabilidade [20].

Dependendo da estrutura de predição e do modelo utilizado em um cenário de aprendizado supervisionado, alguns algoritmos e ferramentas podem ser mais adequados ao problema [20]. Abaixo serão apresentados os principais conceitos dos Algoritmos Árvore de Decisão e Regressão Logística, que são os que serão adotados na metodologia deste trabalho.

3.6.1 Árvore de Decisão

Pode-se considerar que os algoritmos de árvores de decisão surgiram através da evolução das técnicas que apareceram durante o desenvolvimento do aprendizado de máquina. Esses modelos trabalham testando de forma automática todos os valores possíveis de um atributo, para conseguir identificar quais os que estão mais associados aos itens de saída selecionados. Os valores com forte associação são os fatores explicativos ou pressupostos chaves, que normalmente são chamados de regras sobre o dado [7].

Esses algoritmos são utilizados para representar resultados de Mineração de Dados em formato de árvore, lembrando um gráfico organizacional horizontal. A partir de um conjunto de dados com diversas colunas e linhas, um algoritmo de árvore de decisão solicita que o usuário escolha uma das colunas como sendo o objeto de saída para ser o primeiro ramo (nó) da árvore. Os demais atributos (demais colunas) são posteriormnte classificados como nós dos nós anteriores. Com isso é possível que o usuário veja qual o atributo que mais direciona o seu objeto de saída, entendendo o motivo da escolha [7].

As árvores de decisão são formadas por nós que representam os atributos da base de dados, por ramos que derivam desses nós e recebem os possíveis valores do atributo representado pelo nó. Finalmente, as árvores possuem também nós folha, que são os últimos possíveis nós, e representam os valores do atributo rótulo no conjunto de treinamento [20].

A principal função de uma árvore de decisão é repartir, recursivamente, a base de treinamento, até que cada subconjunto derivado dos particionamentos contenham apenas casos de um único rótulo [13]. Para chegar nesse ponto, o algoritmo examinará e comparará a distribuição das classes durante a construção do modelo. Os resultados provenientes da construção da árvore são dados organizados de maneira compacta, e são utilizados posteriormente para classificar novos casos [20]. A Figura 3.5 apresenta um exemplo de árvore de decisão:

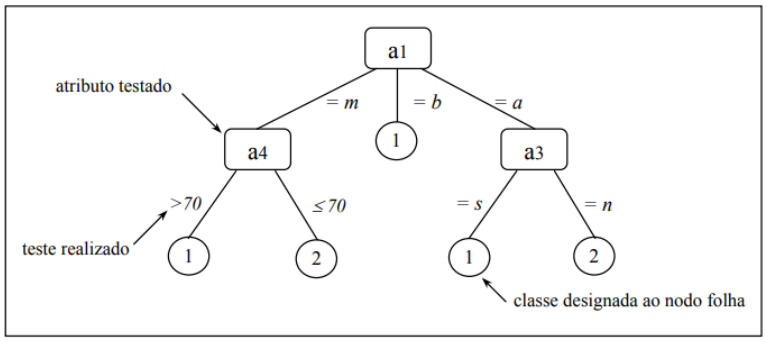


Figura 3.5: Exemplo de um classificador utilizando árvore de decisão

Fonte: [13].

Na Figura 3.5, os atributos a1, a3 e a4 são quem representam os nós. Eles estão dispostos na árvore de acordo com seu nível informativo. Os possíveis valores do nó são testados pelos ramos e são representados saindo dele. Esses testes são realizados através dos valores dos atributos. Quando esses valores são categóricos, o teste será uma igualdade, já quando os valores forem quantitativos, o teste será representado por um intervalo de valor, como por exemplo > 70. O rótulo associado aos nós folha são representados pelo círculo ao final dos ramos. No exemplo da Figura 3.5 o valor 1 representa uma classe positiva, enquanto o valor 2 uma classe negativa [13].

Para realizar a classificação em um modelo de árvore de decisão, deve-se percorrer o caminho que se inicia no nó raiz (a1), se estendendo até as folhas [13]. A situação é mostrada na Figura 3.6:

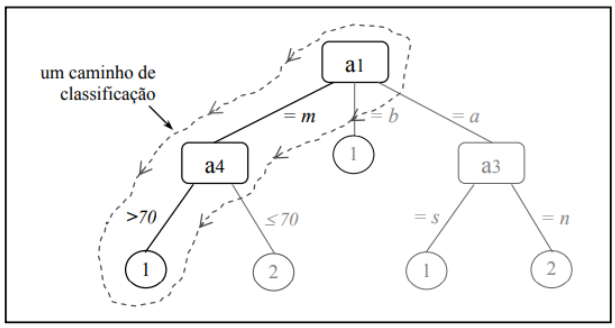


Figura 3.6: Exemplo de um classificador utilizando árvore de decisão

Fonte: [13].

O caminho exemplificado na Figura 3.6 representa uma classificação, onde o caso analisado se equadra na situação grifada, ou seja, o valor do atributo a1 é igual a m e o valor do atributo a4 é maior que 70 [13].

A partir dos possíveis caminhos de uma árvore de decisão é possível a criação de regras. Geralmente, as árvores são utilizadas em conjunto com as regras. Pelo fato das árvores poderem crescer muito, muitas vezes as regras as substituem. Esse fato ocorre pelo motivo das regras poderem ser modularizadas de uma forma mais fácil [13].

A regra extraída do caminho destacado na árvore da Figura 3.6, pode ser descrita como: "Se a1 = m e se a4 > 70 então 1".

Os modelos de árvores de decisão podem ser elaborados por diferentes algoritmos de classificação existentes. Não existe uma regra de quais parâmetros utilizar para se obter melhores desempenhos, o problema em análise quem ditará qual agoritmo poderá apresentar melhor desempenho [20].

Segundo [20], a formação de uma árvore de decisão segue os seguintes passos:

1. associar a partição do nó-raiz ao espaço de objetos;
2. verificar se o nó atual é um nó folha checando se pelo menos um dos seguintes quesitos é verdadeiro:
   - todos os objetos contidos na partição do nó atual são da mesma classe;
   - todos os atributos de objetos já foram utilizados no teste de algum nó no caminho deste até a raiz;
   - a quantidade de objetos na partição do nó atual é inferior ao limite estabelecido (o limite mínimo é 1);
   - no caso do nó atual ser uma folha, encerrar a exploração deste.
3. dividir a partição do nó atual segundo um atributo que não foi utilizado em nenhum outro teste sobre atributo no caminho entre o nó atual e o nó raiz;
4. aplicar recursivamente o passo 2 e 3 do algoritmo para cada nó filho do nó atual.

A avaliação de um modelo de árvore de decisão é extremamente importante após a sua construção. Para a realização da avaliação é necessário utilizar os dados que não foram usados no treinamento, ou seja, os dados de Teste. Através desta estratégia é possível estimar como a árvore faz a generalização dos dados além de como a mesma se adapta a novas situação. A partir disto é possível estimar também a proporção de acertos e erros que ocorreram na construção do modelo [20].

Para que um modelo de árvore de decisão consiga ser ótimo em relação a altura da árvore, eficiência de classificação e tempo de contrução, algumas questões precisam ser superadas. [20] lista algumas destas questões, que ainda são temas de pesquisa até hoje, como por exemplo:

- escolha da melhor partição para um nó, em geral, por escolha do atributo;
- estratégias para limitação no crescimento da árvore;
- tratamento de valores desconhecidos no conjunto objetos para treino e para teste;
- partições baseadas em características discretas e contínuas.

Para definir se uma árvore de decisão é considerada ótima seria necessário testar todos os caminhos possíveis, porém, isso poderia ser inviável computacionalmente. Com isso, os algoritmos buscam escolhas de locais ótimos para determinar a melhor sequência de atributos e a regra com maior pureza [1].

As medidas de impureza, como por exemplo Entropia e Índice Gini, auxiliam a determinar o melhor ponto de corte para determinado nó, selecionando o que possuir maior ganho de informação entre todos os possíveis candidatos [1].

1. **Entropia**

   Nos modelos de árvores de decisão, a entropia é uma forma de mensurar a impureza em cada conjunto. É um conceito matemático que mensura a aleatoriedade ou incerteza. O nível de entropia pode variar entre 0 e 1, significando que quanto mais perto do zero, mais puro é o conjunto (possui menos aleatoriedade), e quanto mais próximo de 1, possui maior aleatoriedade (vários elementos distintos) [1].

   Dado um conjunto S de s amostras com m classes distintas $C_i, i = 1, ..., m$, a entropia pode ser calculada por:

$$Entropia(S) = -\sum_{i=1}^{m} p_i log_2(p_i) \tag{3.10}$$

   onde p_i representa a probabilidade de um elemento pertencer a clase C_i, que é calculada pela proporção s_i/s, sendo s_i o total de elementos pertencentes à classe C_i e S o total de elementos [1].

2. **Ganho de Informação**

   Após o cálculo da entropia do conjunto, é realizado o cálculo do ganho de informação para determinar o melhor atributo para o nó, que é feito através da diferença entre a entropia antes da divisão e a entropia média após a divisão do conjunto. O Ganho de Informação representa a quantidade de informação que se espera ao realizar a divisão. Esse cálculo é feito para cada atributo em cada nó da árvore, e o atribuito que possuir maior ganho de informação (ou menor entropia) é escolhido [1].

   Matematicamente, o ganho de informação é dado por:

$$Ganho(S, A) = Entropia(S) - \sum_{j-1}^{m} \frac{|s_j|}{|s|} Entropia(S_j) \tag{3.11}$$

   Onde A é o atributo que será testado, S_j é a soma dos valores do nó, S é o número total de amostras, e $Entropia(S_j)$ é a entropia do nó atual [1].

   Visando a ilustração das métricas de seleção de atributos em uma árvore de decisão, [1] adaptou um exemplo de 30 alunos de uma escola onde o objetivo do modelo é prever quem jogará tênis no intervalo adotando três atributos: Sexo (masculino ou feminino),

Classe (IX ou X) e Altura (160cm a 180cm), onde desses 30 alunos, 15 jogam tênis no intervalo (Figura 3.7).

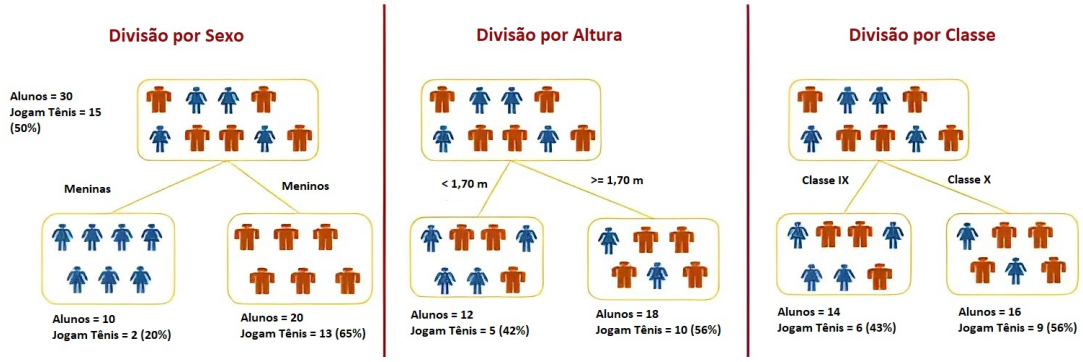


Figura 3.7: Imagem com a distribuição de cada variável do exemplo

Fonte: [1], adaptado pelo Autor, 2023.

Como forma de exemplo, [1] utilizou as variáveis sexo e classe para apresentar os cálculos

1 - Calcular a entropia do nó pai:

$$Entropia(S) = -\left(\frac{15}{30}\right) log_2\left(\frac{15}{30}\right) - \left(\frac{15}{30}\right) log_2\left(\frac{15}{30}\right) = 1$$

2 - Calcular a entropia de cada nó individual e calcular a média ponderada de todos os subnós disponíveis na divisão:

2.1 - Entropia para Sexo:

- Entropia para o subnó feminino:

$$Entropia(SexoF) = -\left(\frac{2}{10}\right) log_2\left(\frac{2}{10}\right) - \left(\frac{8}{10}\right) log_2\left(\frac{8}{10}\right) = 0,72$$

- Entropia para o subnó masculino:

$$Entropia(SexoM) = -\left(\frac{13}{20}\right) log_2\left(\frac{13}{20}\right) - \left(\frac{7}{20}\right) log_2\left(\frac{7}{20}\right) = 0,93$$

- Entropia da divisão por sexo:

$$Entropia(Sexo) = \left(\frac{10}{30}\right) \times 0,72 + \left(\frac{20}{30}\right) \times 0,93 = 0,86$$

2.2 - Entropia para Classe:

- Entropia para o subnó Classe IX:

$$Entropia(ClasseIX) = -\left(\frac{6}{14}\right) log_2\left(\frac{6}{14}\right) - \left(\frac{8}{14}\right) log_2\left(\frac{8}{14}\right) = 0,99$$

- Entropia para o subnó Classe X:

$$Entropia(ClasseX) = -\left(\frac{9}{16}\right) log_2\left(\frac{9}{16}\right) - \left(\frac{7}{16}\right) log_2\left(\frac{7}{16}\right) = 0,99$$

- Entropia da divisão por Classe:

$$Entropia(Classe) = \left(\frac{14}{30}\right) \times 0,99 + \left(\frac{16}{30}\right) \times 0,99 = 0,99$$

Com o cálculo da entropia já é possível identificar que a divisão pela variável Sexo seria o primeiro nó da árvore, uma vez que ele é o que apresentou menor valor. De toda forma, para saber quanta informação se obtém ao fazer essas divisões, é necessário calcular o ganho de informação [1]:

$$Ganho(Sexo) = Entropia(S) - Entropia(Sexo) = 1 - 0,86 = 0,14$$

$$Ganho(Classe) = Entropia(S) - Entropia(Classe) = 1 - 0,99 = 0,01$$

Dessa maneira, como o ganho de informação é maior fazendo a divisão por Sexo, seleciona-se esse atributo. Logo, é possível interpretar que é reduzida a incerteza no resultado da previsão em 0,14 (bits) [1].

3. **Índice de Gini**

O Índice de Gini é uma outra medida de impureza que calcula a heterogeneidade dos dados, sendo utilizado para decidir qual o ponto de corte ideal nas divisões dos nós [1]. Essa medida é calculada por:

$$Gini(S) = 1 - \sum_{i=1}^{m} p_i^2 \tag{3.12}$$

onde p_i é a probabilidade de uma tupla em S pertencer a classe C_i.

O índice de Gini adota divisões binárias para cada variável, em que se pode calcular a soma ponderada da impureza de cada divisão. Se a divisão binária de um atributo particionar os dados em S_1 e S_2, o índice de Gini de S será [1]:

$$Gini(S, A) = \frac{S_1}{S} Gini(S_1) + \frac{S_2}{S} Gini(S_2) \tag{3.13}$$

A homogeneidade do nó será tão maior quanto menor for o valor do Índice de Gini. Se o nó for considerado puro, a impureza de Gini será igual zero. O atributo de divisão selecionado será aquele que possuir menor Índice Gini [1].

Para exemplificar o cálculo do Índice de Gini, adotou-se o mesmo exemplo apresentado anteriormente [1]:

1 - Calcular o Índice Gini para os subnós e calcular sua média ponderada:

1.1 - Índice de Gini para Sexo:

- Gini para subnó feminino:

$$Gini(SexoF) = 1 - \left(\frac{2}{10}\right)^2 - \left(\frac{8}{10}\right)^2 = 0,32$$

- Gini para o subnó masculino:

$$Gini(SexoM) = 1 - \left(\frac{13}{20}\right)^2 - \left(\frac{7}{20}\right)^2 = 0,46$$

- Gini ponderado da Divisão por Sexo:

$$Gini(Sexo) = \left(\frac{10}{30}\right) \times 0,32 + \left(\frac{20}{30}\right) \times 0,46 = 0,41$$

1.2 - Índice de Gini para Classe:

- Gini para subnó Classe IX:

$$Gini(ClasseIX) = 1 - \left(\frac{6}{14}\right)^2 - \left(\frac{8}{14}\right)^2 = 0,49$$

- Gini para subnó Classe X:

$$Gini(ClasseX) = 1 - \left(\frac{9}{16}\right)^2 - \left(\frac{7}{16}\right)^2 = 0,49$$

- Gini ponderado da Divisão por Classe:

$$Gini(Classe) = \left(\frac{14}{30}\right) \times 0,49 + \left(\frac{16}{30}\right) \times 0,49 = 0,49$$

Logo, ao realizar-se os cálculos é possível se observar que o menor índice de Gini é o encontrado na divisão por Sexo. Com isso, da mesma maneira que ocorreu no cálculo da entropia, a variável de Sexo será escolhida para o nó principal da árvore [1].

4. **Representação dos nós para atributos contínuos**

As variáveis quantitativas permitem uma maior variedade de testes e, consequentemente, implicam em cálculos mais complexos. Os teste mais utilizados para partição de atributos contínuos são: testes simples ou pesquisa exaustiva, testes múltiplos (segmentação global e segmentação ao nível do nó) e combinação linear de características [**?**].

O teste simples, também chamado de pesquisa exaustiva, é o normalmente mais utilizado. Nesse caso a divisão do atributo será sempre binária. Considerando um atributo quantitativo X a ser utilizado como nó teste, mesmo que ele possua um domínio infinito, o

número de exemplos em um conjunto de treinamento T será finito e, com isso, o número de diferentes valores para esse atributo também será finito [**?**].

Logo, os exemplos do conjunto T deverão ser ordenados de acordo com seus valores para a variável X. Considerando que os diferentes valores de X sejam, em ordem crescente, $\{a_1, a_2, ..., a_m\}$, o conjunto T deverá ser dividido em duas partes: T_1, cujos exemplos possuem valores $\{a_1, a_2, ..., a_i\}$ e T_2, com valores $\{a_{i+1}, a_2, ..., a_m\}$ para o atributo X [**?**].

Para cada possível valor a_i, $1 = 1, ..., m-1$, deverá ser calculado o ganho (independente do critério utilizado) para a respectiva divisão. Após considerar todas as possíveis divisões, é adotada aquela que fornecer o maior ganho [**?**].

Por fim, é necessário a definição do valor que será usado como limite (valor usado para dividir os exemplos no nó). Tendo em mãos o valor de a_i que definiu o melhor ganho, o valor normalmente utilizado como limiar é $\frac{a_i+a_{i+1}}{2}$, pois com isso é esperado que a árvore resultante apresente melhores resultados para exemplos que não participaram do conjunto de treinamento [**?**].

3.6.2 Regressão Logística

Os modelos de regressão são umas das ferramentas estatísticas mais importantes para a análise de dados quando o principal inutuito é modelar a relação entre as variáveis/atributos. Esses modelos possuem como principal objetivo explorar a relação entre as variáveis independentes, ou explicativas e a variável dependente, ou variável resposta. Um caso particular desses modelos são os casos onde a variável resposta apresenta apenas duas categorias. O modelo de regressão logística é o mais popular entre esses modelos [4].

O principal objetivo da regressão logística é modelar em um conjunto de dados a relação logística entre a variável resposta (rótulo) dicotômica e as demais variáveis explicativas, podendo ser elas numéricas e/ou categóricas [4].

Em diversas situações o principal interesse é analisar o comportamento dessa variável resposta binária, nomeada como variável y, em relação a um conjunto de covariáveis nomeadas $\mathbf{X}$. O modelo de regressão logística é o responsável por estimar as probabilidades de ocorrência de um determinado evento utilizando como base as covariáveis [29].

Seja y_i uma variável resposta binária e $\mathbf{x}_i = (1, x_{i1}, \ldots, x_{ip})^T$ o vetor de p covariáveis do i-ésimo indivíduo $(i = 1, \ldots, n)$ avaliado e, seja $\mathbf{X}$, uma matriz de tamanho $n \times (p + 1)$ da forma $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)^T$. Admitindo que y_i segue uma distribuição Bernoulli(π_i), em que π_i é a probabilidade de ocorrência de um determinado evento, o interesse principal é calcular essa probabilidade através das covariáveis preditoras presentes em $\mathbf{x}_i$. Para que os valores de probabilidade encontrados permaneçam no intervalo $[0, 1]$, deve ser utilizada a função de ligação logito [29]. Logo, a esperança $E[y_i|x_i]$ pode ser escrita da seguinte maneira:

$$E[y_i|\mathbf{x}_i] = \pi_i(\mathbf{x}_i) = \frac{e^{g(\mathbf{x}_i)}}{1 + e^{g(\mathbf{x}_i)}}, i = 1, ..., n \tag{3.14}$$

onde $g(\mathbf{x}_i)$ é o preditor linear da forma

$$g(\mathbf{x}_i) = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + ... + \beta_p x_{ip} \tag{3.15}$$

e $\beta_0, \beta_1, ..., \beta_p$ representam os parâmetros do modelo [29].

A Figura 3.8 representa graficamente o formato da curva logística.

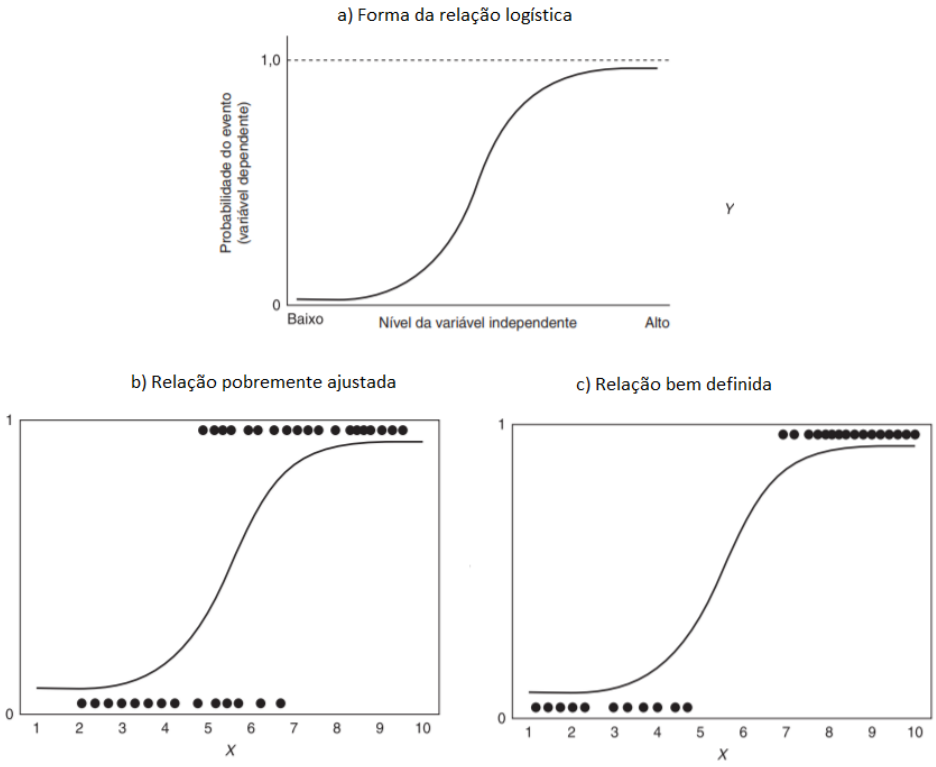


Figura 3.8: Curva logística

Fonte: [28].

1. **Estimação dos parâmetros**

   Segundo [29], para a estimação do vetor $\beta = (\beta_0, \beta_1, \ldots, \beta_p)^T$, é necessário a utilização do método da máxima verossimilhança. Seja $\mathbf{x}_i$ as observações referentes ao i-ésimo indivíduo e $\mathbf{y} = (y_1, \ldots, y_n)^T$. Então a função de log-verossimilhança é dada por

$$l(\beta; X, y) = \sum_{i=1}^{n} \{y_i g(x_i) - log[1 + e^{g(x_i)}]\} \tag{3.16}$$

   Ao se derivar a expressão acima em relação aos parâmetros $\beta_j, j = 1, ..., p$, as estimativas dos parâmetros serão calculadas pela solução simultâneas das equações abaixo, onde x_{ij} representa a j-ésima variável do i-ésimo indivíduo [29].

$$\sum_{i=1}^{n} [y_i - \pi_i(x_i)] = 0 \text{ e } \sum_{i=1}^{n} x_{ij}[y_i - \pi_i(x_i)] = 0, j = 1, ..., p \tag{3.17}$$

   A partir de métodos numéricos é possível estimar-se os parâmetros $\hat{\beta}$. Essas estimatias

serão denotadas por $\hat{\beta} = \hat{\beta}_0, \hat{\beta}_1, ..., \hat{\beta}_p$ [29].

Seja $I(\beta)$ a matriz de Informação de Fisher de ordem $(p+1) \times (p+1)$ dada por

$$I_{jj} = \frac{\partial^2 l(\beta)}{\beta_j^2} = -\sum_{i=1}^{n} x_{ij}^2 \pi_i (1 - \pi_i), \tag{3.18}$$

$$I_{jl} = \frac{\partial^2 l(\beta)}{\beta_j \beta_l} = -\sum_{i=1}^{n} x_{ij} x_{il} \pi_i (1 - \pi_i) \tag{3.19}$$

para $j, l = 0, 1, \ldots, p$ e $\pi_i = \pi_i(x_i)$. É possível obter-se as variâncias e covariâncias do vetor $\hat{\beta}$ a partir da matriz inversa da Informação de Fisher, ou seja, $Var(\hat{\beta}) = I^{-1}(\beta)$. Porém, essa quantidade dependerá de parâmetros que ainda são desconhecidos. Para solucionar esse problema, deverá ser utilizado a estimativa da variância a partir da estimativa da matriz de Informação de Fisher, logo, $\hat{Var}(\hat{\beta}) = \hat{I}^{-1}(\hat{\beta})$ [29]. A matriz de Fisher estimada pode ser escrita da forma $\hat{I}^{-1}(\hat{\beta}) = X^T V X$, em que

$$X = \begin{bmatrix} 1 & x_{11} & \ldots & x_{1p} \\ 1 & x_{21} & \ldots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & \ldots & x_{np} \end{bmatrix}_{n \times (p+1)} \quad \text{e} \quad V = \begin{bmatrix} \hat{\pi}_1(1-\hat{\pi}_1) & 0 & \ldots & 0 \\ 0 & \hat{\pi}_2(1-\hat{\pi}_2) & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & \hat{\pi}_n(1-\hat{\pi}_n) \end{bmatrix}_{n \times n}$$

e $\hat{\pi}_i$ é a estimativa da probabilidade de ocorrência do evento em estudo para i-ésimo indivíduo. A variância estimada de um determinado parâmetro $\hat{\beta}_j$ é encontrada no termo jj da matriz de variâncias estimadas e a covariância entre $\hat{\beta}_j$ e $\hat{\beta}_l$ é encontrada no termo jl, ou no termo lj, já que é uma matriz simétrica [29]. Com isso, o desvio padrão estimado de um determinado coeficiente $\hat{\beta}_j$ é dado por

$$\hat{\sigma}(\hat{\beta}_j) = \sqrt{\hat{Var}(\hat{\beta}_j)} = \sqrt{\hat{I}_{jj}^{-1}(\hat{\beta})} \tag{3.20}$$

2. **Predição**

Nos casos em que a variável resposta é dicotômica é necessário a determinação de uma regra a ser utilizada na predição da classificação, visto que a probabilidade estimada $\hat{\pi}$ está compreendida entre 0 e 1. Normalmente é intuitivo supor que se $\hat{\pi}_i$ apresentar um valor próximo de 1, deve-se adotar que $\hat{Y}_i = 1$ e, se $\hat{\pi}_i$ corresponder um valor pequeno ou próximo de zero, deve-se adotar que $\hat{Y}_i = 0$. Porém, como escolher um ponto "limite" onde os valores acima dele serão classificados como "evento" ($\hat{Y}_i = 1$) e valores abaixo dele classificados como "não evento" ($\hat{Y}_i = 0$)? Esse ponto é chamado como ponto de corte [26].

Uma maneira muito utilizada para determinação desse ponto é através da chamada Curva ROC (*Receiver Operating Characteristic Curve*). Geometricamente, essa curva é um gráfico de pares $1 - P(\hat{Y}_i = 0|Y = 0)$ e $P(\hat{Y}_i = 1|Y = 1)$, representados em um plano chamado de plano ROC unitário. A definição de plano ROC unitário, é pelo fato das coordenadas deste gráfico representarem medidas de probabilidade, e consequentemente variarem entre zero e um [26].

Para determinar o ponto de corte é necessário basear-se numa combinação ótima de sensibilidade e de especificidade (essas medidas estão sendo apresentadas no tópico Medidas de Avaliação), pois iniciam com o pressuposto que classificar o indivíduo como "evento" dado que ele é "não evento" (falso positivo) e classificar o indivíduo como "não evento" dado que ele é "evento" (falso negativo) pode acarretar prejuízos para um investigador. Pela análise da curva ROC, o ponto de corte é escolhido através da combinação da sensibilidade e 1-especificidade que mais se aproxima do canto superior esquerdo do gráfico [26].

A Figura 3.9 apresenta uma representação da curva ROC:

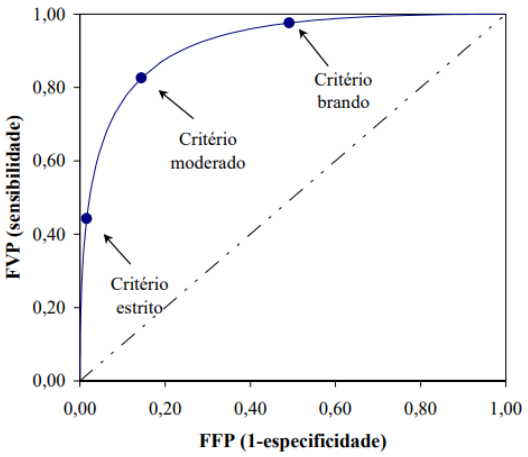


Figura 3.9: Curva ROC, para uma dada capacidade de discriminação, com a variação do critério de decisão

Fonte: [3].

3.7 MODELO DE PROCESSO CRISP-DM

O modelo de processo CRISP-DM significa Cross Industry Standard Process for Data Mining (processo padrão da indústria cruzada para mineração de dados), foi desenvolvido como um projeto neutro de indústria e ferramentas para a aplicação de técnicas de mineração de dados [20].

Nesse modelo, o projeto de mineração de dados está dividido em seis fases. A sequência dessas fases não é obrigatório, pois está atrelada ao resultado de cada fase ou qual tarefa particular de uma fase precisa ser executada na próxima fase [20].

A Figura 3.10 ilustra a sequência da metodologia. As setas indicam as dependências mais importantes e frequentes entre as fases. O círculo externo na Figura simboliza o ciclo da mineração de dados. Pela Figura pode-se perceber que processo de mineração tem uma continuidade mesmo após a definição de uma solução. Os próximos projetos de mineração poderão se beneficiar com as experiências anteriores.

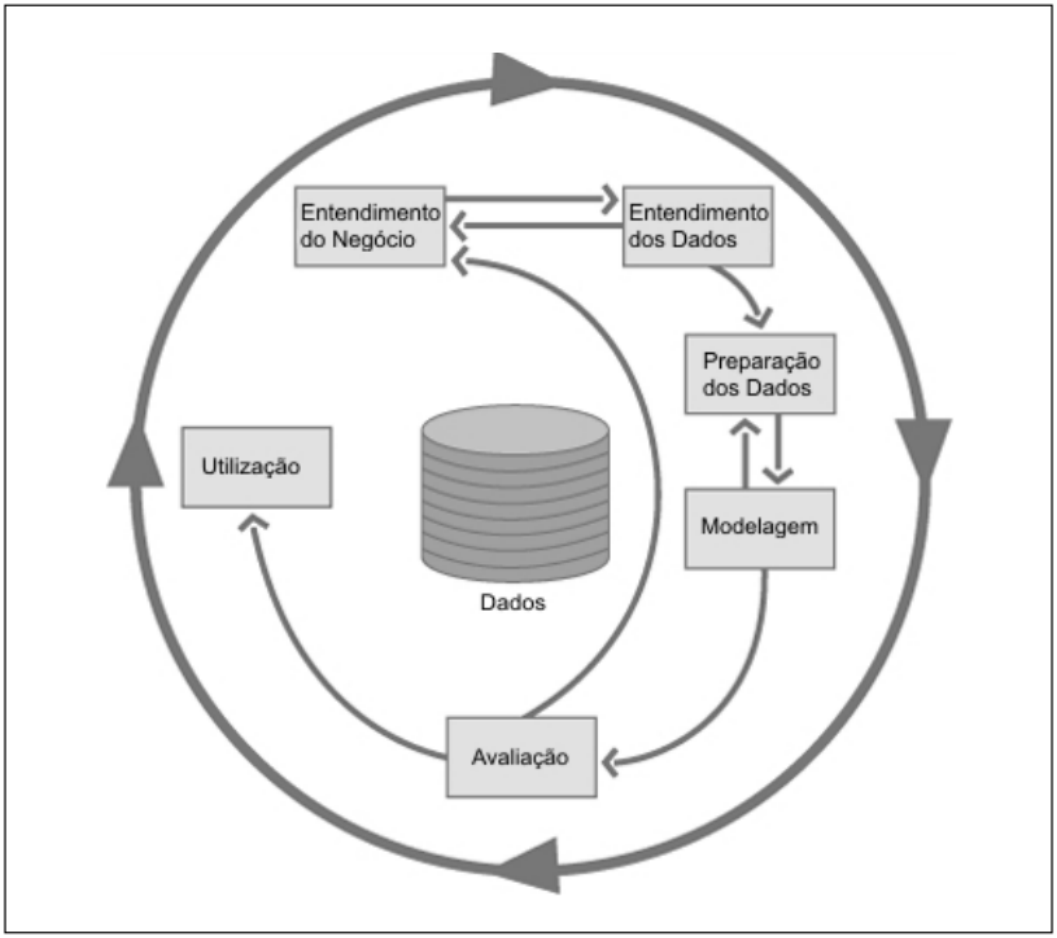


Figura 3.10: Metodologia CRISP-DM para mineração de dados.
Fonte: [20]

Descreve-se a seguir, de forma resumida, cada uma das etapas da metodologia CRISP-DM. Os conceitos foram apresentados segundo o trabalho de Petermann [20].

Entendimento do Negócio *(Business Understanding)*

Etapa de entendimento das regras de negócio da empresa e definição dos objetivos do projeto. Após entendimento prévio é possível determinar o problema de mineração de dados e traçar um plano para atingimento dos objetivos.

Entendimento dos dados *(Data Understanding)*

Etapa onde ocorre a coleta dos dados e se iniciam as atividades para entendimento dos mesmos, buscando identificar possíveis problemas na sua qualidade, discernimentos possíveis e detecção de sub-conjuntos interessantes para criação e hipóteses.

Preparação dos dados *(Data Preparation)*

Etapa onde é realizada a construção do conjunto de dados (data set) final. Essas tarefas de preparação podem ser realizadas várias vezes, e não seguem uma ordem específica. Nesse momento que são realizadas a seleção de tabelas, registros e atributos, além de realizar transformações e limpezas dos dados.

Modelagem *(Modelling)*

Etapa onde são definidos os modelos a serem utilizados, além de possíveis ajustes de seus parâmetros, uma vez que podem existir várias técnicas para o mesmo problema. Pode-se ter requisitos específicos de formação de dados para algumas técnicas, logo, pode ser necessário o retorno à fase de preparação de dados.

Avaliação *(Evaluation)*

Nesse momento o modelo construído é avaliado para validação de que os objetivos levantados inicialmente foram atendidos. É de extrema importância para determinar se alguma regra do negócio foi deixada de lado ou não foi suficientemente considerada.

Utilização ou aplicação *(Deployment)*

É o momento de utilização/aplicação do modelo criado anteriormente. Nesse momento que os dados finais são utilizados pelo solicitante através de relatórios ou então um modelo que gera um processo repetitivo de ajustes e evolução.

4. METODOLOGIA

Para o desenvolvimento desse trabalho, onde o principal objetivo foi desenvolver um modelo de classificação, através de técnicas de mineração de dados, para a classificação do perfil dos clientes (Cliente Venda ou Cliente Não Venda), foram utilizadas bases de dados reais de uma empresa de *Outsourcing*. A operação utilizada como estudo de caso trabalha com a venda de produtos financeiros. As tratativas das bases de dados e aplicação dos algoritmos foram através da Linguagem Python [31], utilizando-se o Jupyter Notebook e também pelo Software Excel.

A estrutura deste Estudo de Caso utilizou a metodologia CRISP-DM, que é a mais usual em problemas que envolvam Mineração de Dados [12].

A proposta para utilização deste modelo é pelo fato de ser um processo aplicável aos diferentes segmentos da indústria, acarretando em uma maior agilidade, redução de custos e melhor gerenciamento no processo de mineração de dados.

Descreve-se a seguir, as etapas seguidas para desenvolvimento desse trabalho, seguindo a metodologia CRISP-DM:.

4.1 ENTENDIMENTO DO NEGÓCIO

A vivência dentro da empresa citada anteriormente permitiu o entendimento do negócio da operação. Entendendo seus processos e o seu funcionamento foi identificado o problema que se tornou objetivo desse trabalho: Existia uma base de clientes relativamente extensa, e o contato com os mesmos era feito de forma aleatória, com isso identificou-se a oportunidade de aplicação de um modelo de classificação com a base de clientes já contactados e assim criar um modelo mais assertivo.

4.2 ENTENDIMENTO DOS DADOS

Para o desenvolvimento do trabalho foram utilizadas três bases de dados reais fornecidas pela empresa:

1. A base de dados de cadastro dos clientes, onde trazem todos os dados pessoais dos mesmos. Essa base que foi o ponto de partida e traz todas as variáveis utilizadas no modelo;

2. A base de dados com informação de contato. Através dessa base foi possível determinar quais clientes de fato foram contactados;

3. A base de vendas. Essa base traz como informação quais clientes efetivaram de fato a venda. Foram realizadas análises sobre as bases listadas anteriormente, buscando um maior entendimento e identificação de problemas sobre os dados contidos nelas.

4.3 PREPARAÇÃO DOS DADOS

Uma das partes mais importantes do processo, essa etapa é onde foram realizados os pré-processamento dos dados.

As três bases foram consolidadas. A partir da base de dados cadastrais foi realizado o filtro apenas dos clientes que realmente receberam o contato segundo a base de SMS (*Short Message Service* - Serviço de Mensagens Curtas). Após esse filtro, foi criada a coluna de Perfil do Cliente (Rótulo) atráves da base de vendas. O cliente que efetivou a venda em algum momento, posterior ao início de envio dos SMS's, foi chamado de "Cliente Venda", e os demais de "Cliente Não Venda".

Dos atributos contidos na base final, optou-se por manter para utilização do modelo apenas os listados abaixo:

- **ORGÃO:** Convênio ao qual o cliente está relacionado. Este Convênio está diretamente ligado ao Órgão Público que o cliente está associado, como por exemplo, o nome do Governo ou Prefeitura que o mesmo faz parte;
- **ESTADO:** Estado de residência do Cliente;
- **SEXO:** Sexo do Cliente;
- **IDADE:** Idade do Cliente;
- **GRUPO FUNÇÃO:** Profissão do Cliente;
- **SALDO DISPONÍVEL:** Valor, em reais, que o cliente possuí disponível para saque junto à instituição. Esse é o valor que é "vendido" ao cliente.
- **REDA:** Define se é um funcionário contrato (Reda = SIM) ou um funcionário público do órgão (Reda = NÃO).
- **PERFIL:** Define se é um cliente NOVO (nunca realizou compra), se é um cliente TOMADOR SEM SAQUE (já teve compra e não tem nenhum saldo utilizado) ou se é um TOMADOR (cliente que já teve compra e tem saldo utilizado).
- **REFIN:** Define se o cliente é passível de realizar refinanciamento (REFIN).
- **RÓTULO:** Classificação do cliente segundo o fato de ter ou não realizado uma compra. As possíveis classificações são Venda ou Não Venda.

Optou-se por manter apenas os atributos mencionados anteriormente pelo fato dos demais dados serem confidenciais e/ou redundantes na base de dados. Um exemplo dessa situação é o uso da data de nascimento, que se torna um atributo redundante uma vez que já está sendo utilizado a idade do cliente nos modelos.

A partir de então foram iniciadas as análises gerais sobre a base final e análises descritivas realizadas sobre as variáveis existentes.

Foram levantadas a quantidade de dados ausentes em cada um dos atributos da base e apenas a variável "GRUPO FUNÇÃO" possuia dados nessa condição. Pela proporção de dados ausentes nessa coluna (29,23%), optou-se por removê-los da base [25].

Além disso, identificou-se que o range de idades elegíveis para realização da venda são a de Clientes com idade superior ou igual a 25 anos e inferior a 80 anos. Por esse motivo, os clientes que se encontraram fora desse range (1.914) foram removidos da base.

Dentro dos atributos ORGÃO E GRUPO FUNÇÃO haviam um volume de categorias muito alto, 62 classes diferentes para o ORGÃO e 175 classes diferentes para o GRUPO FUNÇÃO. Visando um melhor processamento dos algoritmos buscou-se uma forma de sumarizar as categorias desses atributos.

No atributo GRUPO FUNÇÃO notou-se que das 175 diferentes profissões apenas 19 delas representavam praticamente 90% dos dados. Logo, para essas profissões manteve-se o nome utilizado, e as outras 156 foram renomeadas como OUTRO. Com isso, na base final esse atributo possuía apenas 20 diferentes categorias.

No atributo ORGÃO, as categorias que possuíam a nomenclatura GOV foram todas renomeadas para GOV, as que se referiam a prefeituras de cidades foram renomeadas por PREFEITURA, e os demais receberam o nome OUTRO. Com isso, esse atributo que anteriormente possuia 62 classes diferentes ficou com apenas 3 categorias na base final.

Ainda na etapa de pré processamento dos dados, foi necessária a realização da transformação de algumas das variáveis utilizadas no modelo. Das 10 variáveis escolhidas, apenas duas são variáveis numéricas. Para as demais, foram necessários processos que as convertam de categóricas para númericas, uma vez que a maioria dos algoritmos de Machine Learning exigem que os atributos sejam numéricos [9].

Dessa forma, para aplicação nos algoritmos dentro da Linguagem Python, as variáveis categóricas (após o processo de sumarização) foram transformadas em variáveis Dummies, ou seja, cada variável foi transformada em novas colunas contendo a informação binária relativa ao pertencimento daquele dado àquela classe. Por exemplo, a variável ÓRGÃO foi transformada em duas novas colunas chamadas: ÓRGÃO_OUTRO e ÓRGÃO_PREFEITURA. Caso aquele elemento da base pertenceça ao órgão Outro, o valor dessa coluna será 1 e da outra será 0. Caso ele pertença ao órgão Prefeitura, o valor dessa coluna será 1 e da outra será 0. Caso não pertença ao órgão GOV, ou seja, não pertença nem ao órgão Outro e nem ao Prefeitura, ambas colunas terão o valor 0. Esse mesmo raciocínio foi aplicado às demais colunas categóricas.

Como as variáveis numéricas IDADE e SALDO DISPONÍVEL possuem escalas consideravelmente desproporcionais, foi aplicado nessas duas variáveis o processo de normalização dessas

variáveis. Dessa maneira, ambas as variáveis passaram a integrar o intervalo de 0 a 1.

Por fim, a base de dados utilizada na construção do modelo é uma base desbalanceada, ou seja, uma das classes tem uma frequência muito menor que a outra. Na construção do modelo isso é um grande problema, pois o algoritmo aprende muito sobre a classe predominante e pouco sobre a classe com menor ocorrência. Para evitar esse problema foi utilizado o algoritmo SMOTE, apresentado na seção 3.5, para que a base dados fosse balanceada. O algoritmo foi aplicado na base de treino, e com isso, novos dados da classe de menor ocorrência foram criados de forma sintética.

Dentro dos algoritmos de classificaão da biblioteca scikit-learn do Python existe um parâmetro chamado *class_weight*. Quando esse parâmetro assume o valor "balanced", o modelo entende que a base de dados é desbalanceada, e internamente já aplica métodos de balanceamento no momento do treino. Neste trabalho adotou-se também essa metodologia para fim de comparação entre as duas práticas.

Após essas etapas, a base final estava pronta para ser aplicada nos modelos propostos.

4.4 MODELAGEM

Etapa onde são definidos os modelos a serem utilizados, além de possíveis ajustes de seus parâmetros, uma vez que podem existir várias técnicas para o mesmo problema. Pode-se ter requisitos específicos de formação de dados para algumas técnicas, logo, pode ser necessário o retorno à fase de preparação de dados.

Nessa etapa foram aplicados os algoritmos de Machine Learning para desenvolvimento dos modelos de classificação. Para esse trabalho foram adotados 02 (dois) algoritmos diferentes, que são: Árvore de Decisão e Regressão Logística.

Tanto a regressão logística quanto as árvores de decisão são flexíveis em termos de manipulação de variáveis categóricas e numéricas. A regressão logística pode lidar facilmente com variáveis categóricas através de codificação adequada, enquanto as árvores de decisão podem lidar com ambos os tipos de variáveis.

Conforme prática apresentada na fundamentação teórica 3.3, a base final foi dividida em duas bases: uma base de treino e uma base de teste. Foi adotado uma porcentagem de 30% dos dados para a base de teste, buscando-se manter a mesma proporção de classe VENDA e NÃO VENDA no treino e no teste.

Após as divisões foi utilizada a biblioteca scikit-learn, do Python [31], para aplicação dos algoritmos propostos.

4.5 AVALIAÇÃO

Nesse momento o modelo construído é avaliado para validação de que os objetivos levantados inicialmente foram atendidos. É de extrema importância para determinar se alguma regra do negócio foi deixada de lado ou não foi suficientemente considerada.

Nesse momento foram utilizadas medidas de avaliação dos modelos de classificação desenvolvidos, buscando-se determinar qual dos dois modelos foi mais satisfatório (melhores medidas) e se as medidas são satisfatórias para o caso real estudado.

Dentre as medidas de avaliação existentes, foram utilizadas para esse trabalho a precisão, recall e F1-score. A escolha dessas métricas permite uma avaliação abrangente do desempenho dos modelos de classificação. Enquanto a precisão fornece uma medida da proporção de predições corretas, o recall mede a capacidade do modelo de encontrar todas as instâncias positivas/negativos corretamente. O F1-score, por sua vez, combina a precisão e o recall em uma única métrica que considera tanto os verdadeiros positivos quanto os falsos positivos e falsos negativos.

Além disso, em problemas de classificação com classes desequilibradas, onde uma classe possui muito mais exemplos do que outras, o recall e o F1-score levam em consideração as instâncias positivas corretamente classificadas, o que é particularmente relevante nessas situações, pois o objetivo é garantir que todas as instâncias positivas sejam identificadas corretamente.

4.6 Utilização ou aplicação

Na empresas, essa etapa é onde o modelo criado (e validado) será colocado em prática, adotando-o para classificação dos clientes ainda não contactados. Nesse trabalho ela não será aplicável.

5. RESULTADOS

5.1 PRÉ-PROCESSAMENTO DOS DADOS

Após a realização do cruzamento das três bases de dados utilizadas nesse trabalho, resultou-se em uma base de dados final com 367.176 linhas.

Dentro do Python, o primeiro procedimento de pré-processamento realizado foi a tratativa dos dados ausentes. Após análise preliminar foi identificado que apenas a variável GRUPO FUNÇÃO possuía valores ausentes ou então valores classificados como N€O INFORMADO, NA0 INFORMADO, NAO INFORMADO ou NAO LOCALIZADO. Essas quatro últimas classificações, na prática, equivalem também a valores ausentes. As linhas nessas situações totalizaram 107.323, que representa 29,23% da base total. Como essa variável representa a profissão do cliente, optou-se por excluir esses dados da base original, o que resultou em uma base para aplicação do modelo com tamanho consideravelmente bom, 259.853 linhas.

Ainda nesta etapa foram realizadas análises descritivas sobre as variáveis utilizadas no modelo. Identificou-se, pela análise gráfica (Figura 5.1) e pelo conhecimento do negócio, que idades a partir de 25 anos e inferiores a 80 anos estão mais propícias à finalização da venda.

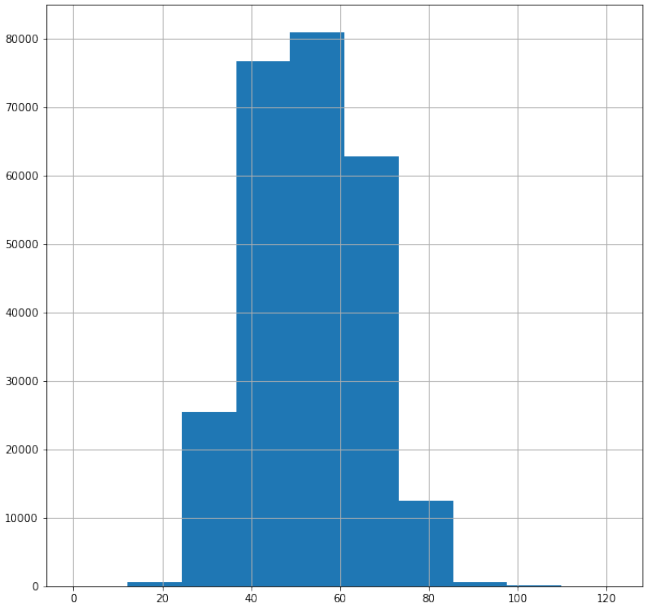


Figura 5.1: Distribuição da Variável Idade dos Clientes.

Fonte: Elaborado pelo Autor, 2023.

Com isso, optou-se por manter na base de dados apenas os clientes com idade maior ou igual a 25 anos e com menos de 80 anos, resultando em um dataframe com 257.939 linhas.

Da base final avaliou-se que apenas 0,72% dos dados são da categoria de venda (Figura 5.2), que é a categoria de interesse dos modelos. Isso representa uma base desbalanceada, o que necessitou de técnicas específicas para o momento de execução do modelo de classificação.

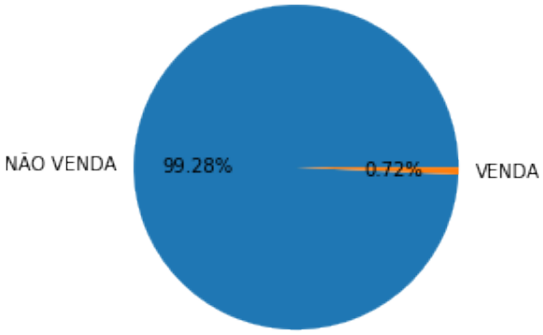


Figura 5.2: Distribuição da Classificação dos Clientes (Rótulo).

Fonte: Elaborado pelo Autor, 2023.

Dentre os clientes analisados, 54,86% são do sexo Feminino e 45,14% do sexo Masculino (Figura 5.3).

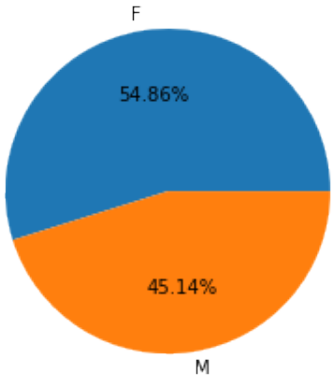


Figura 5.3: Distribuição da Variável Sexo dos Clientes.

Fonte: Elaborado pelo Autor, 2023.

É possível notar também que a grande maioria dos clientes, 81,97%, estão localizados nos estados de Rio de Janeiro e Bahia (Figura 5.4).

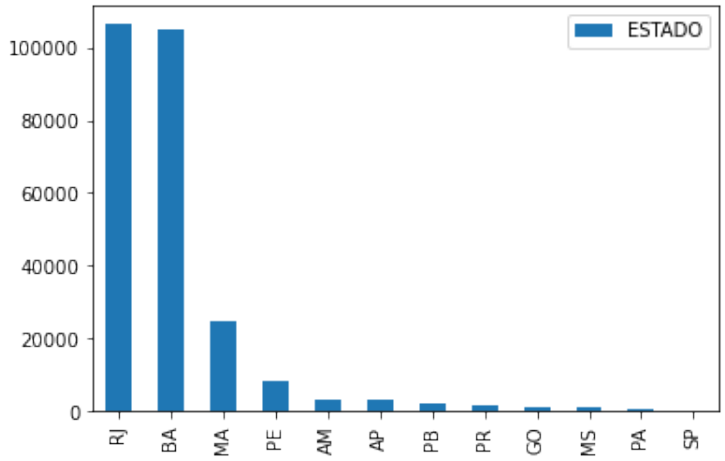


Figura 5.4: Distribuição dos clientes segundo seu Estado.

Fonte: Elaborado pelo Autor, 2023.

Além disso, 28,50% dos clientes possuem a profissão de Professor, seguido pela profissão de Polícia, 17,08%. Na Figura 5.5 é possível avaliar o ranking das dez principais profissões dos clientes na base de dados.

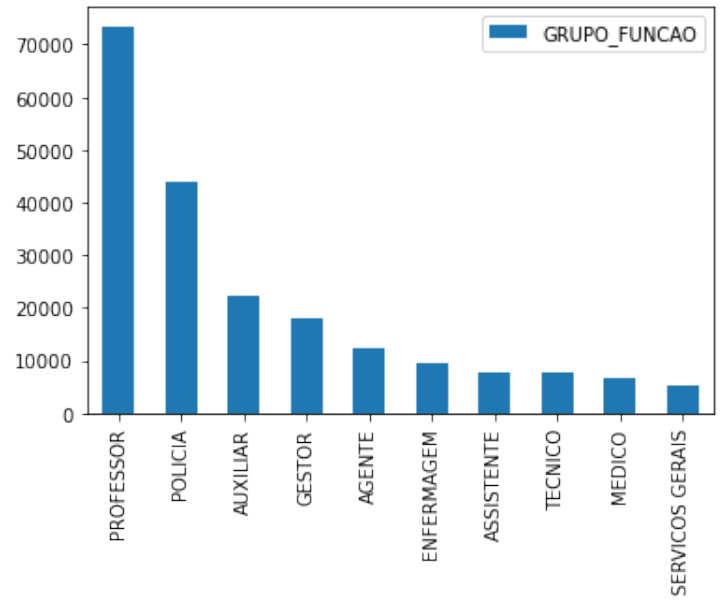


Figura 5.5: Distribuição das 10 principais profissões dos clientes.

Fonte: Elaborado pelo Autor, 2023.

Na Tabela 5.1 é apresentada a distribuição de frequência do atributo GRUPO FUNÇÃO após aplicação da sumarização. A categoria OUTRO foi criada pela sumarização das outras 156 categorias com menor frequência.

Ainda na etapa de pré-processamento, os dados categóricos foram convertidos em variáveis binárias. Dessa maneira, cada possível categoria das variáveis categóricas é transformada em uma nova coluna contendo o valor 0 ou 1. O valor será 1 caso a categoria da coluna "original" for a mesma da nova coluna criada e 0 caso seja diferente.

Tabela 5.1: Distribuição de frequência do GRUPO FUNÇÃO após sumarização das classes.

Grupo Função	Frequência	Freq. Relativa	Freq. Acumulada
MERENDEIRA	2123	0,82%	**0,82%**
ANALISTA	2172	0,84%	**1,67%**
GUARDA	2279	0,88%	**2,55%**
DIRETOR	2372	0,92%	**3,47%**
DIGITADOR/ESCREVENTE	2641	1,02%	**4,49%**
COORDENADOR	3393	1,32%	**5,81%**
PENSIONISTA	3513	1,36%	**7,17%**
SOCIO	4829	1,87%	**9,04%**
BOMBEIRO	4900	1,90%	**10,94%**
SERVICOS GERAIS	5074	1,97%	**12,91%**
MEDICO	6775	2,63%	**15,54%**
TECNICO	7652	2,97%	**18,50%**
ASSISTENTE	7774	3,01%	**21,52%**
ENFERMAGEM	9594	3,72%	**25,24%**
AGENTE	12411	4,81%	**30,05%**
GESTOR	17889	6,94%	**36,98%**
AUXILIAR	22148	8,59%	**45,57%**
POLICIA	44054	17,08%	**62,65%**
PROFESSOR	73500	28,50%	**91,14%**
OUTRO	22846	**8,86%**	100,00%

Fonte: Elaborado pelo Autor, 2023.

Após a aplicação da técnica SMOTE, a base de dados de treino tornou-se uma base balanceada, sendo 50% da classe VENDA e 50% da classe NÃO VENDA, conforme apresentado na Figura 5.6:

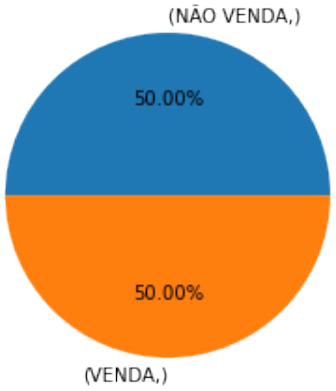


Figura 5.6: Distribuição da Classificação dos Clientes (Rótulo) após aplicação do SMOTE.

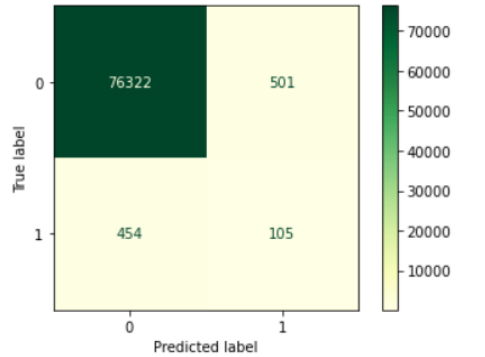
Fonte: Elaborado pelo Autor, 2023.

5.2 Aplicação dos algoritmos

Nas Figuras onde são apresentadas as matrizes de confusão e as medidas de avaliação dos modelos, os rótulos de valor 0 são referentes a classe Não Venda, e os rótulos de valor 1 à classe

Venda.

A primeira metodologia adotada foi a aplicação dos modelos sem nenhuma técnica de balanceamento dos dados. A matriz de confusão e as medidas de avaliação desse modelo estão repesentadas na Figura 5.7:



Relatório de Classificação:

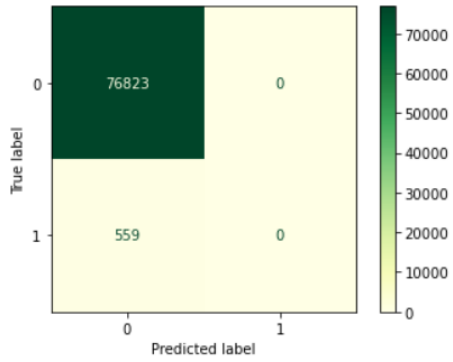
	precision	recall	f1-score
0	0.9941	0.9935	0.9938
1	0.1733	0.1878	0.1803

Figura 5.7: Matriz de confusão e medidas de avaliação resultante do algoritmo Árvore de Decisão aplicado na base original.

Fonte: Elaborado pelo Autor, 2023.

Esse caso resultou em um modelo que acerta muito a classe de Não Venda e pouco a classe de Venda. Essa situação era esperada, pelo de fato de tratar-se de uma base de dados desbalanceada. Com isso, o modelo aprende muito da classe com maior ocorrência, tenho uma maior acertividade nessa situação. Dentre todas as possíveis vendas, apenas 0,1878 (aproximadamente 19%) foi identificado. E dentre todos os casos que foram classificados como Venda, apenas 0,1733 (aproximadamente 18%) estavam corretos.

Ainda analisando a base de dados original, sem nenhum tipo de balanceamento, o próximo algoritmo aplicado foi o de regressão logística. A matriz de confusão e as medidas de avaliação desse modelo estão repesentadas na Figura 5.8:



Relatório de Classificação:

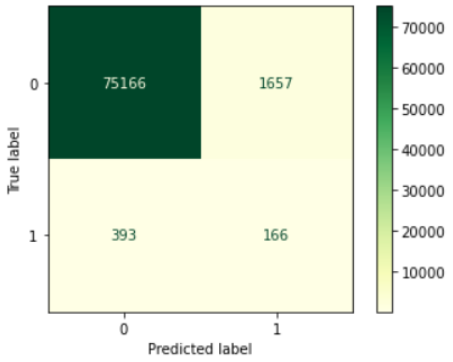
	precision	recall	f1-score
0	0.9928	1.0000	0.9964
1	0.0000	0.0000	0.0000

Figura 5.8: Matriz de confusão e medidas de avaliação resultante do algoritmo Regressão Logística aplicado na base original.

Fonte: Elaborado pelo Autor, 2023.

Assim como na árvore de decisão, esse algoritmo também resultou em um modelo que acerta muito a classe de Não Venda, porém nesse caso ele não acertou nada da classe de Venda. Isso fez com que o modelo obtivesse excelentes medidas em relação a classe majoritária e péssimas medidas em relação à minoritária (todos os valores zerados).

O próximo passo foi a aplicação do algoritmo SMOTE na base de treino. Com isso, foi realizada a separação em treino e teste e após isso a base de treino sofreu o balanceamento. Ao aplicar a árvore de decisão, os resultados obtidos foram (Figura 5.9):



Relatório de Classificação:

	precision	recall	f1-score
0	0.9948	0.9784	0.9865
1	0.0911	0.2970	0.1394

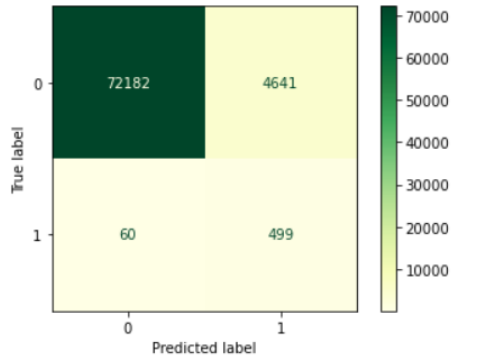
Figura 5.9: Matriz de confusão e medidas de avaliação resultante do algoritmo Árvore de Decisão aplicado na base tratada pelo SMOTE.

Fonte: Elaborado pelo Autor, 2023.

Em comparação com o modelo utilizando a base original, é possível identificar singelas melhorias do modelo. Nessa situação, os casos de Não Venda continuaram com uma boa acertividade, mantendo as medidas de avaliação com valores altos. Ao analisar os casos de venda,

dentre todas as possíveis vendas, houve um aumento da acertividade para 0,2970 (aproximadamente 30%). E avaliando todos os casos que foram classificados como Venda, ocorreu uma redução da acertividade para 0,0911 (aproximadamente 10%), ou seja, dentre tudo que foi classificado como venda, apenas 10% realmente era.

Ao aplicar a Regressão Logística na mesma base tratada pelo SMOTE, os resultos foram (Figura 5.10):



Relatório de Classificação:

	precision	recall	f1-score
0	0.9992	0.9396	0.9685
1	0.0971	0.8927	0.1751

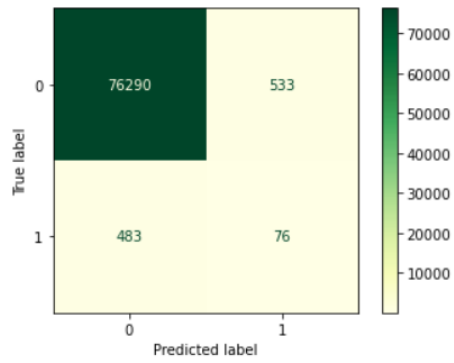
Figura 5.10: Matriz de confusão e medidas de avaliação resultante do algoritmo Regressão Logística aplicado na base tratada pelo SMOTE.

Fonte: Elaborado pelo Autor, 2023.

Ao se comparar esse modelo de Regressão Logística, onde a base de treino foi balanceada com o SMOTE, com o modelo aplicado na base original, é possível identificar um enorme ganho. Após a aplicação do balanceamento através do SMOTE, o modelo manteve boas medidas de avaliação em relação aos casos de Não Venda e aumentou expressivamente os acertos dos casos de venda. Dentre todos os possíveis casos de vendas, esse modelo passou a acertar 0,8927 (aproximadamente 90%) dos casos. E dentre todos os casos que ele avaliou como venda, 0,0971 (aproximadamente 10%) realmente era.

A última análise realizada foi utilizando a base de dados original porém adotando um parâmetro interno dos modelos chamado class_weight. Atribuindo o valor "balanced" a esse parâmetro, o próprio modelo realiza o balanceamento da base de treino. Para o algoritmo de árvore de decisão, o resultado obtido foi o apresentado na Figura 5.11:

Ao comparar esse modelo com o balanceado pelo SMOTE, é possível notar algumas mudanças nas medidas de avaliação. Por mais que os acertos da classe de Não Venda tenha mantido-se alto, mudanças expressivas podem ser notadas em relação a classe minotária (Venda). Dentre todas as vendas reais, houve uma redução do % de acertos. Anteriormente estava em aproximadamente 31%, e o valor abaixou para aproximadamente 13% (0,1360). Logo, em uma situação prática, várias possíveis vendas deixariam de ser abordadas nessa situação. Já para todos os casos classificados como venda, 0,1248 (aproximadamente 13%) eram de fato uma venda. Houve um aumento em relação ao caso com o SMOTE, ou seja, esse modelo classifica erroneamente



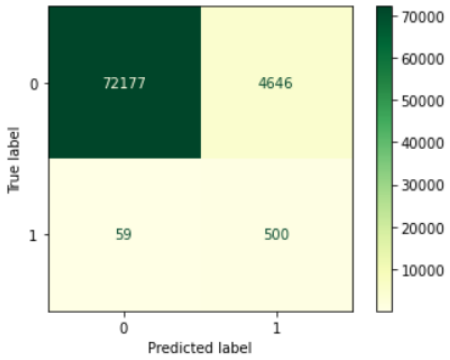
Relatório de Classificação:

	precision	recall	f1-score
0	0.9937	0.9931	0.9934
1	0.1248	0.1360	0.1301

Figura 5.11: Matriz de confusão e medidas de avaliação resultante do algoritmo Árvore de Decisão aplicado na base original com o balanceamento pelo algoritmo.
Fonte: Elaborado pelo Autor, 2023.

menos casos que o anterior, porém, têm-se uma perda ao deixar de classificar muitos casos que realmente eram vendas.

Por fim, aplicou-se a Regressão Logística à essa base original utilizando o parâmetro class_weight. Os resultados são apresentados na Figura 5.12:



Relatório de Classificação:

	precision	recall	f1-score
0	0.9992	0.9395	0.9684
1	0.0972	0.8945	0.1753

Figura 5.12: Matriz de confusão e medidas de avaliação resultante do algoritmo Regressão Logística aplicado na base original com o balanceamento pelo algoritmo.
Fonte: Elaborado pelo Autor, 2023.

Nesse último modelo, os ganhos em relação ao caso onde a base foi balanceada pelo SMOTE é praticamente zero. Ou seja, tanto com o balanceamento pelo SMOTE, quanto pelo balanceamento dentro do próprio modelo, os valores das medidas de avaliação mantiveram-se praticamente iguais.

Nas Tabelas 5.2, 5.3 e 5.4 é apresentado, de forma resumida, as medidas de avaliação resultantes de cada um dos modelos listados anteriormente:

Tabela 5.2: Comparação da Precisão entre os algorítmos adotados.

Base	CLASSE VENDA		CLASSE NÃO VENDA	
	Árvore de Decisão	Regressão Logística	Árvore de Decisão	Regressão Logística
Base Original	0,1733	0,0000	0,9941	0,9928
Balanceamento com SMOTE	0,0911	0,0971	0,9948	0,9992
Balanceamento pelo modelo	0,1248	0,0972	0,9937	0,9992

Fonte: Elaborado pelo Autor, 2023.

Tabela 5.3: Comparação do Recall entre os algorítmos adotados.

Base	CLASSE VENDA		CLASSE NÃO VENDA	
	Árvore de Decisão	Regressão Logística	Árvore de Decisão	Regressão Logística
Base Original	0,1878	0,0000	0,9935	1,0000
Balanceamento com SMOTE	0,2970	0,8927	0,9784	0,9396
Balanceamento pelo modelo	0,1360	0,8945	0,9931	0,9395

Fonte: Elaborado pelo Autor, 2023.

Tabela 5.4: Comparação do F1-Score entre os algorítmos adotados.

Base	CLASSE VENDA		CLASSE NÃO VENDA	
	Árvore de Decisão	Regressão Logística	Árvore de Decisão	Regressão Logística
Base Original	0,1803	0,0000	0,9938	0,9964
Balanceamento com SMOTE	0,1394	0,1751	0,9865	0,9685
Balanceamento pelo modelo	0,1301	0,1753	0,9934	0,9684

Fonte: Elaborado pelo Autor, 2023.

Na forma de trabalho atual, os contatos são realizados de forma aleatória através de envio de SMS, o que gera um custo por contato. Nesse modelo, os SMS são enviados de forma aleatória e para um volume muito grande de clientes, resultando em um custo alto. Ao adotar-se o modelo de classificação proposto atráves do algoritmo de regressão logística realizando o tratamento dos dados desbalanceados, poderia reduzir-se o volume de mensagens enviadas, resultando também em uma redução do custo. Por exemplo, se cada SMS enviado tem um custo de R$0,10, ao enviar-se de forma aleatória para uma base de 77.382 clientes teria-se um custo aproximado de R$7.738,00, lembrando que o retorno seria de 0,72% de venda. Com o modelo, nessa base de 77.382 poderia-se enviar as mensagens para apenas 5.140 clientes, o que resultaria em um custo de R$514,00, ou seja, uma economia de R$7.224,00, além de ter um retorno esperado de 10% de vendas.

6. CONCLUSÃO

Dentre todos os casos avaliados, para a classificação do perfil dos Clientes em Cliente Venda ou Não Venda, objetivo principal deste trabalho, o modelo que resultou em medidas de avaliação mais satisfatórias foi o modelo utilizando-se Regressão Logística com balanceamento dos dados de treino. Tanto o caso de balanceamento utilizando a técnica SMOTE quanto o balanceamento através do próprio algoritmo do modelo resultaram em medidas de avaliação semelhantes.

Em modelos de machine learning, a análise da satisfação das medidas de avaliação utilizadas vai de encontro com o objetivo do modelo dentro da empresa. A partir da base de dados utilizada para a construção dos modelos e pela vivência prática dentro da organização, anteriormente a acertividade era de 0,72%, ou seja, de todos os clientes que eram contactados, menos de 1% resultava em uma venda. Com a aplicação do modelo de Regressão Logística utilizando balanceamento da base de treino esse número aumentou drásticamente. De todos os clientes contactados, espera-se que aproximadamente 10% resulte em uma venda. Além disso, utilizando-se o modelo, espera-se que aproximadamente 90% de todos os possíveis Clientes Venda sejam contactados. Por fim, esse modelo proporcionaria também uma redução dos custos, uma vez que geraria um mailing mais acertivo para contactação, reduzindo o volume de clientes contactados aleatoriamente.

Sendo assim, é possível concluir que o modelo de classificação encontrado, através da técnica de Regressão Logística, é de extrema relevância para a empresa, resultando em uma redução dos custos com o envio dos SMS e um aumento expressivo do volume de retorno desses contatos (Vendas).

Sugere-se para trabalhos futuros a utilização de outros modelos de classificação não abordados nesse trabalho. Além disso, outros pontos podem ser abordados em estudos futuros, como:

- Adoção de outras linguagens de programação/bibliotecas onde seja permitido a utilização das variáveis categórias sem a necessidade de conversões para numéricas;
- Aplicação de outras metodologias de mineração de dados, como por exemplo, Algoritmos de Agrupamento;
- Desenvolvimento e avaliação de um modelo que se auto alimente com novas informações reais geradas, aprimorando seu resultado e assertividade.

Referências Bibliográficas

[1] Abreu, L. E. de: *People Analytics: uso de árvores de decisão na retenção de talentos*. Trabalho de Conclusão de Curso, Universidade Estadual Paulista (Unesp), 2022.

[2] Barella, V. H.: *Técnicas para o problema de dados desbalanceados em classificação hierárquica*. Dissertação de Mestrado, Universidade de São Paulo, 2015.

[3] Braga, A. C.: *Curvas ROC: aspectos funcionais e aplicações*. Tese de Doutoramento, Universidade do Minho, 2001.

[4] Cabral, C. I. S.: *Aplicação do modelo de regressão logística num estudo de mercado*. Dissetação de Mestrado, Universidade de Lisboa, 2013.

[5] Carter, R.: *A lista definitiva de estatísticas de Big Data para 2023*, 2022. https://findstack.com.br/resources/big-data-statistics/, acessado em 18/01/2023.

[6] Chawla, N. V., Bowyer, K. W., Hall, L. O. e Kegelmeyer, W. P.: *SMOTE: synthetic minority over-sampling technique*. Journal of artificial intelligence research, 16:321–357, 2002.

[7] Cister, A. M.: *Mineração de Dados para a Análise de Atrito em Telefonia Móvel*. Tese de Doutoramento, Universidade Federal do Rio de Janeiro, 2005.

[8] Fayyad, U. M., Piatetsky-Shapiro, G., Smyth, P. *et al.*: *Knowledge Discovery and Data Mining: Towards a Unifying Framework*. Em *KDD*, vol. 96, pp. 82–88, 1996.

[9] Figueiredo, C. N. L.: *Identificação de delirium em contexto hospitalar através de algoritmos de machine learning*. Dissertação de Mestrado, Universidade do Moinho, 2022.

[10] Fontana, É.: *Introdução aos algoritmos de aprendizagem supervisionada*. Departamento de Engenharia Química, Universidade Federal do Paraná, 2020.

[11] Friedman, J. H.: *Stochastic gradient boosting*. Computational statistics & data analysis, 38(4):367–378, 2002.

[12] Gama Neto, M. V. d.: *O processo CRISP-DM aplicado na construção de uma solução para Análise de Risco de Crédito*. Trabalho de Conclusão de Curso, Universidade Federal de Pernambuco, 2018.

[13] Garcia, S. C.: *O uso de árvores de decisão na descoberta de conhecimento na área da saúde*. Dissetação de Mestrado, Universidade Federal do Rio Grande do Sul, 2003.

[14] Goldschmidt, R. e Passos, E.: *Data mining: um guia prático*. Gulf Professional Publishing, 2005.

[15] Hurwitz, J. e Kirsch, D.: *Machine learning for dummies*. IBM Limited Edition, 75, 2018.

[16] John, G. H.: *Enhancements to the data mining process*. Stanford University, 1997.

[17] Lu, J.: *Cost-Sensitive Decision Trees for Imbalanced Classification*. https://johdev.com/jupyter/2020/02/26/Decision_Tree_Imbalance.html, acessado em 19/04/2023.

[18] Maione, C.: *Balanceamento de dados com base em oversampling em dados transformados*. Tese de Doutoramento, Universidade Federal de Goiás, 2020.

[19] Mariano, D. C. B.: *Data Mining*. Sagah, 2020.

[20] Petermann, R. J. *et al.*: *Modelo de mineração de dados para classificação de clientes em telecomunicação*. Dissertação de Mestrado, Pontifícia Universidade Católica do Rio Grande do Sul, 2006.

[21] Ribeiro, H. d. S.: *Classificação de clientes utilizando mineração de dados*. Trabalho de Conclusão de Curso, Pontifícia Universidade Católica de Goiás, 2020.

[22] Santos, H. G. d., Nascimento, C. F. d., Izbicki, R., Duarte, Y. A. d. O., Chiavegatto Filho, P. e Dias, A.: *Machine learning para análises preditivas em saúde: exemplo de aplicação para predizer óbito em idosos de São Paulo, Brasil*. Cadernos de Saúde Pública, 35:e00050818, 2019.

[23] SAS: *Mineração de Dados: O que é e qual sua importância?* https://www.sas.com/pt_br/insights/analytics/mineracao-de-dados.html, acessado em 19/01/2023.

[24] Schaeffer, A. G.: *Data Mining no Varejo: estudo de caso para loja de materiais de construção*. Dissertação de Mestrado, Universidade Federal do Rio Grande do Sul, 2003.

[25] Silva, D. F. B. F. d.: *Pré-processamento de Dados e Comparação entre Algoritmos de Machine Learning para a Análise Preditiva de Falhas em Linhas de Produção para o Controlo*. Dissertação de Mestrado, Instituto Superior de Engenharia do Porto, 2021.

[26] Silva, J. P. B. C. d.: *Modelos de Regressão Linear e Logística utilizando o software R*. Dissertação de Mestrado, Universidade Aberta, 2017.

[27] Singh, K.: *How to Improve Class Imbalance using Class Weights in Machine Learning*. https://www.analyticsvidhya.com/blog/2020/10/improve-class-imbalance-class-weights/, acessado em 16/04/2023.

[28] Smolski, F. M. d. S.: *Software R: curso avançado - Capítulo 7 Regressão Logística*. https://smolski.github.io/livroavancado/reglog.html, acessado em 30/01/2023.

[29] Soares, Ailton e Silvestre, I.: *Tutorial - Regressão Logística*. http://lea.estatistica.ccet.ufrn.br/tutoriais/regressao-logistica.html, acessado em 30/01/2023.

[30] Steiner, M. T. A., Soma, N. Y., Shimizu, T., Nievola, J. C. e Steiner Neto, P. J.: *Abordagem de um problema médico por meio do processo de KDD com ênfase à análise exploratória dos dados*. Gestão & Produção, 13:325–337, 2006.

[31] Van Rossum, G. e Drake Jr, F. L.: *The python language reference*. Python software foundation, 2014.

[32] Velasquez, L. H.: *Uma visão geral sobre machine learning*. https://statplace.com.br/blog/uma-visao-geral-sobre-machine-learning/, acessado em 25/01/2023.