UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO

CENTRO TECNOLÓGICO

DEPARTAMENTO DE TECNOLOGIA INDUSTRIAL

LUCAS LINHARES FRAGA DOS REIS

UTILIZAÇÃO DE TÉCNICAS DE MINERAÇÃO DE DADOS PARA OTIMIZAR OS LUCROS DE UMA CAMPANHA DE MARKETING

VITÓRIA

2022

LUCAS LINHARES FRAGA DOS REIS

UTILIZAÇÃO DE TÉCNICAS DE MINERAÇÃO DE DADOS PARA OTIMIZAR OS LUCROS DE UMA CAMPANHA DE MARKETING

Projeto de Pesquisa do Trabalho de Conclusão de Curso em Engenharia de Produção apresentado à Universidade Federal do Espírito Santo, sob a orientação do Prof. Rodolfo da Silva Villaça.

VITÓRIA

2022

RESUMO

A forma que decisões são tomadas por empresas está sendo cada vez mais apoiadas em dados, isso graças a evolução da tecnologia, que facilita o armazenamento e processamento de grandes quantidades de dados como nunca visto antes, de forma a viabilizar a criação de produtos de dados cada vez mais assertivos. Nesse cenário, esse trabalho propõe a utilização de técnicas de aprendizado de máquina para a otimização de uma campanha de marketing, através da segmentação dos clientes de uma empresa, buscando por padrões demográficos e firmográficos que indiquem a propensão de um cliente a aceitar ou não uma determinada oferta contida em uma campanha.

Palavras-chaves: Python; Análise de dados; Modelo de clusterização; Modelo de classificação; Segmentação de campanha; Marketing

ABSTRACT

The way that decisions are made by companies is increasingly being supported by data, thanks to the evolution of technology, which facilitates the storage and processing of large amounts of data as never seen before, in order to enable the creation of data products. increasingly assertive. In this scenario, this work proposes the use of machine learning techniques for the optimization of a marketing campaign, through the segmentation of a company's customers, looking for demographic and firmographic patterns that indicate the propensity of a customer to accept or not a particular offer contained in a campaign.

Palavras-chaves: Python; Data analysis; Clustering model; Classification model; Campaign segmentation; Marketing

LISTA DE ILUSTRAÇÕES

SUMÁRIO

1. INTRODUÇÃO

A constante busca das empresas por eficiência operacional, unida aos avanços tecnológicos que facilitaram o armazenamento e processamento de grandes quantidades de dados, tornaram possível a utilização de técnicas de aprendizado de máquina para diversos segmentos, não mais limitado a segmentos específicos ou puramente tecnológicos, havendo uma verdadeira onda de investimentos desse tipo também em segmentos considerados tradicionais (ZEN, 2022).

Um dos vários segmentos que encontrou ampla utilização das técnicas de aprendizado de máquina para otimizar seus resultados foi o de campanhas de marketing, uma campanha de marketing é um trabalho de promoção de uma empresa, produto, marca ou serviço, com a proposta de alcançar determinado objetivo relacionado à venda de um produto ou serviço (FERREIRA, 2018).

O direcionamento ao público-alvo tornou a utilização de aprendizado de máquina eficaz nesse segmento, com o uso de dados demográficos e firmográficos dos clientes, tornou-se possível encontrar padrões que permitem entender o quão propenso determinado cliente está a aceitar uma proposta feita por uma campanha. Assim, considerando que existe um custo associado a cada cliente impactado, a empresa pode optar ou não pela veiculação da campanha para determinado grupo de clientes, e assim, obter campanhas mais assertivas.

Neste cenário, ao fazer uso de uma base de dados pública e fictícia, esse estudo propõe a utilização de técnicas de aprendizado de máquina para otimizar a segmentação de uma campanha de marketing de uma empresa de entrega por aplicativo. Além disso, também é explorado o impacto financeiro gerado pelo uso dessas técnicas e como a taxa de conversão de clientes que aceitam a campanha se altera.

1.1 Objetivos

1.1.1. Objetivo geral

O presente estudo teve como objetivo geral utilizar técnicas de mineração de dados e inteligência artificial, para propor uma segmentação de clientes para uma campanha de marketing. Onde a ideia geral desta segmentação é aumentar a proporção de clientes que aceitarão a proposta feita por esta campanha. Os dados analisados neste trabalho são públicos e disponibilizados por uma empresa de entrega de alimentos de grande participação no mercado nacional. Tratam-se de dados fictícios que são utilizados pela empresa para testes em seu processo seletivo.

1.1.2. Objetivos Específicos

A fim de alcançar o objetivo geral, o presente estudo teve os seguintes objetivos específicos:

- Realizar o tratamento e pré-processamento dos dados brutos.
- Identificar quais variáveis se mostraram mais relevantes para a taxa de conversão da campanha de marketing através da análise exploratória dos dados e utilizando testes estatísticos.
- Aplicar técnicas de aprendizado de máquina visando construir um modelo de clusterização para entender características que grupos de clientes tem em comum.
- Aplicar técnicas de aprendizado de máquina visando construir um modelo preditivo que identifica grupos de clientes mais propensos a aceitar a campanha.
- Gerar uma recomendação de público que deverá receber a campanha, a fim de otimizar sua taxa de conversão.
- Simular o impacto financeiro desta otimização na campanha.

1.2. Metodologia

A base de dados utilizada neste trabalho faz referência a uma campanha piloto, veiculada por telefone, que visava entender o percentual de clientes, de uma amostra de 2240, que aceitaram a proposta feita pela campanha, onde era sabido informações sócio demográficas e firmográficas como idade, estado civil, renda, valores gastos e tempo como cliente.

Para a realização do pré-processamento, tratamento, análise e desenvolvimento do modelo de aprendizado de máquina, foi utilizado o ambiente Google Collaboratory (GOOGLE, 2022), a linguagem de programação Python 3 e suas bibliotecas, como Pandas (PANDAS, 2022), Plotly (PLOTLY, 2022) e SKLearn (SCKIT-LEARN, 2022), o código do trabalho está disponível para consulta online, através deste endereço[1].

1.3. Estrutura do Trabalho

A estrutura deste trabalho segue o seguinte formato:

- No capítulo 2 será apresentado o referencial teórico, onde serão abordados conceitos e ferramentas que tangem o trabalho sobre um ponto de vista técnico, tratando de assuntos como campanha de marketing, estruturação de problemas, técnicas de modelagem de dados e aprendizado de máquina.

- No capítulo 3 será apresentado o estudo de caso e o passo a passo de sua construção, onde são trazidos trechos de código utilizado e a aplicação de técnicas trazidas no referencial teórico.

- No capítulo 4 constará a conclusão, onde será retomada os objetivos do propostos, quais foram os resultados alcançados e uma proposta de trabalho futuro a ser realizado.

---

1 https://colab.research.google.com/drive/1Qd7USxDWWF2ZT2zBMcRbDEaDczucjRey?usp=sharing

2. REFERENCIAL TEÓRICO

Neste capítulo serão apresentados conceitos e ferramentas que tangem o trabalho.

2.1 Análise do problema

Atualmente, grande parte das empresas necessita de novas estratégias e modelos disruptivos para se manterem competitivos em um mercado cada vez mais acirrado. Algumas dessas estratégias se traduzem em frameworks, que permitem entender o problema e buscar inovações (CONNECTCOM). Para o nosso estudo, foi utilizado uma técnica chamada os 4Ps de Data, onde a ideia principal é garantir que o executor do projeto tenha contexto suficiente do problema a ser resolvido.

Para isso, algumas perguntas são respondidas antes da utilização de qualquer ferramenta de análise de dados, perguntas essas que podem ser observadas na Figura 1, como o Problema a ser resolvido, o Potencial que essa solução tratá, qual Produto será o melhor entregável e qual a Proposta de valor que o produto trará.

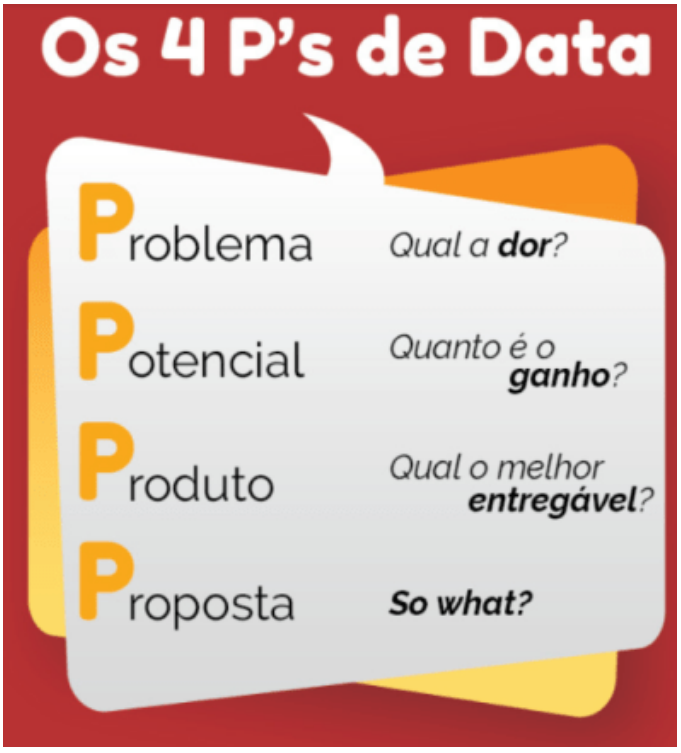


Figura 1: Visão geral sobre a metodologia 4P's de Data.
Fonte: Adaptado (PEDOTE, 2020)

Dessa forma, se tornam mais claros os objetivos do projeto, de onde estamos partindo e onde queremos chegar.

2.2 Segmentação de campanha

Os tipos de segmentação são cada vez mais utilizados na publicidade para criar campanhas assertivas. Com o uso de dados geográficos, demográficos, comportamentais e até mesmo psicológicos, é possível entender o perfil do público-alvo e o que ele deseja. A partir disso, a empresa consegue encontrar as melhores formas de oferecer seus produtos e serviços, com o objetivo de mostrar ao público que suas necessidades podem ser supridas (TRESMEIOS). Atualmente, o uso de aprendizado de máquina para identificação de públicos com maior probabilidade de aceitar a proposta feita pela campanha se torna cada vez mais comum, é tornou-se possível identificar características em comum de grupos de clientes e partir disso, fazer uma proposta de produto que faça sentido para esse público, de forma a reduzir os custos de veiculação da campanha e aumentar sua taxa de conversão.

2.3 Pré-processamento de Dados

Quando tratamos sobre pré-processamento de dados, existem várias técnicas e abordagens diferentes, neste subcapítulo, iremos explorar as técnicas que foram utilizadas neste estudo.

2.3.1. Identificação e remoção de Outliers utilizando Intervalo Interquartil

Outliers são registros de dados que tem características diferentes da distribuição padrão de uma base de dados(DINO, 2022), esses registros podem ter impacto negativo na estruturação de análises ou modelos, por se tratarem de um comportamento não usual de uma amostra ou até por se tratar de um erro de preenchimento. Dessa forma, faz-se necessário a identificação e remoção desses pontos de dados. Para isso, utilizamos uma técnica chamada intervalo interquartil,

que avalia a dispersão de dados somente depois de ordená-los em ordem crescente. O intervalo interquartil é calculado com base no cálculo de quartis, sendo o primeiro quartil (inferior), o quartil intermediário (mediana), o terceiro quartil (superior), que estão ligados ao conceito de quantil como pode ser observado na figura 2. A diferença entre o quartil superior e o quartil inferior determina o intervalo interquartil (WIKIPEDIA, 2022).

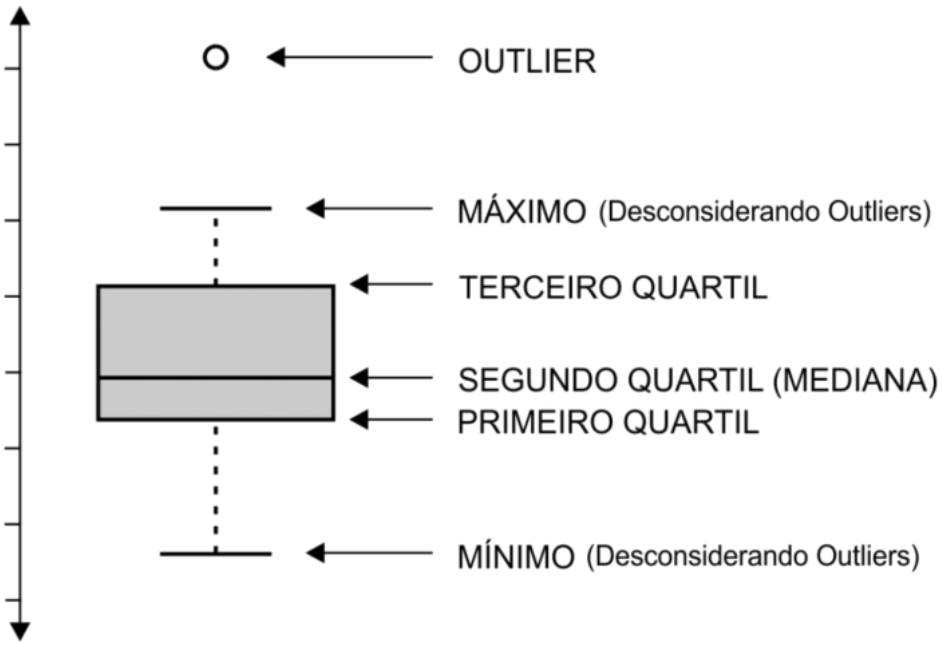


Figura 2: Diagrama de caixa utilizado para identificação de outliers.
Fonte: (YUKIO, 2018)

2.3.2. Engenharia de variáveis

A engenharia de variáveis é o processo de selecionar, manipular e transformar o dado bruto em variáveis que podem ser usadas (PATEL, 2021). É comum que a base de dados extraída inicialmente não possua todas as informações necessárias para a criação da análise ou do modelo para determinado tipo de problema, por isso, torna-se necessário criar novas variáveis utilizando as já disponíveis, como demonstrado na figura 3, onde com as variáveis Pé quadrado (Sq Ft.) e Valor (Amount) é construída a variável Custo por pé quadrado (Cost Per Sq Ft), que para algumas aplicações, pode ser mais útil do que as variáveis que a originaram, ampliando a possibilidade de resolução de problema dos dados sem a necessidade de fontes externas.

Sq Ft.	Amount	Cost Per Sq Ft
2400	9 Million	4150
3200	15 Million	4944
2500	10 Million	3950
2100	1.5 Million	510
2500	8.9 Million	3600

Figura 3: Exemplo de criação de uma variável utilizando outras pré existentes.
Fonte: (PATEL, 2021)

2.3.3. Análise exploratória de dados (EDA)

Em estatística, Análise Exploratória de Dados ou do inglês Exploratory Data Analysis (EDA) é uma abordagem para analisar bases de dados e extrair suas principais características, geralmente utilizando técnicas estatísticas e outros métodos de visualização de dados (WIKIPEDIA, 2022). A EDA é passo fundamental para qualquer projeto de dados, ela permite que a pessoa que estiver analisando se aprofunde nos dados que serão utilizados, de forma a extrair uma prévia das soluções que o modelo ou produto de dados trará depois de prontos. Como na figura 4, onde é possível através de um gráfico de dispersão utilizado na EDA, identificar que as duas variáveis, gorjeta (tips) e valor da conta (Total Bill), possuem um certo grau de correlação e uma pode ser dependente da outra.

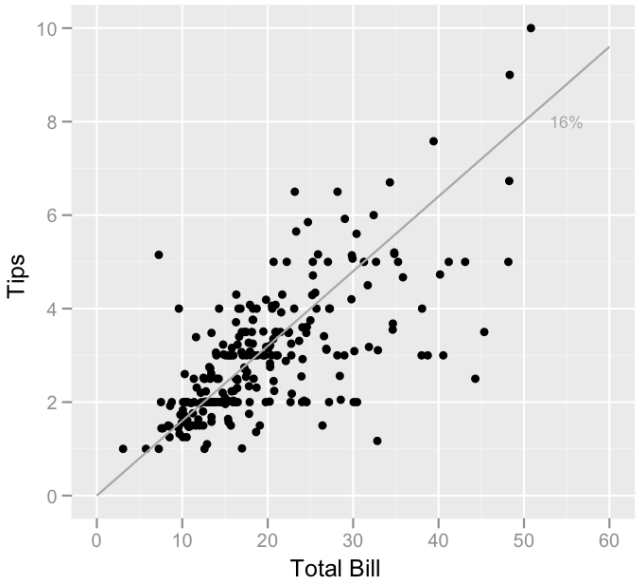


Figura 4: Gráfico de dispersão de valor da gorjeta vs valor total da conta.
Fonte: (WIKIPEDIA)

2.3.4. Redimensionamento

Quando seus dados possuem diferentes escalas, muitos algoritmos de aprendizado de máquina podem se beneficiar do redimensionamento para uma mesma escala. Normalmente isso é chamado de normalização e as variáveis são redimensionadas entre o intervalo de 0 e 1 (KUMAR, 2020). A utilização de redimensionamento é essencial para que o algoritmo de aprendizado de máquina não dê um peso maior para uma variável só porque essa variável possui valores maiores em sua escala, como por exemplo, se uma coluna estiver em gramas e outra em quilogramas, o algoritmo entenderá que 100 gramas é maior que 3 quilogramas, dado que a escala não está sendo considerada.

2.4 Teste Qui Quadrado

É um teste de hipóteses que se destina a encontrar um valor da dispersão para duas variáveis categóricas nominais e avaliar a associação existente entre variáveis qualitativas. O teste de Qui-Quadrado é considerado um teste não paramétrico, pois não depende de parâmetros populacionais (média e variância). O princípio básico deste teste é comparar proporções, ou seja, possíveis divergências entre as

frequências observadas e esperadas para um certo evento (GUIMARÃES).

O teste Qui-Quadrado é utilizado principalmente para identificar se a proporção da distribuição de uma variável em relação a outra desvia de forma significativa do que é esperado, identificando se as variáveis são independentes ou não entre si.

2.5 Clusterização (Kmeans)

K-Means é um algoritmo de clusterização (ou agrupamento) disponível na biblioteca Scikit-Learn do Python. É um algoritmo de aprendizado não supervisionado (ou seja, que não precisa de inputs de confirmação externos) que avalia e clusteriza os dados de acordo com suas características (ANASTACIO, 2020). Esse algoritmo funciona em alguns passos pré definidos, o primeiro deles é a definição da quantidade de clusters, depois disso, o algoritmo define o centróide de cada cluster aleatoriamente, após o centróide definido, ele calcula para cada ponto do espectro de dados, o centróide de menor distância, onde cada um desses pontos passará a pertencer a este centróide, com isso, acontece o reposicionamento do centróide, onde é definida pela média da posição de todos os pontos desse cluster, isso é repetido iterativamente, até que a posição ideal seja encontrada, e assim, os clusters são definidos. Como ilustrado na figura 5, onde uma amostra de dados é dividida em 3 clusters, onde cada ponto está dentro de uma região que faz referência a seu respectivo cluster e ali estão outros pontos com características semelhantes às suas.

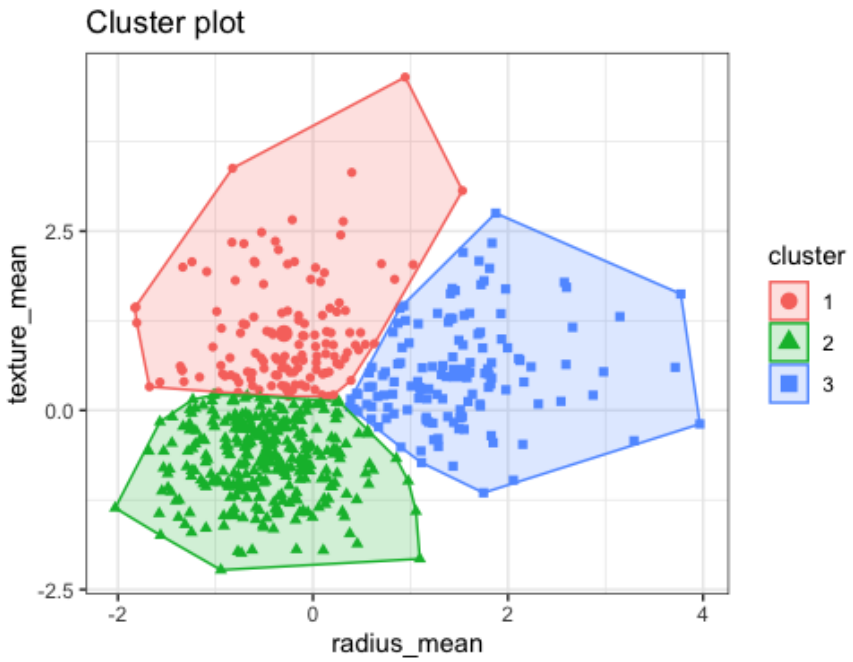


Figura 5: Gráfico ilustrando a distribuição de clusters.
Fonte: (BOOKDOWN)

2.6. Predição

2.6.1. Seleção de variáveis

Seleção de variáveis é o processo de reduzir o número de variáveis de entrada quando estamos desenvolvendo um modelo de aprendizado de máquina. Esse processo é realizado para reduzir o custo computacional da modelagem ou, em alguns casos, para aumentar a performance do modelo (BROWNLEE, 2019). Existem diversas técnicas para a seleção de variáveis, neste estudo, optamos pela seleção por correlação, onde é calculada a correlação entre as variáveis de entrada, e caso exista variáveis correlacionadas entre si, apenas uma é mantida, evitando assim a multicolinearidade, característica que pode impactar a performance do modelo, pois inclui informações redundantes. A análise visual desta técnica pode ser dada através da matriz de correlação como demonstrado na figura 6, onde variáveis muito correlacionadas entre si se aproximam de 1 e as poucos correlacionadas, se aproximam de 0.

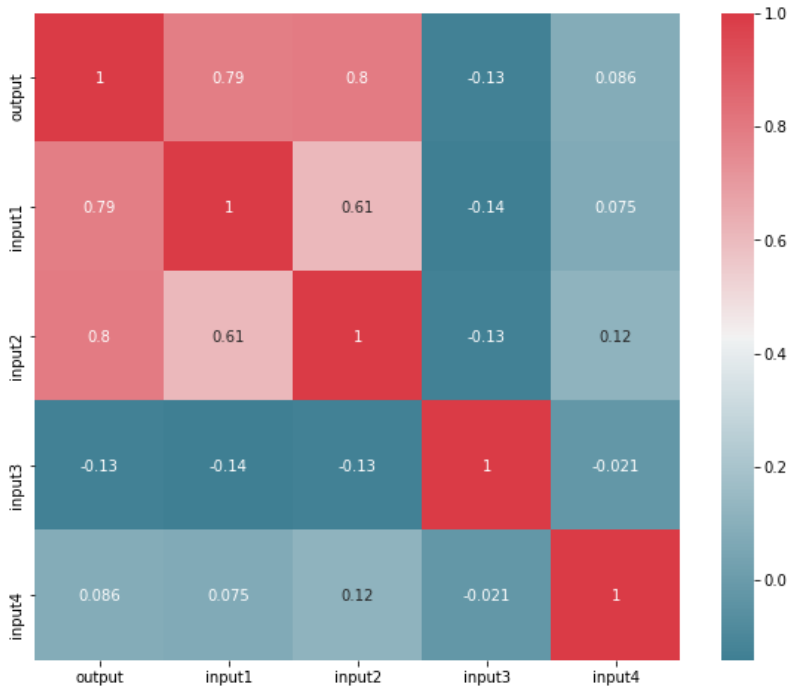


Figura 6: Matriz correlação entre variáveis de entrada.
Fonte: (LUNA, 2021)

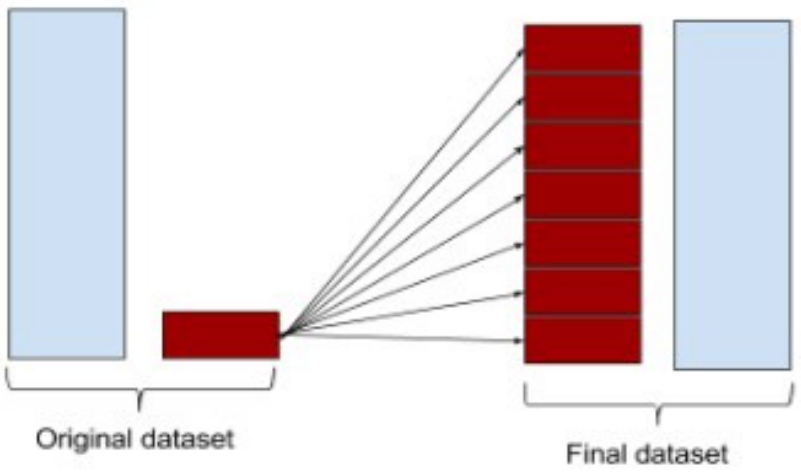
2.6.2. Separação dos dados

A separação dos dados ocorre tipicamente com sua divisão em duas partes, onde uma parte será usada para treinar e outra para testar o modelo (GILLIS, 2022), como ilustrado na figura 7. A separação dos dados é parte fundamental para o desenvolvimento de modelos de inteligência artificial, pois auxiliam na avaliação dos modelos, e o quão bem eles irão performar com dados externos, dado que ele é testado em dados nos quais não teve contato prévio. Não existe um valor determinado para que essa separação seja bem sucedida, mas boa parte dos modelos utilizam a separação onde que a base de teste tenha entre 10 e 30 por cento do tamanho total da base de dados.

Figura 7: Representação gráfica da separação dos dados em teste e treino.
Fonte: Adaptado (MAYO, 2020).

2.6.3. Superpopulação da base

Superpopulação tem como objetivo aumentar o peso da classe minoritária, ao criar mais exemplos desta classe (KUO, 2018). Se torna necessário o uso de técnicas de superpopulação de bases, quando estamos diante de uma base de dados onde a variável principal está desbalanceada, ou seja, uma das classes possui muito mais amostras do que a outra, como por exemplo, para análise de fraudes em cartão de crédito, onde apenas uma minoria das transações são fraudulentas e graças a esse desbalanceamento, se não houver o tratamento ideal da base, o modelo pode indicar que todas as transações são não fraudulentas. Existem diversas técnicas de superpopulação de uma base de dados, mas neste estudo iremos focar na superpopulação aleatória, onde amostras da classe minoritária são replicadas aleatoriamente até que se atinja tamanhos iguais de amostras entre as classes, como ilustrado na figura 8.



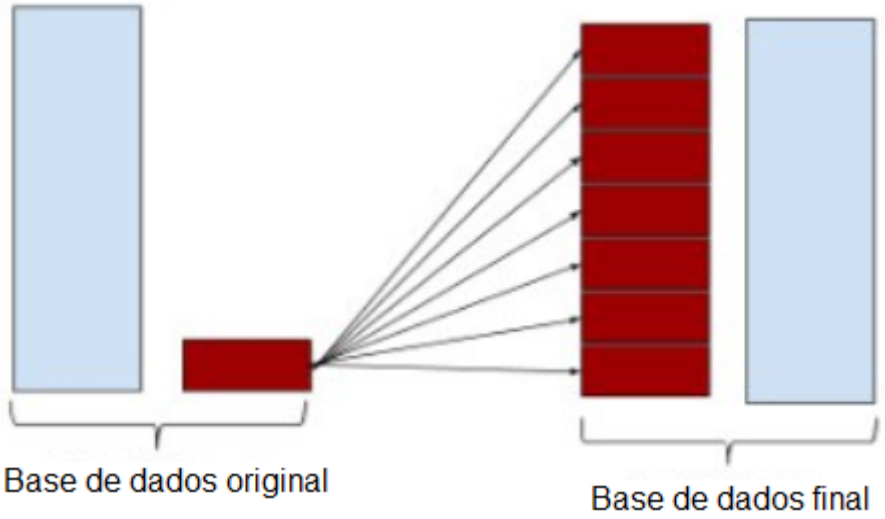


Figura 8: Representação gráfica da superpopulação da base.
Fonte: Adaptado (KUO, 2018)

2.6.4. Regressão Logística

A regressão logística é uma técnica estatística que tem como objetivo modelar, a partir de um conjunto de observações, a relação “logística” entre uma variável resposta dicotômica e uma série de variáveis explicativas numéricas (contínuas, discretas) e/ou categóricas. (CABRAL, 2013).

A regressão logística utiliza a curva logística para assim representar a relação entre a variável dependente e as independentes. Os valores previstos portanto permanecem entre 0 e 1, sendo definidos pelos coeficientes estimados (SMOLSKI). A figura 9 ilustra a curva logística que relaciona a probabilidade de um evento ocorrer e o nível da variável independente. Na regressão logística, a variável resposta é dicotômica, atribuindo-se o valor 1 ao acontecimento de interesse (sucesso) e 0 ao acontecimento complementar (insucesso)

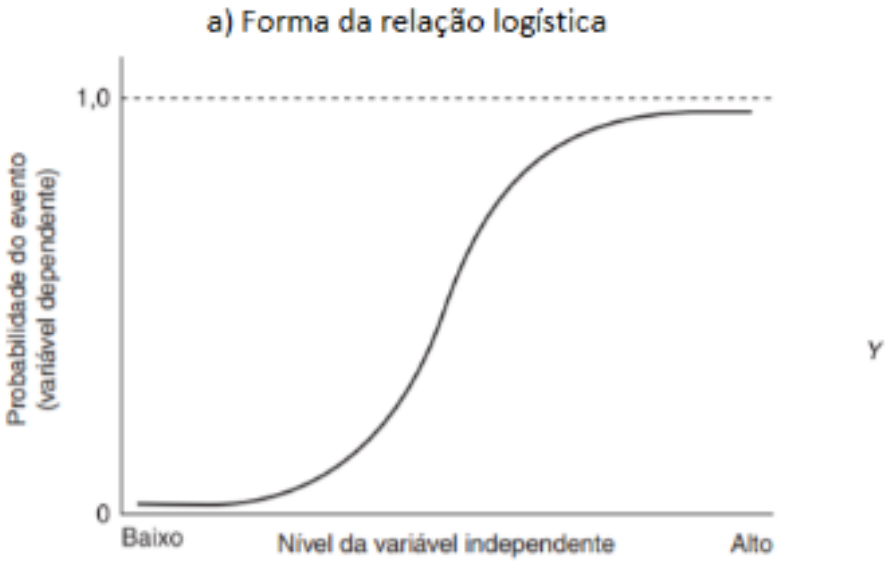


Figura 9: Curva logística. Fonte: (SMOLSKI)

Esta técnica é amplamente utilizada pelo seu ótimo custo benefício em relação à precisão e interpretabilidade, diferente de outras técnicas de aprendizagem de máquina, não se trata de um algoritmo “caixa preta”, onde não é possível explicar com clareza os fatores que influenciaram sua decisão.

2.7 Avaliação do modelo

Uma das fases fundamentais na criação de um modelo de aprendizado de máquina é a sua avaliação, essa avaliação é feita utilizando diversas métricas e técnicas e busca entender as fortalezas e fraquezas do modelo, de forma a definir se o modelo está bom o suficiente para ser usado para uma tomada de decisão.

2.7.1. Matriz confusão

A matriz confusão é uma ferramenta que auxilia na identificação o quão bem o modelo está prevendo os casos apresentados. Trata-se de uma tabela que mostrará as frequências de previsões corretas e erradas para cada classe do modelo como ilustrado na figura 10. A tabela irá mostrar a frequência de quatro situações,

Verdadeiros positivos (TP, do inglês True Positive), Falsos positivos (FP do inglês False Positive), Falsos Verdadeiros (TN, do inglês True Negative) e Falsos negativos (FN, do inglês False Negative). Onde casos de verdadeiros positivos ocorrem quando no conjunto real, a classe que estamos buscando foi prevista corretamente, enquanto que casos de falsos positivos, ocorrem quando no conjunto real, a classe que estamos buscando prever foi prevista incorretamente, já para os casos de falsos verdadeiros, ocorrem quando no conjunto real, a classe que não estamos buscando prever foi prevista corretamente, e por último, para os casos de falsos negativos, ocorrem quando no conjunto real, a classe que não estamos buscando prever foi prevista incorretamente (SOUZA, 2019).

Para cada tipo de aplicação busca-se minimizar ou maximizar determinado tipo de relação previsão x realidade, buscando um equilíbrio que faça sentido para o negócio, como por exemplo, para fraude bancária, o objetivo é ter o menor número de falsos negativos possível, visto que uma transação fraudulenta classificada como verdadeira, pode ser mais impactante negativamente do que uma transação não fraudulenta que foi negada.

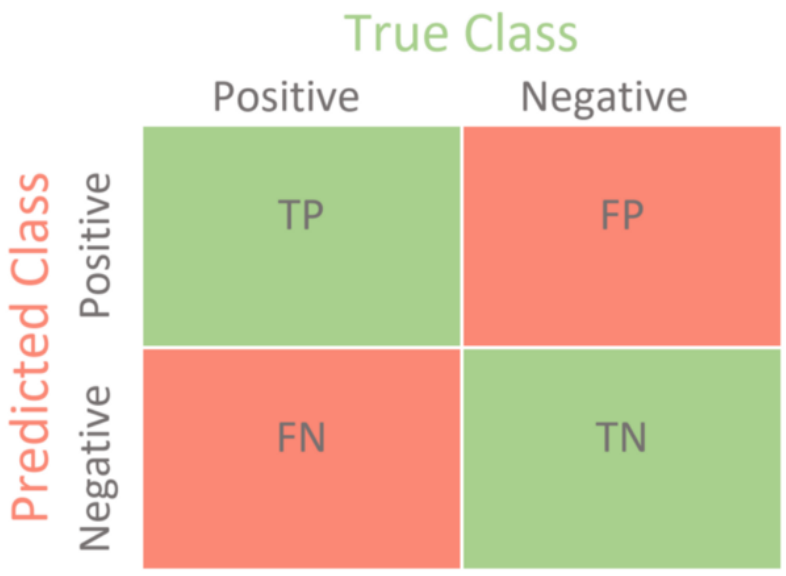


Figura 10: Matriz Confusão. Fonte: (MOHAJON, 2020)

2.7.2. Métricas de avaliação do modelo

Após o modelo pronto, torna-se necessário checar métricas que avaliam o quão preciso o modelo está, para isso, quatro principais métricas são utilizadas, são elas, Acurácia, Precisão, Recall e F1-Score.

A acurácia é a quantidade de acertos do modelo dividido pelo total da amostra (PÁDUA), com ela calculamos o quão certo um modelo está, a fórmula para seu cálculo pode ser demonstrada na figura 11.

$$Acurácia = \frac{Verdadeiros\ Positivos\ (TP) + Verdadeiros\ Negativos\ (VN)}{Total}$$

Figura 11: Fórmula da acurácia. Fonte: (PÁDUA, 2020)

Enquanto que a precisão é a quantidade de previsões classificadas como positivas, que realmente são positivas, a fórmula para seu cálculo pode ser demonstrada na figura 12.

$$Precisão = \frac{Verdadeiros\ Positivos\ (TP)}{Verdadeiros\ Positivos\ (TP) + FalsosPositivos\ (FP)}$$

Figura 12: Fórmula da Precisão. Fonte: (PÁDUA, 2020)

Já o Recall é a porcentagem de dados classificados como positivos comparado com a quantidade real de positivos que existem em nossa amostra (PÁDUA), a fórmula para seu cálculo pode ser demonstrada na figura 13.

$$Recall = \frac{Verdadeiros\ Positivos\ (TP)}{Verdadeiros\ Positivos\ (TP) + Falsos\ Negativos\ (FN)}$$

Figura 13: Fórmula do Recall. Fonte: (PÁDUA, 2020)

E por último, o F1 - Score é a métrica que une precisão e recall a fim de trazer um número único que determine a qualidade geral do nosso modelo (PÁDUA), a fórmula para seu cálculo pode ser demonstrada na figura 14.

$$F1 = \frac{2 * precisão * recall}{precisão + recall}$$

Figura 14: Fórmula do F1 - Score. Fonte: (PÁDUA, 2020)

Para cada uma dessas métricas, valores considerados bons, geralmente são determinados de acordo com alguma referência pré existe, de forma a variar bastante para cada contexto.

2.7.3. Validação cruzada

Normalmente antes de treinar nosso modelo separamos de forma aleatória nossos dados em base de treino e teste. Após o modelo treinado utilizamos a base de teste para avaliar o desempenho do modelo utilizando dados nunca antes vistos pelo mesmo. É uma boa abordagem porém pode não ser suficiente para avaliar o modelo e é nessa hora que entra a Validação Cruzada (RABELLO, 2019). Um dos métodos para a Validação Cruzada, é o K-fold, onde os dados são de teste e treino são particionados, porém eles não se resumem apenas a partição inicial, eles podem obter diversas partições como demonstrado na figura 15, e para cada uma dessas configurações, as métricas de qualidade são medidas e o resultado final, onde antes era apenas em relação a uma partição, agora é a média de todas as partições definidas, trazendo assim uma maior confiabilidade do resultado final do modelo.

Figura 15: Representação gráfica validação cruzada. Fonte: (RABELLO, 2019)

2.8. Importância das variáveis

A importância de variáveis é um método comum para os resultados de um modelo de aprendizado de máquina interpretáveis, essa técnica permite que tenhamos uma visão geral da tomada de decisão do modelo (SERENGIL, 2021). Para cada variável é calculado o quanto ela impactou na decisão do modelo, como demonstrado na figura 16, o que ajuda a direcionar acionáveis quando ações humanas serão tomadas baseadas no resultado de um modelo.

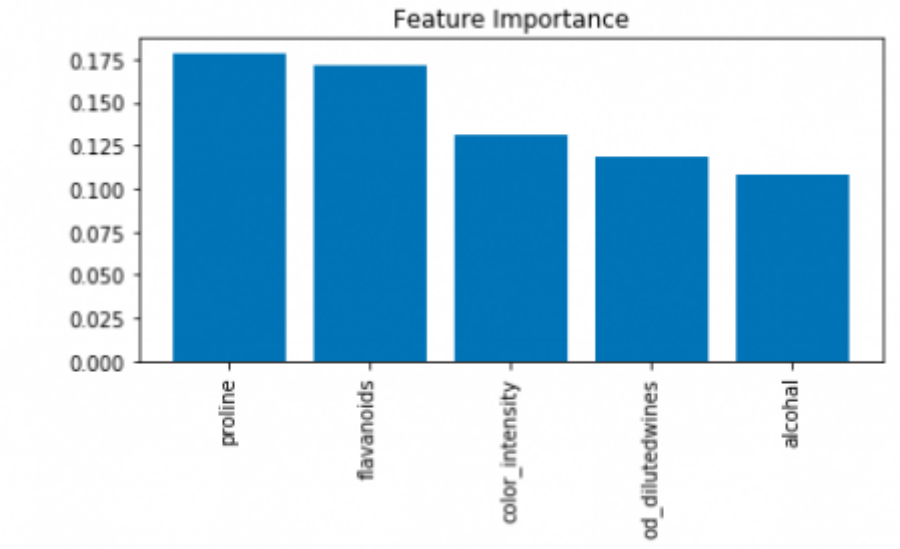


Figura 16: Gráfico de variáveis e seu respectivo grau de importância para o modelo
Fonte: (SERENGIL, 2021)

3. ANÁLISE DE DADOS

Neste capítulo detalhamos os procedimentos realizados, desde a limpeza da base bruta de dados, como as análises estatísticas e a criação do modelo que visa criar uma melhor segmentação para uma campanha de marketing.

3.1 Conjunto de dados

A base de dados analisada faz referência a uma campanha piloto de marketing. Essa base possui 2240 entradas, onde cada entrada faz referência a um cliente, e

25 colunas contendo informações demográficas e firmográficas, sendo essas apresentadas na figura 17, juntamente com sua respectiva descrição.

Feature	Description
AcceptedCmp1	1 if customer accepted the offer in the 1st campaign, 0 otherwise
AcceptedCmp2	1 if customer accepted the offer in the 2nd campaign, 0 otherwise
AcceptedCmp3	1 if customer accepted the offer in the 3rd campaign, 0 otherwise
AcceptedCmp4	1 if customer accepted the offer in the 4th campaign, 0 otherwise
AcceptedCmp5	1 if customer accepted the offer in the 5th campaign, 0 otherwise
Response (target)	1 if customer accepted the offer in the last campaign, 0 otherwise
Complain	1 if customer complained in the last 2 years
DtCustomer	data of customer's enrollment with the company
Education	customer's level of education
Marital	customer's marital status
Kidhome	number of small children in customer's household
Teenhome	number of teenagers in customer's household
Income	customer's yearly household income
MntFishProducts	amount spent on fish products in the last 2 years
MntMeatProducts	amount spent on meat products in the last 2 years
MntFruits	amount spent on fruits products in the last 2 years
MntSweetProducts	amount spent on sweet products in the last 2 years
MntWines	amount spent on wines products in the last 2 years
MntGoldProds	amount spent on gold products in the last 2 years
NumDealsPurchases	number of purchases made with discount
NunCatalogPurchases	number of purchases made using catalog
NunStorePurchases	number of purchases made directly in stores
NumWebPurchases	number of purchases made through company's web site
NumWebVisitsMonth	number of visits to company's web site in the last month
Recency	number of days since the last purchase

Figura 17: Variáveis e suas respectivas descrições. Fonte: (IFOOD, 2022)

3.2. Pré processamento dos dados

3.2.1. Padronização dos nomes das variáveis

Para a fase de pré-processamento dos dados, inicialmente foi necessário renomear algumas colunas, para nomes mais intuitivos e padronizados, as alterações feitas

podem ser verificadas no trecho do código demonstrado na figura 18.

```
# change some columns names to a easier understanding
cols_to_rename_dict = {
    'mntwines':'spent_on_wine',
    'mntfruits':'spent_on_fruit',
    'mntmeatproducts':'spent_on_meat',
    'mntfishproducts':'spent_on_fish',
    'mntsweetproducts':'spent_on_sweet',
    'mntgoldprods':'spent_on_gold_products',
    'numdealspurchases':'purchases_deals',
    'numwebpurchases':'purchases_web',
    'numcatalogpurchases':'purchases_catalog',
    'numstorepurchases':'purchases_store',
    'numwebvisitsmonth':'visits_web_month',
    'acceptedcmp1':'accepted_cmp_1',
    'acceptedcmp2':'accepted_cmp_2',
    'acceptedcmp3':'accepted_cmp_3',
    'acceptedcmp4':'accepted_cmp_4',
    'acceptedcmp5':'accepted_cmp_5',
    'z_costcontact':'cost_contact',
    'z_revenue':'revenue'
    }
df = df.rename(columns = cols_to_rename_dict)
```

Figura 18: Trecho de código usado para renomear as colunas.

3.2.2. Identificação de registros duplicados

Após a renomeação das colunas, foi procurado por entradas duplicadas ou identificadores duplicados, como demonstrado na figura 19, trata-se de um problema comum em base de dados, porém para o nosso caso, não foi encontrado nenhum registro duplicado.

```
# search for duplicated ids
duplicated_count_id = df[df.duplicated(subset=['id'])].shape[0]
print('There are', duplicated_count_id, 'duplicated rows on id column.')
```

```
There are 0 duplicated rows on id column.
```

Imagem 19: Trecho de código que busca por registros duplicados na base de dados

3.2.3. Padronização de formato das colunas

O próximo passo do pré processamento de dados foi garantir que todas as colunas estão no formato certo, para isso é necessário uma análise individual das colunas para entender se seu formato atual realmente corresponde ao tipo de informação inserida, para o nosso caso, apenas a coluna “dt_custumer” estava errada, coluna essa que faz referência a data em que a pessoa passou a ser cliente da empresa, e estava em formato “object” enquanto o formato correto deveria ser “datetime”, a figura 20 demonstra trecho de código utilizado para essa transformação.

```
# convert Dt_Customer to datetime
df['dt_customer'] = pd.to_datetime(df['dt_customer'])
```

Figura 20: Trecho de código usado para alterar o formato da coluna “dt_customer” .

3.2.4. Geração de novas variáveis

Como próximo passo, foi-se utilizada a técnica de engenharia de variáveis, que consiste em gerar novas variáveis utilizando outras variáveis já disponíveis, essa é uma técnica que se faz importante, pois com ela é possível obter informações que até então estavam implícitas na base de dados. No total, 16 novas variáveis foram criadas, a figura 21 traz o exemplo de uma das variáveis criadas, as demais podem ser encontradas no código completo, através deste endereço[1].

1 https://colab.research.google.com/drive/1Qd7USxDWWF2ZT2zBMcRbDEaDczucjRey?usp=sharing

```
# create total ammount of money spent by a customer
df['spent_total'] = df['spent_on_wine']\
                  + df['spent_on_fruit']\
                  + df['spent_on_meat']\
                  + df['spent_on_fish']\
                  + df['spent_on_sweet']
```

Figura 21: Trecho de código usado para criar uma nova variável utilizado variáveis pré existentes.

3.2.5. Identificação de registros inconsistentes

Após essas variáveis serem criadas, foi possível verificar algumas inconsistências em nossa base de dados, essas inconsistências faziam referência ao valor gasto em determinada categoria de produto, de forma que esse valor não poderia ser negativo, a figura 22 demonstra o trecho como foi feita essa identificação e limpeza.

```
# delete inconsistent rows on spent metrics
df_before_spent_inconsistencys = df.copy()
df = df[df['spent_on_regular_products'] >= 0]
print(df_before_spent_inconsistencys.shape[0] - df.shape[0], 'incosistent rows were deleted.')
```

```
4 incosistent rows were deleted.
```

Figura 22: Trecho de código que retira registros inconsistentes da base.

3.2.6. Identificação de registros nulos

Outro passo muito importante na realização do tratamento de dados, é o tratamento de valores nulos, nesse passo buscamos entender em quais colunas possuíamos valores nulos, quantos eram e qual sua representatividade em relação a base total. Com esses registros identificados, dada a baixa representatividade, foi feita a sua exclusão, como demonstrado no trecho de código representado na figura 23.

```
# check which columns contains missing values
df.columns[df.isnull().any()].tolist()
```

```
['income', 'avg_pct_of_income_spent_monthly']
```

```
# As income is the only raw that have missing values (avg_pct_of_income_spent_monthly was made from income), let's deep dive on it
# check how many rolls do we have
null_rows_count = df[df['income'].isnull()].shape[0]
total_rows_count = df['income'].shape[0]
null_rows_pct = null_rows_count/total_rows_count
null_rows_pct = round(null_rows_pct*100,2)
print('There are', null_rows_count, 'rows containing missing values in Income column, which represents ', null_rows_pct, '% of total data points.')
```

```
There are 23 rows containing missing values in Income column, which represents  1.03 % of total data points.
```

```
df = df.dropna('any')
```

Figura 23: Trecho de código que retira registros nulos da base.

3.2.7. Identificação de dados fora do padrão (outliers)

Como último passo da limpeza e tratamento de dados, faz-se muito importante a identificação e tratamento de outliers, para isso, inicialmente, foi feita uma análise da descrição estatística da base, representada na figura 24, onde é possível observar a quantidade de registros, a média de seus valores, seu desvio padrão, seu valor mínimo, seu valor máximo e os percentis 25, 50 e 75, como descrito na figura abaixo.

```
# check if there is something unusual on data, especially comparing min and max values with mean and 50% percentile
df.describe().transpose()
```

	count	mean	std	min	25%	50%	75%	max
years_of_study	2229.0	18.797667	4.432404	9.000000	17.000000	17.000000	20.000000	26.000000
income	2229.0	52288.719157	24812.230638	2447.000000	35441.000000	51390.000000	68487.000000	666666.000000
age	2229.0	45.209062	11.986089	18.000000	37.000000	44.000000	55.000000	121.000000
is_together	2229.0	0.646478	0.478170	0.000000	0.000000	1.000000	1.000000	1.000000
total_sons	2229.0	0.951996	0.751379	0.000000	0.000000	1.000000	1.000000	3.000000
is_parent	2229.0	0.716465	0.450815	0.000000	0.000000	1.000000	1.000000	1.000000
family_size	2229.0	2.598475	0.905210	1.000000	2.000000	3.000000	3.000000	5.000000
spent_on_wine	2229.0	305.321220	336.840098	0.000000	24.000000	177.000000	505.000000	1493.000000
spent_on_fruit	2229.0	26.412741	39.838511	0.000000	2.000000	8.000000	33.000000	199.000000

Figura 24: Trecho de código e resultado que gera o descritivo estatístico da base.

Mas sabe-se que essa análise descritiva é limitada e pouco precisa, sendo útil apenas para ter-se uma perspectiva geral, para uma análise mais completa e precisa, é necessário a utilização de técnicas estatísticas, e para esse estudo, foi utilizada a técnica de intervalo interquartil, com ela, foi identificada a presença de 15 registros que fogem do padrão, esses registros foram retirados da base, a sua função foi descrita na figura 25.

```python
# function that removes outliers from df
def remove_outliers_iqr(df_name, column_name, iqr_multiplier):
  # defining first and third quartile
  q1 = df_name[column_name].quantile(0.25)
  q3 = df_name[column_name].quantile(0.75)

  # creating inter quartile variable
  iqr = q3 - q1

  # creating lower and upper limit
  lower_limit = q1 - iqr_multiplier * iqr
  upper_limit = q3 + iqr_multiplier * iqr

  # remove outliers from df
  df_wo_outliers = df_name[~((df_name[column_name] < lower_limit) | (df_name[column_name] > upper_limit))]

  # inform how many columns were deleted
  rows_before = df_name.shape[0]
  rows_after = df_wo_outliers.shape[0]
  print(rows_before - rows_after, 'rows contains outliers in', column_name, 'column were deleted.')

  # returns df without outliers
  return df_wo_outliers
```

Figura 25: Trecho de código que identifica e remove dados fora do padrão.

3.3. Análise exploratória

Com o fim da fase de tratamento dos dados, tornou-se possível iniciar a análise exploratória, que tem como objetivo entender as características dos clientes em relação a aceitar ou não a campanha piloto veiculada. Para isso, todas as variáveis da base foram testadas para entender a taxa de aceite da campanha considerando as diferentes características dos clientes. Além disso, para cada uma das variáveis, foi realizado o teste Qui Quadrado a fim de entender se as diferenças observadas entre as taxas de sucesso da campanha para cada uma das categorias são dadas ao acaso ou não, onde o trecho de código que executa esse teste está representado na figura 26, neste estudo iremos apresentar as principais descobertas que essa análise proporcionou.

```python
# function that tests dependence between two variables
def chi2_test(df_name_c2, var1_c2, var2_c2, probability_c2, bins_c2):
  # bin column (as we are dealing with numeric columns)
  binned_column = pd.cut(df_name_c2[var2_c2], bins = bins_c2, include_lowest = True)

  # create contingency table
  contigency = pd.crosstab(index = df_name_c2[var1_c2],columns =  binned_column).to_numpy()

  # calculate chi2 test
  stat, p, dof, expected = chi2_contingency(contigency)

  # interpret chi2 test and return the result
  critical = chi2.ppf(probability_c2, dof)
  if abs(stat) >= critical:
    return 'Chi2 test result: Variables are dependent'
  else:
    return 'Chi2 test result: Variables are independent'
```

Figura 26: Trecho de código que realiza o teste Qui Quadrado.

Para as análises a seguir, utilizamos o padrão de roxo escuro para o grupo de clientes que aceitou a proposta da campanha, enquanto usamos o rosa claro para o grupo que não aceitou a proposta da campanha, a linha tracejada indica a média da taxa de sucesso da campanha, que é de 15%, além disso, abaixo da legenda do eixo X, é possível encontrar o resultado do teste Qui Quadrado para independência entre as variáveis.

A primeira variável testada foi escolaridade (medida em anos de estudo), para essa variável pode-se observar uma clara correlação positiva entre a quantidade de anos de estudo que o cliente possui e a taxa de aceite da campanha, onde o grupo de clientes de maior escolaridade, tendem a aceitar mais a campanha.

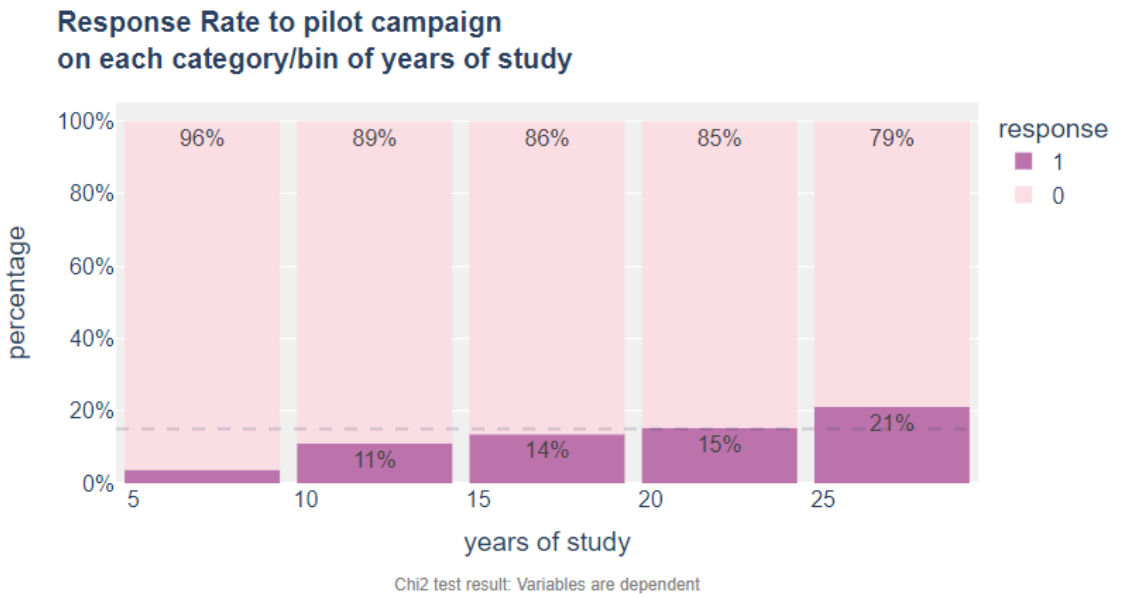


Figura 27: Gráfico que representa a taxa de resposta a campanha para cada categoria da variável anos de estudo.

A próxima variável testada foi renda, onde também foi possível encontrar uma correlação positiva entre a renda dos clientes e sua propensão a aceitar a campanha, onde para clientes de renda acima de $100.000/ano, observou-se que a taxa de sucesso da campanha era 100%.

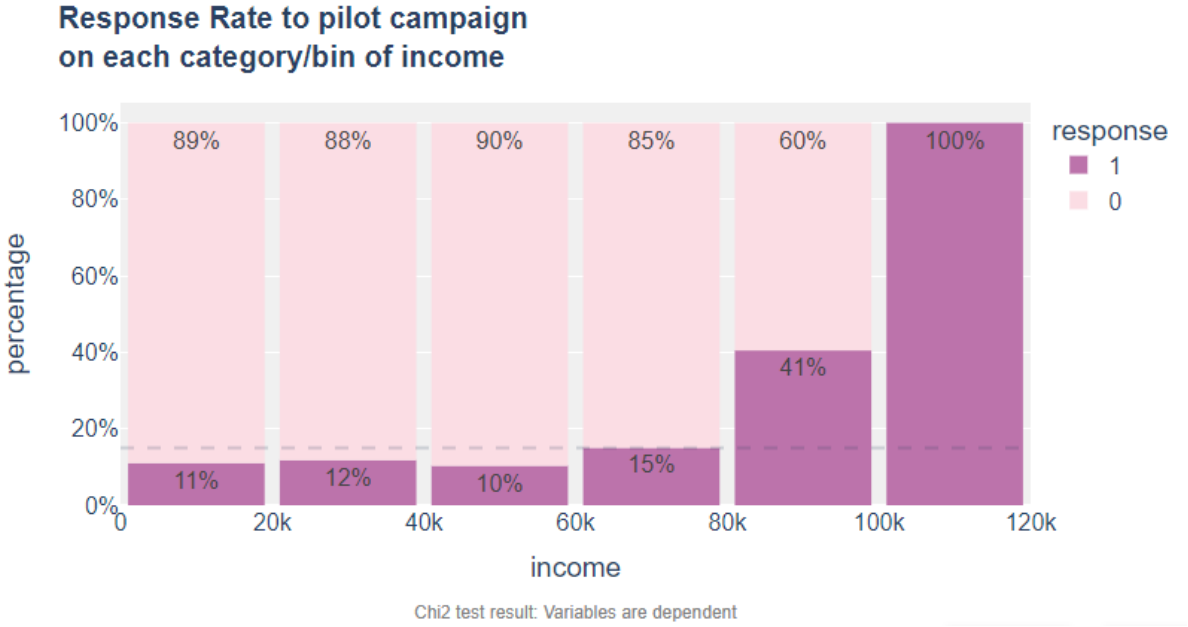


Figura 28: Gráfico que representa a taxa de resposta à campanha para cada categoria da variável renda.

Já a variável idade, se mostrou não significativa para determinar se o cliente aceitou ou não a campanha piloto, o teste Qui Quadrado indicou que as variáveis idade e taxa de aceite da campanha são independentes entre si, o que também pode ser observado na figura 29, onde para os clientes de 20 anos ou mais (grupos que concentram a maior parte da base de clientes), a taxa de aceite da campanha se mantém estável.

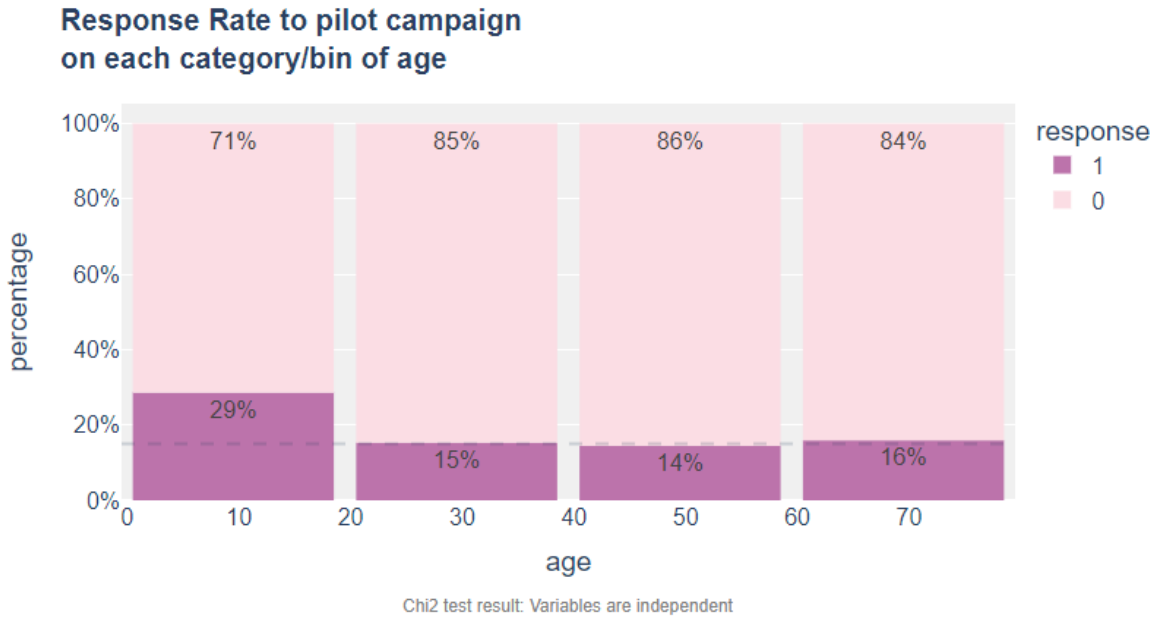


Figura 29: Gráfico que representa a taxa de resposta à campanha para cada categoria da variável idade.

Outra variável que também foi analisada, foi o tamanho da família do cliente, para essa variável, foi-se encontrada uma correlação negativa, onde clientes com famílias maiores, tendem a ter uma taxa de aceitação de campanha menor, como pode ser observado na figura 30.

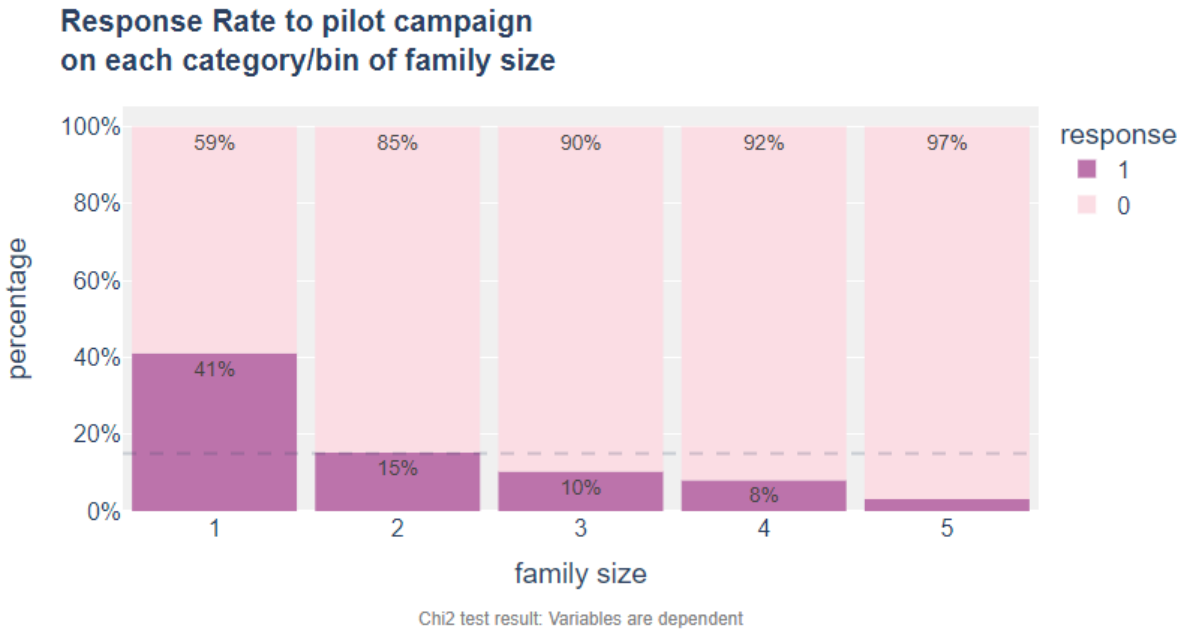


Figura 30: Gráfico que representa a taxa de resposta à campanha para cada categoria da variável tamanho da família.

A análise de todas as variáveis foi feita, porém só foram trazidas para este estudo as mais importantes, as outras análises podem ser acessadas através deste endereço[2].

3.4. Conclusões preliminares

Com o fim da análise exploratória, podemos tirar algumas conclusões sobre as características de grupos de clientes que tiveram uma melhor taxa de aceitação da campanha, como por exemplo, o grupo de clientes com 25 ou mais anos de estudo, o grupo de clientes com renda acima de $80.000/ano e o grupo de clientes com família de 1 pessoa, com essas informações, já torna-se possível gerar uma segmentação mais eficaz para a veiculação da campanha com o objetivo de atingir uma maior taxa de sucesso.

2 https://colab.research.google.com/drive/1Qd7USxDWWF2ZT2zBMcRbDEaDczucjRey?usp=sharing

3.5. Clusterização

Como próximo passo, iremos agrupar os clientes de acordo com suas características semelhantes, utilizando a técnica conhecida como K Means.

3.5.1. Estandardização

Para isso, torna-se necessário a uniformização da escala dos dados, dessa forma, utilizaremos a padronização com média 0 e desvio padrão 1, de forma que mantemos a média e a variância igual para todas as variáveis, e assim, nenhuma delas afetará o processo de clusterização por ter uma escala maior, resultando assim em agrupamentos mais justos. O trecho de código utilizado neste processo está descrito abaixo.

```
# Rescaling the attributes
df_scaled = df.copy()

# fit_transform
df_scaled = StandardScaler().fit_transform(df_scaled)

# get back values to df
df_scaled = pd.DataFrame(df_scaled)
df_scaled.columns = df.columns
```

Figura 31: Trecho de código utilizado para a estandardização dos dados.

3.5.2. Regra do cotovelo

Com as variáveis estandardizadas, é necessário definir o número ótimo de grupos que queremos obter, para isso, utilizamos uma técnica chamada Regra do Cotovelo, onde é plotado um gráfico da média ao quadrado das distâncias dos pontos ao centro do cluster, também conhecida como distorção, para cada número de clusters. Esse ponto ótimo de quantidade de clusters, é definido quando essa distorção começa a diminuir em um ritmo menos acelerado, se assemelhando ao formato de um cotovelo, para o nosso caso, em 2 clusters, como demonstrado na figura 32.

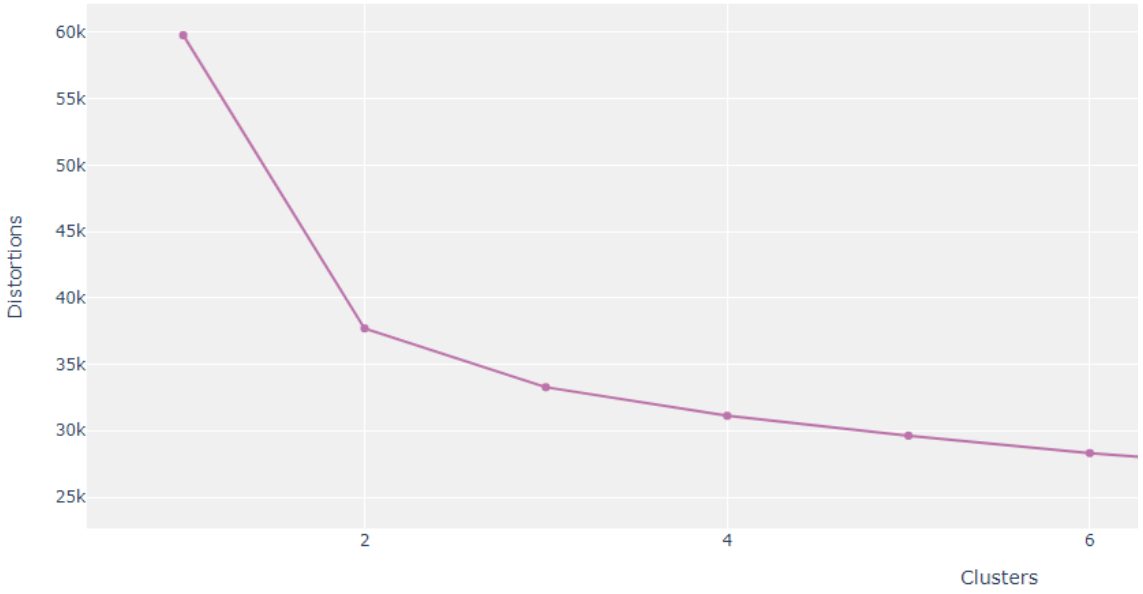


Figura 32: Gráfico utilizado na regra do cotovelo.

3.5.3. Características dos clusters

Após o número de Clusters ser definido, aplicamos a técnica de clusterização e verificamos o comportamento de cada um dos clusters, na figura 33, cada cluster está sendo representado por uma cor, onde o cluster 1 está sendo representado pela cor roxa e o cluster 2, está sendo representado pela cor rosa, além disso, cada uma das barras horizontais representa o valor médio daquele cluster para determinada variável e sua comparação percentual em relação ao outro cluster.

Foi possível identificar que para algumas variáveis, o valor médio dos dois clusters é pouco significativo, como idade e recência por exemplo, enquanto que para outras como renda e ticket médio, existe uma grande discrepância entre os dois clusters.

Após a análise das características de cada um dos clusters e entender que o cluster 1 consome mais que o cluster 2, decidimos alterar seus nomes para cluster Premium e Standard respectivamente.

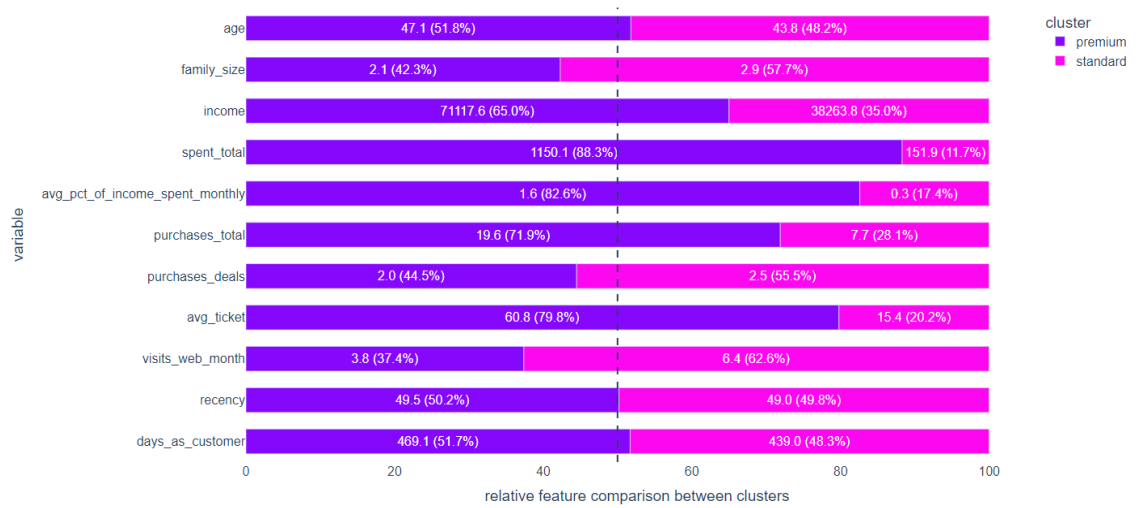


Figura 33: Gráfico utilizado para entender as principais características de cada cluster e como esses clusters se diferenciam entre si.

3.6. Modelagem

Agora com um melhor entendimento das características dos clientes e quais são os grupos com maior propensão a aceitar a campanha, iremos criar um modelo que irá prever quais clientes pertencentes ao cluster Premium possuem maior chance de aceitar a campanha. Para isso, iremos utilizar um modelo linear conhecido como regressão logística.

O modelo de regressão logística foi utilizado principalmente pela sua maior interpretabilidade, de forma que conseguimos traduzir os motivadores do resultado do modelo para uma pessoa tomadora de decisão em uma empresa, por exemplo.

3.6.1. Seleção de variáveis

Primeiramente, iremos definir as variáveis que serão utilizadas no modelo. Para isso, iremos utilizar a correlação linear entre as variáveis, para retirar as variáveis que estiverem altamente correlacionadas entre si (variáveis redundantes foram consideradas com coeficiente de correlação acima de 0,5), característica essa chamada de multicolinearidade, que acaba por reduzir o poder estatístico do modelo. Para isso, a função descrita na figura 34, remove todas as variáveis redundantes, ao identificar variáveis que possuem alta correlação entre si.

```python
# function that drops correlated features and keep the one wich is more correlated with response variable
def drop_correlated_features(df, response_var):
  # sort columns in order by the most correlated variables with response to less
  ix = df.corr().abs().sort_values(response_var, ascending=False).index
  df_sorted = df.loc[:, ix]

  # Create correlation matrix
  correl = df_sorted.corr().abs()

  # Select upper triangle of correlation matrix
  upper = correl.where(np.triu(np.ones(correl.shape), k=1).astype(bool))

  # Find index of feature columns with correlation greater than 0.5
  to_drop = [column for column in upper.columns if any(upper[column] > 0.5)]

  # Drop features
  df_cleaned = df_sorted.drop(to_drop, axis=1)

  # shape of final df
  shape = df_cleaned.shape
  # print shape of cleaned df and columns that we dropped
  print('After dropping highly correlated features, our has {} records and {} features'.format(shape[0], shape[1]))
  print('Dropped features: ', to_drop)
  return df_cleaned
```

Figura 34: Trecho de código utilizado para remoção de colinearidade.

Dessa forma, restaram 18 variáveis, entre as 41 iniciais, que servirão de entrada para o modelo, que estão descritas na figura 35, juntamente com seus respectivos coeficientes de correlação.

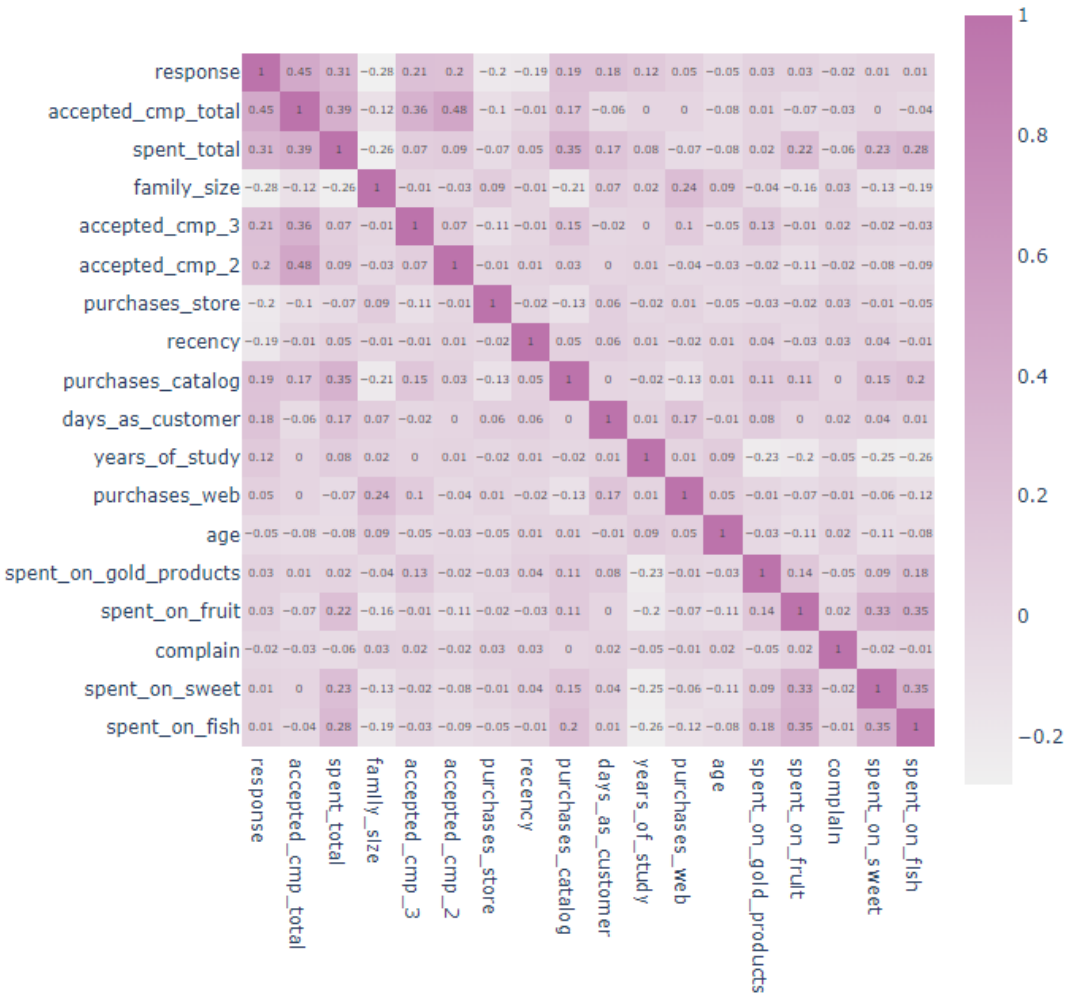


Figura 35: Gráfico de correlação das variáveis que serão utilizadas no modelo.

3.6.2. Separação da base

A próxima fase para a criação do modelo é a separação da base em base de treino e base de teste. Para isso, utilizamos uma proporção de 20% para teste e 80% para treino, como está descrito na figura 36 que demonstra o trecho de código do comando responsável por esta ação.

```python
# split test and train df
def split_data(df, test_size, random_state):
  train, test = train_test_split(df, test_size=test_size, random_state=random_state)
  print('Train Shape:', train.shape)
  print('Test Shape:', test.shape)
  return train, test
```

```python
# split test and train df for premium cluster
train_prem, test_prem = split_data(df_premium, test_size = 0.2, random_state = 10)
```

```
Train Shape: (724, 18)
Test Shape: (181, 18)
```

Figura 36: Trecho de código utilizado para separar a base em teste e treino.

3.6.3. Superpopulação da base

É comum que bases de dados possuam mais registros de uma categoria em relação a outra, quando isso acontece para a variável resposta, dizemos que a base está desbalanceada. A base utilizada neste estudo se enquadra nessa classificação e se trata de uma base desbalanceada, como pode ser observado na figura 37, temos 566 registros de clientes que não aceitaram a campanha e 158 registros de clientes que aceitaram a campanha.

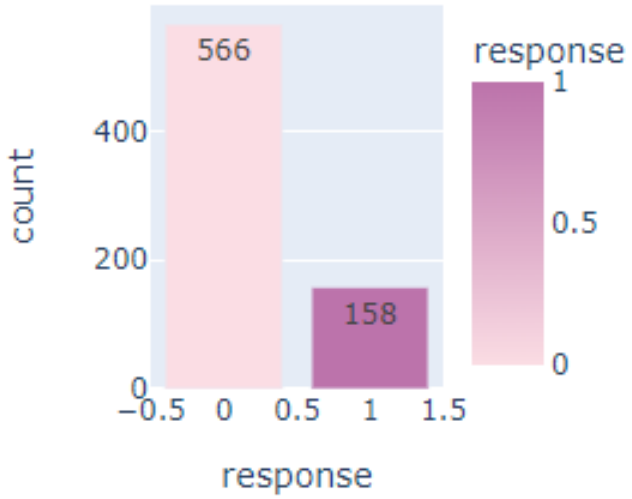


Figura 37: Distribuição da variável resposta antes da superpopulação da base.

Com isso, utilizamos a técnica de superpopulação randômica, onde amostras sintéticas são criadas no intuito de balancear a classe defasada, essas amostras são criadas ao repetir aleatoriamente registros da classe minoritária até que as duas classes possuam a mesma quantidade de registros. A função criada para esse processo pode ser observada na figura 38, e a distribuição final das categorias, com a base já balanceada, na figura 39.

```
# function that apply random oversampling method to our train data
def train_df_to_oversample(df_train, response_var):
  # define dataset x and y
  X = df_train.drop(response_var, axis = 1)
  y = df_train[response_var]

  # define oversampling strategy
  oversample = RandomOverSampler(sampling_strategy='minority')

  # fit and apply the transformation
  X_over, y_over = oversample.fit_resample(X, y)
  df_train = X_over.copy()
  df_train[response_var] = y_over.copy()

  return df_train
```

Figura 38: Trecho de código utilizado para criar a superpopulação da base.

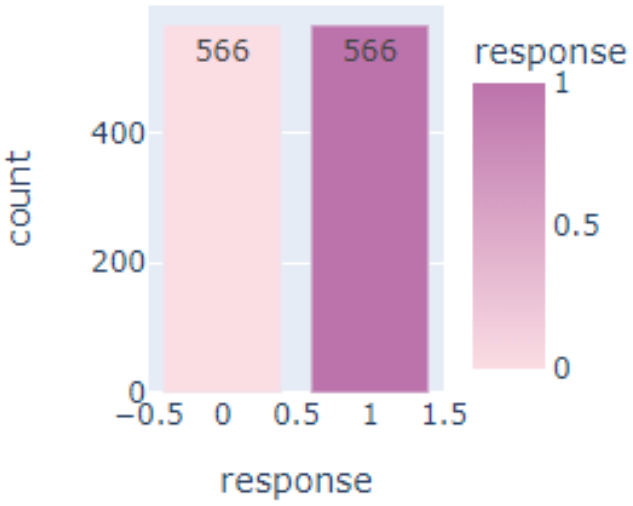


Figura 39: Distribuição da variável resposta depois da superpopulação da base.

3.6.4. Treinamento do modelo

Nessa etapa, com a base de treino já balanceada, treinamos o nosso modelo utilizando a técnica de regressão logística, sem a alteração dos parâmetros de entrada. Já para a validação do modelo, utilizamos a técnica de validação cruzada e a matriz confusão, obtendo como resultado acurácia de 83% com 16% de casos de falso positivo e 18% de falsos negativos como pode ser observado na figura 40.

```
Mean AUC score after a 5 fold cross validation:  0.9081406723795155
AUC score of each fold:  [0.89004037 0.91755939 0.91236589 0.90774532 0.9129924 ]
Accuracy:  0.8303886925795053
AUC:  0.915017979997253
Type 1 error:  0.15901060070671377 | False Positive: predicted customer accepted the campaign, while it was not!
Type 2 error:  0.18021201413427562 | False Negative - predicted response customer do not accepted campaign, while it was!
```

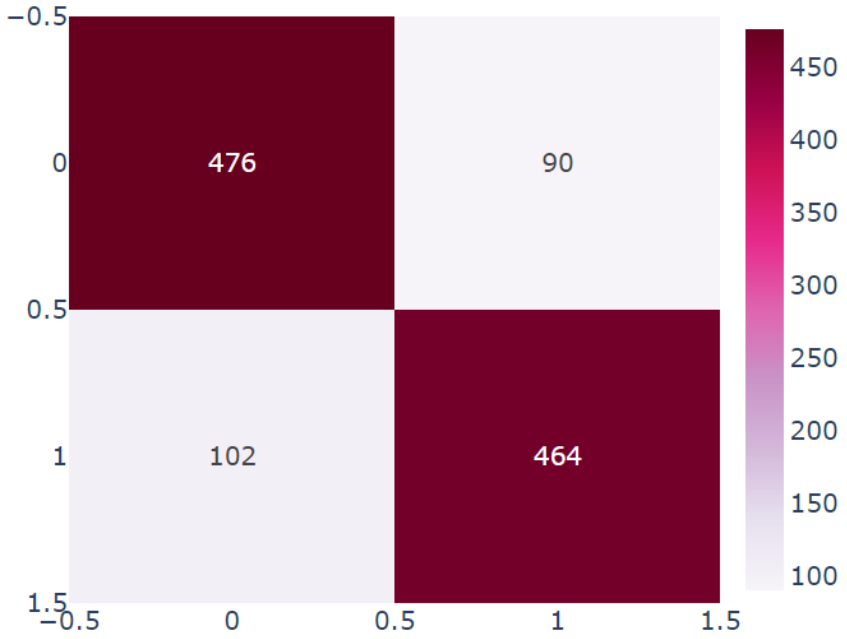


Figura 40: Plotagem da matriz confusão, assim como as métricas de qualidade do modelo para a base de treino.

3.6.5. Teste do modelo

Com o modelo treinado, o utilizamos a base de teste, que são dados até então novos para o modelo e sua tarefa será prever se o cliente aceitou ou não a campanha. Para os dados de treino, os resultados do modelo foram de 82% de acurácia, com 20% de casos de falso positivo e 14% de casos de falso negativo como pode ser visto na figura 41.

Como não houve grande variação entre os resultados obtidos na base de treino e na base de teste, podemos considerar que o modelo está suficientemente generalizado para lidar com casos reais.

```
Accuracy:  0.8176795580110497
AUC:  0.8943264764432648
Type 1 error:  0.19708029197080293 | False Positive: predicted customer accepted the campaign, while it was not!
Type 2 error:  0.13636363636363635 | False Negative - predicted response customer do not accepted campaign, while it was!
```

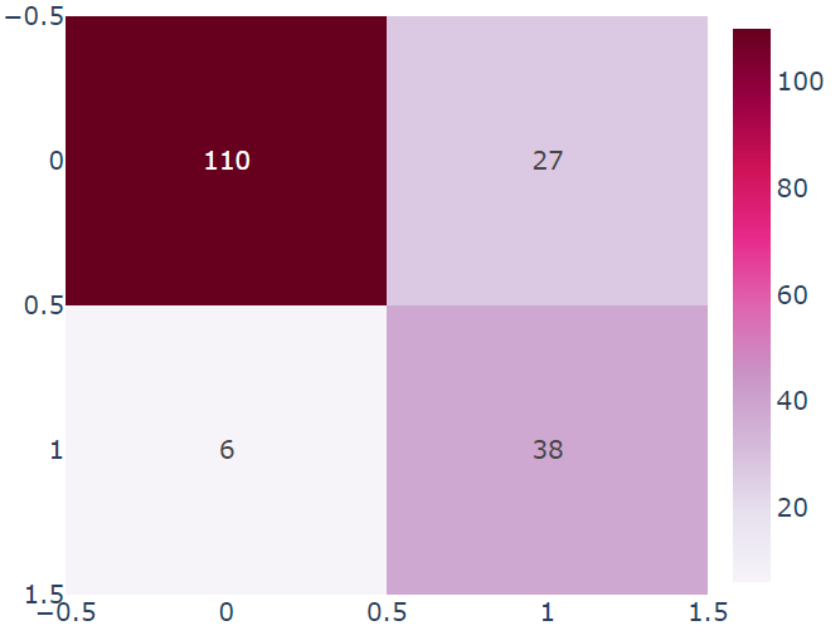


Figura 41: Plotagem da matriz confusão, assim como as métricas de qualidade do modelo para a base de teste.

3.6.6. Importância de variável

Outro passo que é fundamental para uma maior aceitação dos resultados do modelo, principalmente por pessoas não técnicas, é a importância da variável, que explica quais variáveis têm maior impacto para a tomada de decisão sugerida pelo modelo, para o nosso caso, como pode ser observado na figura 42, a variável de maior importância foi a que faz referência ao tamanho da família, seguida pelas variáveis que indicam se o cliente aceitou campanhas passadas ou não, para esse gráfico, é importante ressaltar que o valor da importância da variável é medido em módulo e os valores negativos indicam que a variável estudada é inversamente proporcional ao resultado positivo da variável resposta, como é o caso do tamanho da família, onde quanto menor o tamanho da família, maior a probabilidade da

pessoa aceitar a campanha.

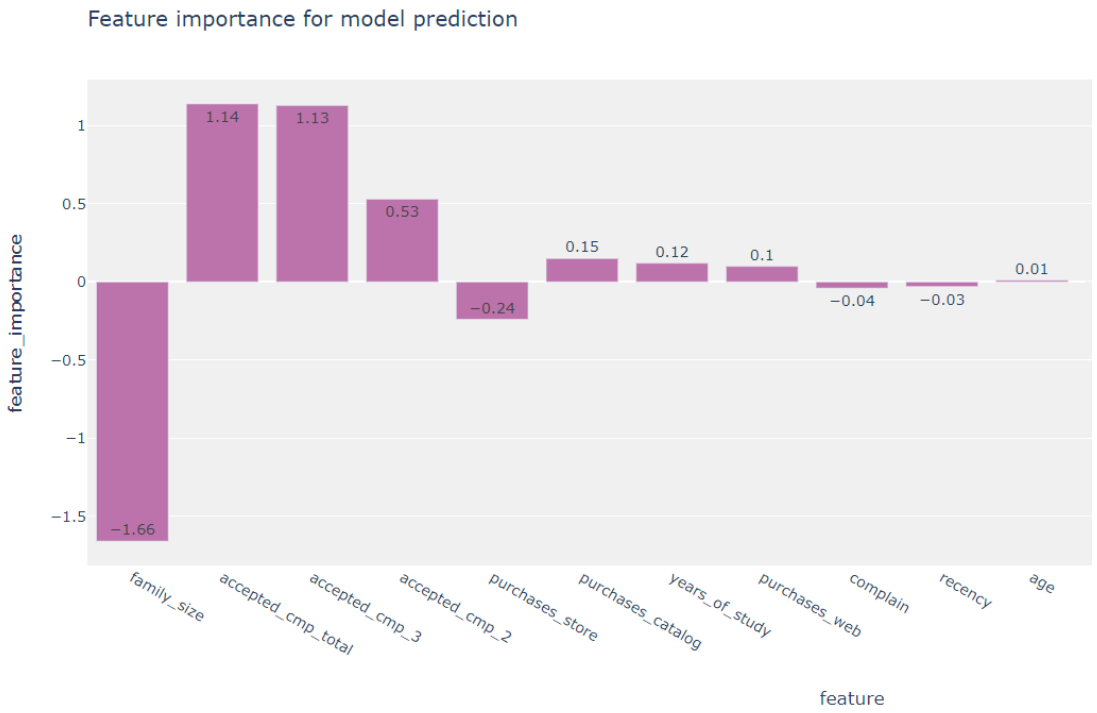


Figura 42: Gráfico que demonstra a importância das variáveis para o modelo.

3.7. Simulação do impacto financeiro

Por fim, no intuito de entender o impacto financeiro gerado pela utilização de segmentação de uma campanha de marketing utilizando técnicas de aprendizado de máquina, geramos uma simulação da receita gerada para o caso do não uso de técnicas de segmentação e com o seu uso.

Temos na base original que o custo para a veiculação da campanha por cliente é de $3 e a receita obtida caso o cliente aceite a proposta, é $11. Com isso, conseguimos descrever a função de lucro da campanha através da função descrita na figura 43.

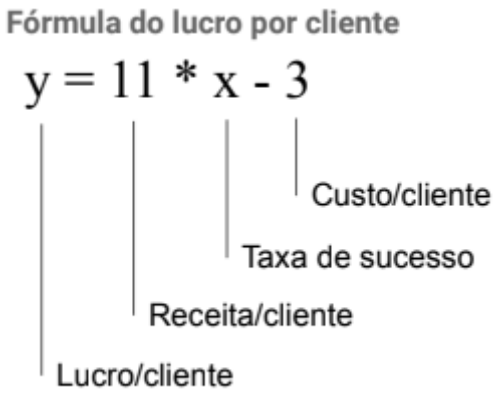


Figura 43: Função de lucro por cliente.

Com isso, é possível calcular a partir de qual taxa de sucesso a campanha gera lucro para a empresa. Ao zerar o valor de y, encontramos que a campanha atinge seu ponto de equilíbrio quando atinge uma taxa de sucesso de 27%, entretanto, sabemos que o cenário atual, onde a campanha foi veiculada sem nenhum tipo de segmentação, tivemos uma taxa de sucesso da campanha de 15%, o que resulta em um prejuízo de $1,35 por cliente como descrito na figura 44.

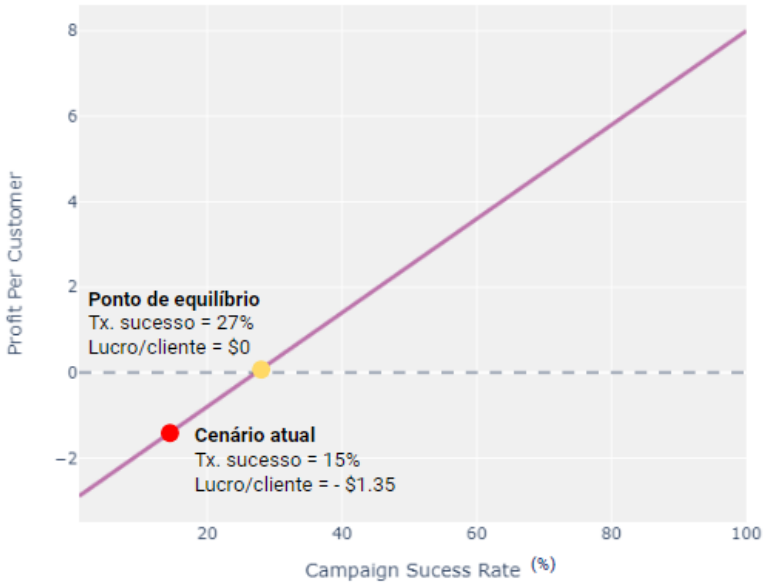


Figura 44: Gráfico de rentabilidade por cliente de acordo com a taxa de sucesso da campanha.

Recapitulando a matriz confusão apresentada anteriormente na figura 41, consideramos que iremos enviar a campanha somente para os clientes onde a previsão do modelo foi de que aceitariam a campanha, estando essa previsão errada, ou não, chegamos em uma taxa de sucesso de aproximadamente 58% como demonstrado na figura 45. De onde, 35% da base total, faz referência aos clientes do cluster Premium, e desses 14% da base total faz referência aos clientes que o modelo previu que aceitariam a oferta da campanha, onde desses, 58% realmente aceitaram a campanha.

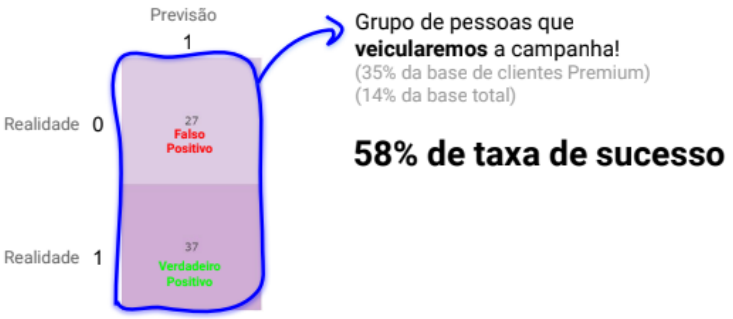


Figura 45: Plotagem da matriz confusão apenas para casos onde a previsão de aceitação da campanha era positiva.

Dessa forma, considerando uma base total de um milhão de clientes (número estimado pela empresa em seu case), é possível definir o impacto financeiro do uso de aprendizado de máquina para segmentação de uma campanha de marketing.

Primeiramente, iremos mensurar o impacto financeiro da campanha caso não fosse usada a segmentação de clientes, ou seja, a campanha fosse enviada para toda a base. Como ilustrado na figura 46, em uma base de 1 milhão de clientes, todos eles seriam impactados pela campanha, logo, ao custo de $3 por clientes, o custo total da campanha seria de $3 milhões, sabendo que a taxa de conversão seria de aproximadamente 15%, aproximadamente 150 mil clientes aceitariam a oferta proposta pela campanha, e considerando a receita de $11 por aceite, a receita total seria de $1,65 milhões, de forma que a receita menos o custo, resultaria em um prejuízo de $1,35 milhões para a veiculação desta campanha sem nenhum tipo de segmentação.

Figura 46: Mensuração do impacto financeiro caso não seja utilizada segmentação de campanha.

Já para o cenário onde utilizamos a segmentação de clientes proposta por esse estudo, como ilustrado na figura 47, para essa mesma base de 1 milhão de clientes, a segmentação sugere que enviemos para apenas um determinado grupo, que representa 14% da base total de clientes, como indicado na figura 45, o que resulta em 140 mil clientes impactados pela campanha, ao custo de $3 por cliente impactado, temos o custo total da campanha em aproximadamente $420 mil,

sabendo que a taxa de conversão para esse grupo segmentado de clientes é de aproximadamente 58%, temos aproximadamente 81 mil clientes aceitando a oferta proposta pela campanha, e considerando a receita de 11$ por aceite, a receita total da campanha seria de $893 mil, assim, considerando a receita menos o custo, essa campanha resultaria em um lucro de aproximadamente $473 mil.

Figura 47: Mensuração do impacto financeiro caso seja utilizada a segmentação de campanha proposta.

4. CONCLUSÃO E TRABALHOS FUTUROS

Como podemos ver neste estudo, a utilização de técnicas de aprendizado de máquina para determinados segmentos deixou de ser apenas um experimento e passou a ser um diferencial competitivo, tornando processos mais assertivos resultando em maiores margens de lucro.

O objetivo deste trabalho foi utilizar técnicas de aprendizado de máquina para otimizar uma campanha de marketing e medir o impacto dessas técnicas utilizando uma base de dados com informações demográficas e firmográficas de clientes de uma empresa de entrega por aplicativo. Para isso, essa base de dados recebeu diversos tratamentos como remoção de dados nulos e fora do padrão, criação de novas variáveis e redimensionamento de escala, além disso, técnicas como clusterização e regressão logística foram aplicadas para determinar os segmentos de público que seriam impactados pela campanha. Como resultado, foi possível aumentar a taxa de aceitação da campanha de 15 para 58 por cento, e onde antes a campanha havia um prejuízo projetado de $1,35 milhões, passou a ter um lucro projetado de $473 mil, atingindo assim o objetivo deste estudo, que era otimizar a campanha de marketing.

Por fim, como trabalhos futuros, desejamos realizar o estudo da projeção de valores gastos por estes indivíduos durante sua jornada como clientes, de forma a entender o grau de fidelidade desses clientes que aceitaram essa campanha, além de testar outros modelos além da regressão logística a fim de entender se existe uma melhora nos resultados obtidos.

5. REFERÊNCIAS BIBLIOGRÁFICAS

ZEN, Leandro. **O uso do aprendizado de máquina (machine learning) e seus benefícios para as empresas.** Disponível em <https://www.linkedin.com/pulse/o-uso-do-aprendizado-de-m%C3%A1quina-machine-learning-e-seus-leandro-zen/?trk=articles_directory&originalSubdomain=pt>. Acessado em 01 de agosto de 2022. 2022.

FERREIRA, Kellison. **4 coisas que não podem faltar na sua campanha de marketing.** Disponível em <https://rockcontent.com/br/blog/campanha-de-marketing/>. Acessado em 01 de agosto de 2022. 2018.

IFOOD. **ifood-data-advanced-analytics-test.** Disponível em <https://github.com/ifood/ifood-data-advanced-analytics-test>. Acessado em 11 de junho de 2022. 2021.

GOOGLE. **Conheça o Colab.** Disponível em <https://colab.research.google.com/notebooks/intro.ipynb?hl=pt_BR>. Acessado em 11 de junho de 2022. 2022.

PANDAS. **User Guide.** Disponível em <https://pandas.pydata.org/docs/user_guide/index.html#user-guide>. Acessado em 11 de junho de 2022. 2022.

PLOTLY. **Plotly Open Source Graphing Library for Python.** Disponível em <https://plotly.com/python/>. Acessado em 11 de junho de 2022. 2022.

SCIKIT LEARN. **Getting Started.** Disponível em <https://scikit-learn.org/stable/getting_started.html>. Acessado em 11 de junho de 2022. 2022.

CONNECTCOM. **DESIGN THINKING E LEAN STARTUP: FRAMEWORKS COM FOCO NA RESOLUÇÃO DE PROBLEMAS.** Disponível em <https://www.connectcom.com.br/design-thinking-e-lean-startup-frameworks-em-seu-problema/>. Acessado em 02 de agosto de 2022. 2019.

PEDOTE, Lucas. **Os 4 Ps de Data: o recheio é contexto e relevância.** Disponível em <https://medium.com/ifood-tech/o-recheio-%C3%A9-contexto-e-relev%C3%A2ncia-4

88e58b30e2d>. Acessado em 02 de agosto de 2022. 2020.

TRESMEIOS. **Tipos de segmentação: seja assertivo em suas campanhas.** Disponível em <https://www.tresmeios.com.br/blog/tipos-de-segmentacao/>. Acessado em 02 de agosto de 2022. 2021.

TRESMEIOS. **Tipos de segmentação: seja assertivo em suas campanhas.** Disponível em <https://www.tresmeios.com.br/blog/tipos-de-segmentacao/>. Acessado em 02 de agosto de 2022. 2021.

WALIA, Mrinal. **How To Solve Customer Segmentation Problem With Machine Learning.** Disponível em <https://www.analyticsvidhya.com/blog/2021/06/how-to-solve-customer-segmentation-problem-with-machine-learning/>. Acessado em 02 de agosto de 2022. 2021.

DINO. **Outlier detection using IQR method and Box plot in Python.** Disponível em <https://towardsdev.com/outlier-detection-using-iqr-method-and-box-plot-in-python-82e1e15232bd>. Acessado em 02 de agosto de 2022. 2022.

WIKIPEDIA. **Amplitude interquartil.** Disponível em <https://pt.wikipedia.org/wiki/Amplitude_interquartil>. Acessado em 02 de agosto de 2022. 2022.

YUKIO. **Localizando Outliers Através do Intervalo Interquartil.** Disponível em <https://estatsite.com.br/2018/12/01/localizando-outliers-atraves-do-intervalo-interquartil-boxplot-codigo-sas/>. Acessado em 02 de agosto de 2022. 2018.

PATEL, Harshil. **What is Feature Engineering — Importance, Tools and Techniques for Machine Learning.** Disponível em <https://towardsdatascience.com/what-is-feature-engineering-importance-tools-and-techniques-for-machine-learning-2080b0269f10#:~:text=Feature%20engineering%20is%20a%20machine,while%20also%20enhancing%20model%20accuracy.>. Acessado em 02 de agosto de 2022. 2021.

WIKIPEDIA. **Exploratory data analysis.** Disponível em <https://en.wikipedia.org/wiki/Exploratory_data_analysis.>. Acessado em 02 de agosto de 2022. 2022.

KUMAR, Nishant. **Feature Scaling :- Normalization, Standardization and Scaling**

!. Disponível em <https://medium.com/analytics-vidhya/feature-scaling-normalization-standardization-and-scaling-c920ed3637e7>. Acessado em 03 de agosto de 2022. 2020.

GUIMARÃES, Amanda. **Estatística: Teste Exato de Fisher e Teste de Qui-Quadrado usando R.** Disponível em <https://medium.com/omixdata/estat%C3%ADstica-teste-exato-de-fisher-e-teste-de-qui-quadrado-usando-r-4ee496da37fc#:~:text=O%20teste%20de%20Qui%2DQuadrado%20%C3%A9%20considerado%20um%20teste%20n%C3%A3o,esperadas%20para%20um%20certo%20evento.>. Acessado em 10 de agosto de 2022. 2019.

ANASTACIO, Bruno. **K-means: o que é, como funciona, aplicações e exemplo em Python.** Disponível em <https://medium.com/programadores-ajudando-programadores/k-means-o-que-%C3%A9-como-funciona-aplica%C3%A7%C3%B5es-e-exemplo-em-python-6021df6e2572>. Acessado em 03 de agosto de 2022. 2020.

BOOKDOWN. **K-means clustering.** Disponível em <https://bookdown.org/tpinto_home/Unsupervised-learning/k-means-clustering.html>. Acessado em 03 de agosto de 2022. 2021.

BROWNLEE, Jason. **How to Choose a Feature Selection Method For Machine Learning.** Disponível em <https://machinelearningmastery.com/feature-selection-with-real-and-categorical-data/#:~:text=Feature%20selection%20is%20the%20process,the%20performance%20of%20the%20model.>. Acessado em 03 de agosto de 2022. 2019.

LUNA, Zipporah. **Feature Selection in Machine Learning: Correlation Matrix | Univariate Testing | RFECV.** Disponível em <https://medium.com/geekculture/feature-selection-in-machine-learning-correlation-matrix-univariate-testing-rfecv-1186168fac12>. Acessado em 03 de agosto de 2022. 2021.

GILLIS, Alexander. **Data Splitting.** Disponível em <https://www.techtarget.com/searchenterpriseai/definition/data-splitting#:~:text=In%20machine%20learning%2C%20data%20splitting,into%20three%20or%20four%20sets>. Acessado em 03 de agosto de 2022. 2022.

MAYO, Matthew. **Dataset Splitting Best Practices in Python.** Disponível em

<https://www.kdnuggets.com/2020/05/dataset-splitting-best-practices-python.html>. Acessado em 03 de agosto de 2022. 2020.

KUO, Chris. **Using Over-Sampling Techniques for Extremely Imbalanced Data.** Disponível em <https://medium.com/dataman-in-ai/sampling-techniques-for-extremely-imbalanced-data-part-ii-over-sampling-d61b43bc4879>. Acessado em 04 de agosto de 2022. 2018.

CABRAL, Cleidy. **Aplicação do Modelo de Regressão Logística num Estudo de Mercado.** Disponível em <https://repositorio.ul.pt/bitstream/10451/10671/1/ulfc106455_tm_Cleidy_Cabral.pdf>. Acessado em 24 de agosto de 2022. 2022.

SMOLSKI, Felipe. **Capítulo 7 Regressão Logística.** Disponível em <https://smolski.github.io/livroavancado/reglog.html>. Acessado em 04 de agosto de 2022. 2022.

PÁDUA, Mateus. **Machine Learning -Métricas de avaliação: Acurácia, Precisão e Recall, F1-score.** Disponível em <https://medium.com/@mateuspdua/machine-learning-m%C3%A9tricas-de-avalia%C3%A7%C3%A3o-acur%C3%A1cia-precis%C3%A3o-e-recall-d44c72307959>. Acessado em 04 de agosto de 2022. 2020.

RABELO, Eduardo. **Cross Validation: Avaliando seu modelo de Machine Learning.** Disponível em <https://medium.com/@edubrazrabello/cross-validation-avaliando-seu-modelo-de-machine-learning-1fb70df15b78>. Acessado em 04 de agosto de 2022. 2019.

SOUZA, Emanuel. **Entendendo o que é Matriz de Confusão com Python.** Disponível em <https://medium.com/data-hackers/entendendo-o-que-%C3%A9-matriz-de-confus%C3%A3o-com-python-114e683ec509>. Acessado em 04 de agosto de 2022. 2019.

MOHAJON, Joydwip. **Confusion Matrix for Your Multi-Class Machine Learning Model.** Disponível em <https://towardsdatascience.com/confusion-matrix-for-your-multi-class-machine-learning-model-ff9aa3bf7826>. Acessado em 04 de agosto de 2022. 2020.

SERENGIL, Sefik. **Feature Importance in Logistic Regression for Machine**

Learning Interpretability. Disponível em <https://sefiks.com/2021/01/06/feature-importance-in-logistic-regression/#:~:text=Feature%20importance%20is%20a%20common,regression%20and%20decision%20trees%20before.>. Acessado em 04 de agosto de 2022. 2021.

SERENGIL, Sefik. **Feature Importance in Logistic Regression for Machine Learning Interpretability.** Disponível em <https://sefiks.com/2021/01/06/feature-importance-in-logistic-regression/#:~:text=Feature%20importance%20is%20a%20common,regression%20and%20decision%20trees%20before.>. Acessado em 04 de agosto de 2022. 2021.