Politécnico
Castelo Branco
Escola Superior
de Tecnologia

Previsão de Abandono para Serviços de Telecomunicações

Projeto I

Francisco Mateus Gonçalves

Nº20221849

Orientadores

Professora Doutora Ana Paula Neves Ferreira da Silva

Professor Doutor Arlindo Ferreira da Silva

Relatório de Projeto I, apresentado à Escola Superior de Tecnologia do Instituto Politécnico de Castelo Branco para cumprimento dos requisitos necessários à obtenção do grau de Licenciado em Engenharia Informática, realizada sob a orientação científica da Professora Adjunta Doutora Ana Paula Neves Ferreira da Silva e do Professor Adjunto Doutor Arlindo Ferreira da Silva, do Instituto Politécnico de Castelo Branco.

Fevereiro 2025

Composição do júri

Presidente do júri

Doutor, Osvaldo Arede dos Santos

Professor Adjunto da Escola Superior de Tecnologia de Castelo Branco

Orientadora

Doutora, Ana Paula Neves Ferreira da Silva

Professora Adjunta da Escola Superior de Tecnologia de Castelo Branco

Arguente

Doutor, Fernando Reinaldo da Silva Garcia Ribeiro

Professor Adjunto da Escola Superior de Tecnologia de Castelo Branco

Resumo

Este projeto foi desenvolvido com o intuito de criar um modelo de previsão de abandono de um serviço (*customer churn*) de telecomunicações utilizando algoritmos de *Machine Learning* (ML).

De modo a ser possível desenvolver um modelo de previsão de *customer churn,* foi necessário estudar as áreas de Inteligência Artificial, dedicadas a este tipo de previsão. Já existentes no setor da telecomunicação.

Elaborou-se um estudo do estado da arte do uso de técnicas de *Machine Learning* na previsão do *customer churn* no setor da telecomunicação. Este estudo evidenciou o crescente interesse atual neste tema e permitiu observar a capacidade dos modelos para fazerem previsões corretas.

Fez-se um estudo dos *datasets* utilizados para avaliar as distribuições, a sua representatividade e consistência.

Foram treinados 6 modelos de previsão e comparados entre si, fazendo ainda uma análise da importância de cada atributo dos modelos resultantes.

Adicionalmente para validar a generalização dos modelos, foram feitos alguns testes de previsão usando ao mesmo tempo dois *datasets* de contextos diferentes.

Desta maneira, a realização deste projeto permitiu identificar os melhores modelos de IA, com ênfase na capacidade de evitar a previsão de falsos casos de retenção.

Palavras-chave

Customer churn, *dataset*, *machine learning*, previsão, telecomunicações.

Abstract

This project was developed with the aim of creating a model to predict the abandonment of a telecommunications service (customer churn) using Machine Learning algorithms.

To be able to develop a predictive model for customer churn, it was necessary to study the areas of Artificial Intelligence, dedicated to this type of prediction that already exists in the telecommunication sector. To understand the work already done dedicated to solving this problem, a state-of-the-art study was carried out on the use of Machine Learning (ML) techniques in predicting customer churn in the telecom sector. This study highlighted the current growing interest in this topic and has shown the capacity of intelligent models to make correct predictions.

A study was carried out on the selected datasets, to evaluate the distribution, their representativeness and consistency.

Six prediction models were trained and compared with each other, and an analysis was also made of the importance of each attribute of the resulting models.

In addition, to validate the generalization of the models, some prediction tests were carried out using two datasets from different contexts at the same time.

In this way, the completion of this project made it possible to identify the best AI models, with an emphasis on the ability to avoid predicting false retention cases.

Keywords

Customer churn, dataset, forecasting, machine learning, telecom.

Índice geral

Índice de figuras

Índice de fórmulas

Lista de tabelas

Lista de abreviaturas, siglas e acrónimos

AUC (*Area Under the Curve*)

CatBoost (*Categorical Boosting*)

CNN (*Convolutional Neural Network*)

DL (*Deep Learning*)

DT (*Decision Tree*)

FPR (*False Positive Rate*)

GB (*Gradient Boosting*)

IA (*Inteligência Artificial*)

KNN (*K-Nearest Neighbors*)

LGBM (*Light Gradient-Boosting Machine*)

LIME (*Local Interpretation Model-Agnostic Explanations*)

LR (*Logistic Regression*)

LSTM (*Long Short-Term Memory*)

MLP (*Multilayer Perceptron*)

ML (*Machine Learning*)

NB (*Naïve Bayes*)

ROC (*Receiver Operating Characterisitic*)

RF (*Random Forest*)

SHAP (*SHapley Additive exPlanations*)

SMOTE (*Synthetic Minority Oversampling Technique*)

SMOTE-ENN (*Synthetic Minority Oversampling Technique and Edited Nearest Neighbors*)

SGB (*Stochastic Gradient Boosting*)

SVM (*Support Vector Machine*)

TPR (*True Positive Rate*)

XGBoost (*Extreme Gradient Boosting*)

1. Introdução

Durante todas as fases de comércio, as empresas têm uma alta dependência da permanência dos seus clientes no negócio [1]. O investimento recorrente dos clientes em serviços ou produtos de um negócio, contribui para o desenvolvimento e sucesso do mesmo [1]. Esta interpretação aplica-se também no setor das telecomunicações, onde é crucial para as operadoras, manter uma boa reputação e satisfazer os clientes de modo a retê-los nos serviços da empresa [1]. Estima-se que o custo associado ao ganho de novos clientes seja 10 vezes superior ao custo da retenção dos existentes [2]. Num ambiente de elevada concorrência, os esforços para a retenção de clientes tornam-se ainda mais relevantes de modo a garantir e maximizar lucros [3].

As principais razões conhecidas para o abandono, são a baixa qualidade dos serviços fornecidos, alto uso dos serviços e baixo custo na troca de fornecedor [1]. Identificando as razões que levam ao *customer churn* e definindo o perfil de um cliente na iminência de abandonar o serviço, as empresas de tecnologia conseguem implementar medidas que contrariem a tendência do *churn*.

Para a resolução deste problema, nos dias de hoje, as empresas de telecomunicações investem em modelos de previsão de *customer churn* eficientes e precisos. Um bom modelo de previsão de abandono de serviços é valioso, uma vez que possibilita uma prevenção contra o abandono do mesmo, esta prevenção no caso das empresas de telecomunicação, pode ser feita simplesmente através de uma proposta que aumente a satisfação do cliente com o serviço contratado. Desta maneira os modelos de previsão de abandono de serviços conseguem ter um impacto direto na maneira como uma empresa lida com os clientes que abandonam um serviço.

1.1. Objetivos

No âmbito da Unidade Curricular de Projeto I, pretende-se verificar o sucesso do uso de modelos de *Machine Learning* (ML), no problema de previsão de *customer churn* no setor das telecomunicações.

Desta maneira, foram definidos objetivos com o intuito de determinar a viabilidade da aplicação de ML para a previsão do *customer churn* neste setor. Assim, os objetivos definidos foram;

- Fazer uma revisão sistemática sobre o uso de modelos de ML na previsão do *customer churn*;
- Identificar *datasets* (conjunto de dados) de acesso livre disponíveis na internet no contexto deste problema;
- Identificar contribuições prévias de software relacionadas com a aplicação de modelos de inteligência artificial (IA) neste contexto;
- Com base no conhecimento adquirido, efetuar um estudo comparativo entre diferentes modelos de ML para a modelação dos dados disponíveis no contexto do problema;

1.2. Planeamento do Projeto

O trabalho proposto foi dividido em sete tarefas que foram realizadas ao longo do 1º semestre do ano letivo 2024/2025. Na **tabela 1** é apresentado o cronograma temporal das tarefas realizadas ao longo do desenvolvimento do trabalho, com uma divisão quinzenal.

A primeira tarefa, diz respeito à contextualização das áreas da inteligência artificial, de modo a ter um entendimento dos métodos a serem utilizados, modelos existentes, métricas de comparação entre modelos, entre outros aspetos.

A segunda tarefa, é relativa à procura dos *datasets* adequados entre os artigos lidos durante o estado da arte e plataformas públicas, de modo a serem estudados no contexto enunciado.

Na terceira tarefa, o objetivo principal é a identificação e familiarização de bibliotecas, para a linguagem Python, relacionadas com a aplicação de técnicas de ML, que contribuam no desenvolvimento prático do projeto.

O estado da arte, é a tarefa relativa à leitura e estudo de artigos científicos, que tenham feito algum tipo de contribuição recente na previsão de *customer churn* com o uso de técnicas de ML.

A quinta tarefa, refere-se à redação deste documento, sendo esta tarefa desenvolvida em paralelo com as restantes.

A análise dos *datasets* selecionados, é a tarefa cujo objetivo principal é a compreensão do contexto, informações e padrões que possam ser relevantes para o estudo proposto.

Por último, a tarefa de treino e avaliação dos modelos ML, diz respeito a todas as adaptações e alterações feitas aos *datasets* adotados, ao treino e avaliação dos modelos.

Tabela 1 — Cronograma do Projeto

Tarefa	2024						2025	
	Outubro		Novembro		Dezembro		Janeiro	
	1ª Quinzena	2ª Quinzena	1ª Quinzena	2ª Quinzena	1ª Quinzena	2ª Quinzena	1ª Quinzena	2ª Quinzena
Contextualização das áreas relevantes								
Procura de *datasets*								
Identificar bibliotecas de desenvolvimento								
Estado da arte								
Redigir o relatório								
Exploração dos *datasets* selecionados								
Treino e Avaliação dos Modelos de ML								

1.3. Estrutura do Relatório

Este relatório é constituído por oito capítulos, os quais podem ser descritos pelo seguinte:

- **Capítulo 1** – Contextualização e descrição do problema, proposta dos objetivos e planeamento associado.
- **Capítulo 2** – Exploração da área de inteligência artificial e dos tipos de aprendizagem máquina, dando ênfase às técnicas que se pretendem usar no trabalho.
- **Capítulo 3** – Revisão do estado da arte, comparação dos modelos presentes nos artigos e conclusões retiradas da análise.
- **Capítulo 4** – Apresentação das tecnologias utilizadas neste trabalho e explicação do seu propósito.
- **Capítulo 5** – Identificação e análise dos *datasets* selecionados.
- **Capítulo 6** – Apresentação do processo completo de treino e avaliação dos modelos de ML.
- **Capítulo 7** – Validação independente dos modelos ML utilizados.
- **Capítulo 8** – Conclusões obtidas durante o desenvolvimento deste trabalho e menção de desafios e trabalho futuro.

2. Inteligência Artificial

A inteligência artificial pode ser definida como “a tecnologia que permite que computadores e máquinas simulem a aprendizagem humana, a compreensão, a resolução de problemas, a tomada de decisões, a criatividade e a autonomia.” [4].

No ano de 1950, Alan Turing, considerado um dos pais da ciência da computação, publicou o artigo “*Computing Machinery and Intelligence*” [5]. Nesse artigo, Alan Turing apresentou pela primeira vez a questão “Será que as máquinas conseguem pensar?”. De modo a explorar essa questão, Turing propôs o “Jogo da Imitação” [5]. O conceito do “Jogo da Imitação” envolve três participantes: um humano que desempenha o papel de interrogador, um segundo humano e uma máquina. O interrogador isolado, troca mensagens de texto com os outros participantes, sem saber a identificação correspondente. O objetivo do interrogador é identificar qual dos participantes é humano. No caso da máquina ser capaz de enganar o interrogador e fazer-se passar por um humano numa quantidade significativa de testes, então Alan Turing sugeriu que a máquina poderia ser considerada “inteligente”.

Em 1956, o termo “inteligência artificial” é cientificamente adotado durante a conferencia de Dartmouth, considerada como o nascimento do novo ramo de estudo [6].

A inteligência artificial engloba um conjunto vasto de outras subáreas. Geralmente é feita uma separação em apenas 3 camadas, a camada da Inteligência Artificial, sendo esta a mais genérica, a camada de ML, que se concentra em algoritmos que permitem às máquinas aprenderem a partir de dados, e por último a camada mais especifica, o *deep learning* (DL) que explora redes neuronais profundas, para analisar grandes volumes de dados de maneira mais detalhada [7].

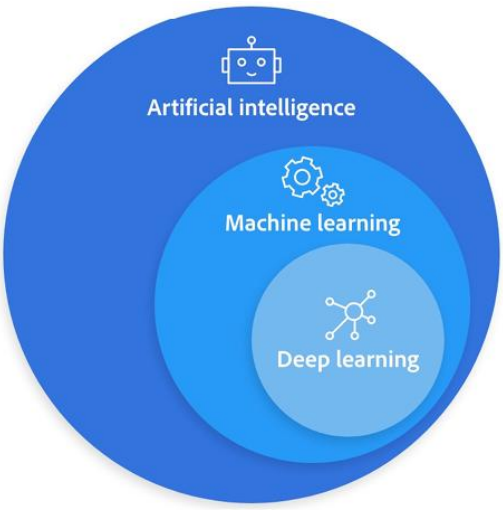

Figura 1 — As Principais Áreas da IA (fonte: [7])

2.1 *Machine Learning*

Machine Learning pode ser entendido como o processo de converter experiências em conhecimento. Em ML o algoritmo de aprendizagem recebe como entrada os dados de treino, representantes das experiências, e estas são transformadas numa saída, que representa o conhecimento prático adquirido [8].

O ciclo de vida de um sistema de ML é um processo iterativo que envolve diversas etapas essenciais, desde a aquisição de dados até à utilização do modelo. A figura abaixo ilustra este ciclo, destacando as quatro etapas principais: treinar o modelo, avaliar os resultados, utilizar o modelo e atualizar os dados.

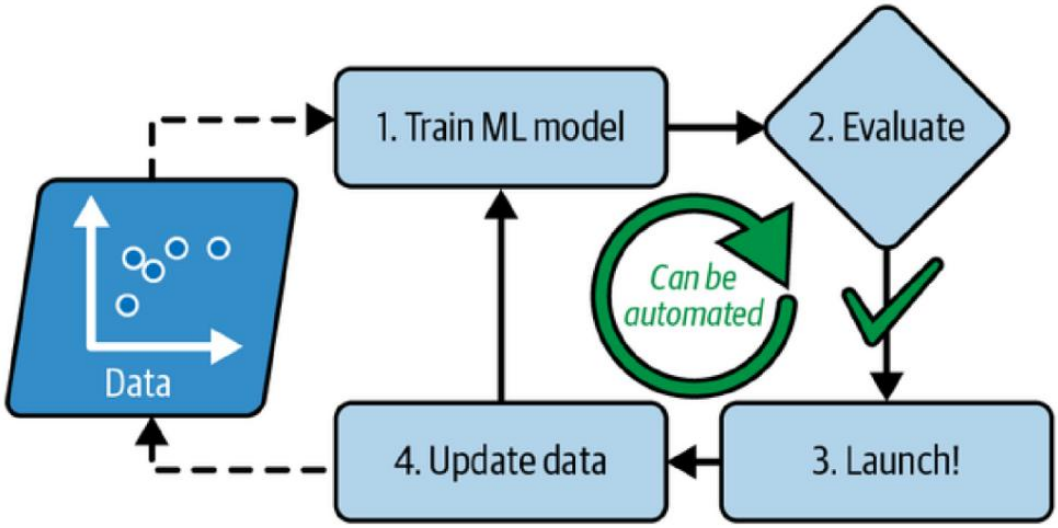

Figura 2 — Abordagem de um Problema com uso de ML (fonte: [9])

Para treinar um modelo de ML, é necessário escolher um mecanismo de aprendizagem adequado, que irá depender dos dados disponíveis e do contexto do problema a ser resolvido. Os principais tipos de aprendizagem máquina são: aprendizagem supervisionada que utiliza dados com um rótulo atribuído, aprendizagem não supervisionada que identifica padrões em dados sem rótulos e aprendizagem por reforço, baseada em recompensas obtidas através de interações com o ambiente [9]. No contexto deste projeto, a aprendizagem supervisionada é a abordagem utilizada.

2.1.1 Aprendizagem Supervisionada

A aprendizagem supervisionada pressupõe que, para o problema em questão, existem dados corretamente rotulados. Esta rotulagem diz respeito à identificação das categorias do atributo alvo de um *dataset*, também conhecida como *label* [9].

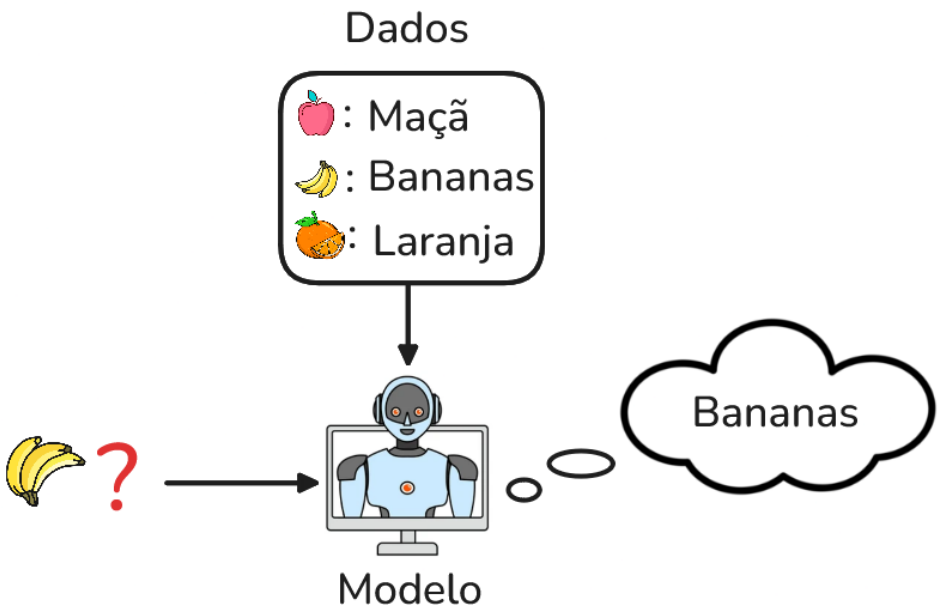

Figura 3 — Ilustração do Conceito Aprendizagem Supervisionada (adaptado de: [10])

A aplicação da aprendizagem supervisionada é feita em dois tipos principais de problemas: classificação e regressão.

Nos problemas de classificação, as categorias de saída possíveis são previamente conhecidas. Quando o modelo recebe uma nova entrada, geralmente dados que não tenham sido usados no conjunto de treino, o modelo deve ser capaz de classificar corretamente a categoria de saída correspondente [9].

Além disso, a aprendizagem supervisionada também pode ser usada para prever valores numéricos. Ao contrário dos problemas de classificação, os problemas de regressão estão associados a uma saída numérica, dentro de um intervalo possível ou esperado. Para isso, são utilizadas modelos de regressão numérica que permitem realizar as previsões [9].

2.2 Métricas de Avaliação para Problemas de Classificação

Os modelos de ML desenvolvidos devem ser avaliados, de modo a verificar se estão aptos a serem utilizados num contexto real. Esta avaliação deve ser feita com dados desconhecidos para o modelo, no caso, dados que não foram utilizados para treinar o modelo.

Uma forma comum de estimar a performance de um modelo é utilizando a validação cruzada. Esta consiste na divisão do *dataset* original em k partes, subconjuntos do *dataset* original chamados *folds*. Cada parte da divisão feita, é usada na sua vez como conjunto de teste, enquanto as restantes são usadas como conjunto de treino. Desta maneira todos os registos do *dataset* contribuem para a avaliação do modelo [11].

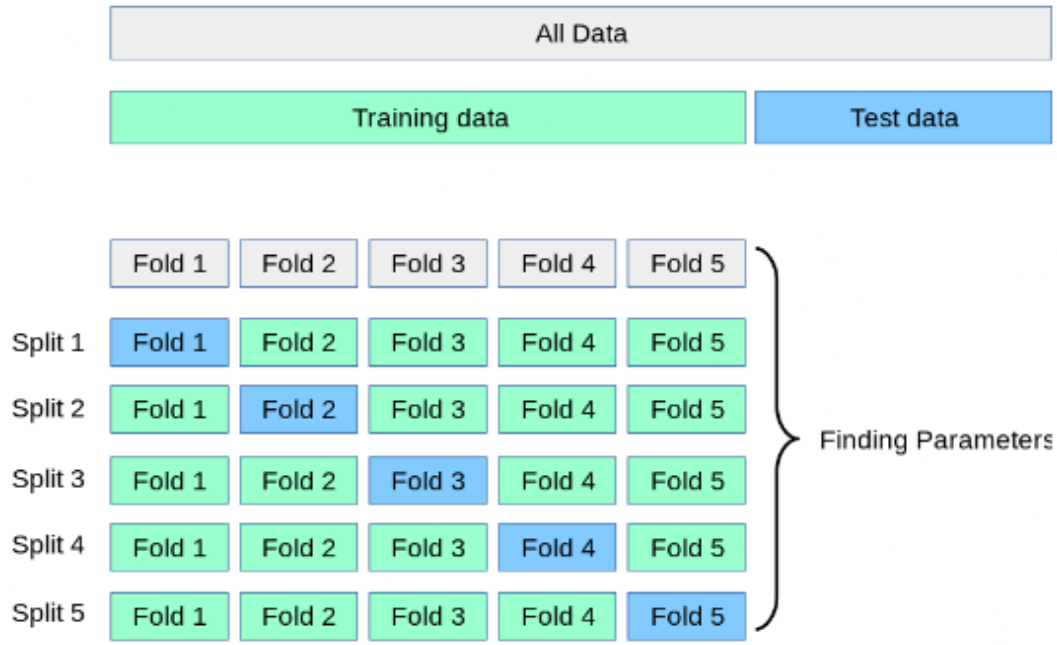

Figura 4 — Ilustração da Validação Cruzada k=5 (Adaptado de: [12])

Dois dos principais objetivos de avaliação de modelos são evitar o *overfitting* e o *underfitting*. O *overfitting* acontece quando o modelo aprende excessivamente os padrões específicos dos dados de treino, apresentando um mau desempenho em dados novos, devido à falta de generalização da sua aprendizagem. O *underfitting* acontece quando o modelo não consegue captar os padrões existentes nos dados, resultando num desempenho negativo [9].

O ideal é treinar um modelo capaz de generalizar a aprendizagem e que garanta um bom desempenho em dados novos, não usados no processo de treino.

Antes de conhecer as métricas, é importante compreender o significado e propósito da matriz de confusão, uma ferramenta fundamental para analisar o desempenho de modelos de classificação.

Num problema de classificação existem 4 tipos de resultados:

- Verdadeiros Positivos (TP): Casos em que o modelo prevê a classe positiva corretamente.
- Falsos Positivos (FP): Casos em que o modelo prevê a classe positiva incorretamente.
- Verdadeiros Negativos (TN): Casos em que o modelo prevê corretamente a classe negativa.
- Falsos Negativos (FN): Casos em que o modelo prevê incorretamente a classe negativa.

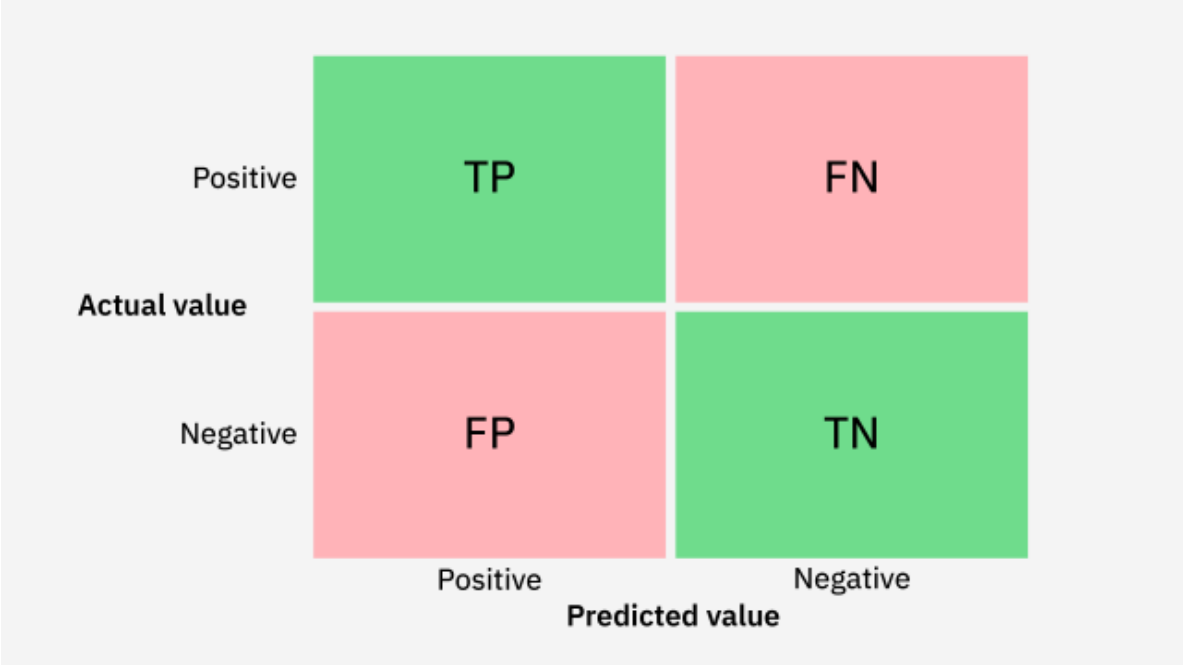

Figura 5 — Matriz de Confusão (fonte: [13])

Com base numa matriz de confusão, é possível calcular métricas relevantes para a avaliação de um modelo de ML. Existem diversas métricas de avaliação para diferentes tipos de problemas. As métricas relevantes para classificação são:

- *Accuracy*: Mede a proporção de previsões corretas em relação ao total de instâncias [14].

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Formula 1 — Cálculo da Accuracy

- *Precision*: Mede a proporção de verdadeiros positivos entre todas as instâncias classificadas como positivas [14].

$$Precision = \frac{TP}{TP + FP}$$

Fórmula 2 — Cálculo da *Precision*

- *Recall (sensivity/TPR)*: Proporção de verdadeiros positivos entre todas as instâncias positivas. É útil quando os falsos negativos são críticos [14].

$$Recall = \frac{TP}{TP + FN}$$

Fórmula 3 — Cálculo do *Recall*

- *FPR (False Positive Rate)*: Mede a proporção de falsos negativos entre todas as instâncias negativas [14].

$$FPR = \frac{FP}{FP + TN}$$

Fórmula 4 — Cálculo do *FPR*

- *F1-Score*: Representa a média harmônica entre a *precision* e a *recall* [14].

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

Fórmula 5 — Cálculo da *F1-Score*

- *ROC Curve (Receiver Operating Characteristic Curve)*: É uma representação gráfica que mostra a *performance* de um modelo de classificação em diferentes pontos de corte para a probabilidade de previsão. Sendo este construído através de duas métricas, o *TPR* e o *FPR* [14].

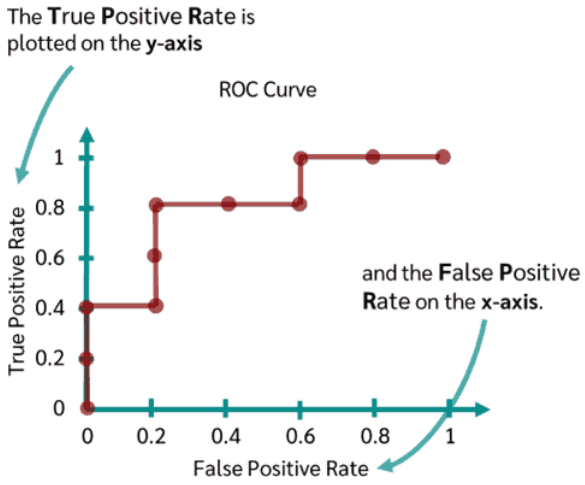

Figura 6 — Exemplo de Curva *ROC* (fonte: [15])

- *AUC (Area Under the Curve)*: É o resultado da área sob a curva *ROC*. É uma boa estimativa do desempenho preditivo do classificador, sendo esta invariante à escala [14]. Além disso, esta é uma métrica independente do limiar de classificação, avaliando a capacidade do modelo de distinguir entre classes positivas e negativas em diferentes configurações. Quanto maior o valor do *AUC*, melhor será o modelo na tarefa de separação de classes.

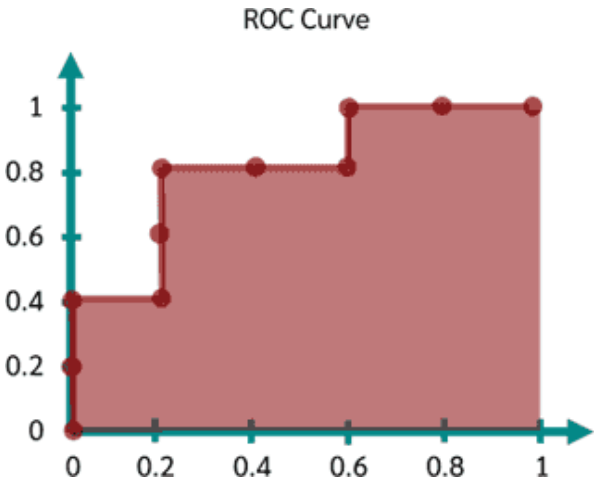

Figura 7 — Representação da *AUC* (fonte: [15])

No contexto do problema abordado neste trabalho as métricas mais relevantes são a *accuracy*, a *precision*, a *recall* e a *AUC*. Deste modo, elas serão utilizadas para avaliar a *performance* dos modelos utilizados, garantindo uma análise abrangente da *performance* dos mesmos. A *AUC* é utilizada para medir a capacidade de discriminação entre as classes, enquanto a *accuracy* avalia o desempenho preditivo geral do modelo, já a *precision* foca-se na eficácia em prever corretamente os casos positivos, e a *recall* destaca a capacidade do modelo em minimizar previsões de falsos negativos, que são um aspeto critico no contexto do problema abordado.

3. Estudo do Estado da Arte

Na realização da revisão de literatura, foi selecionado um conjunto de palavras-chave para identificar os artigos mais relevantes. A pesquisa foi focada na aplicação de técnicas de previsão de *customer churn*. A busca foi realizada dia 16 de novembro de 2024 usando a *string* de pesquisa avançada “client AND retention AND machine AND learning”, tendo sido obtidos 76 resultados. Adicionalmente foram removidos artigos cujo tema fosse dirigido para áreas distantes das telecomunicações através da leitura do título e do *abstract.* Dos 76 artigos obtidos foram eliminados 31, sobrando 45 para uma análise mais atenta. Após esta primeira leitura, selecionaram-se os 10 artigos com maior relevância para apresentar neste capítulo.

Os problemas de *customer churn* têm uma relevância crucial, especialmente no setor das telecomunicações. A quantidade mínima de *customer churn* pode afetar gravemente uma empresa [16], por esse motivo esta tem sido uma área de estudo nos últimos anos. Os métodos de previsão de *customer churn* têm evoluído significativamente ao longo do tempo, devido ao investimento tecnológico no ramo da inteligência artificial e à crescente disponibilidade de dados.

Na literatura o *Logistic Regression* (LR) e o *Decision Tree* (DT) são alguns dos métodos mais utilizados salientando a sua simplicidade, [16-20]. Com a crescente complexidade dos dados e a expansão do poder computacional, surgiram métodos mais sofisticados capazes de capturar padrões complexos. São o caso das *Support Vector Machine* (SVM), [17, 19, 21-24], e dos métodos de *ensemble* que se tornaram muito populares. Exemplos notáveis de técnicas de *ensemble* incluem *Random Forest* (RF), *Categorical Boosting* (CatBoost) e *Extreme Gradient Boosting* (XGBoost), [16-25], que combinam os resultados de diferentes modelos para melhorar a *performance* geral.

3.1 Revisão da Literatura

Nesta secção será apresentado um resumo dos 10 artigos selecionados salientando o objetivo de cada estudo e os resultados demonstrados:

Schaeffer e Sanchez propuseram um sistema de previsão da retenção de clientes, em serviços pré-pagos entre empresas, como é o caso do setor das telecomunicações, através de métricas como minutos ou megabytes pré-pagos em telemóveis [17]. Usando este sistema uma analogia contrária da típica previsão do *customer churn*, necessitando do ajuste de parâmetros como o

período de inatividade dos clientes que seria considerado como abandono. O sistema utiliza dados das transações mensais dos clientes, fazendo uma análise das compras e do consumo de serviços por parte do mesmo. O *dataset* utilizado contém 1230 registos de clientes retidos e possui 129 registos de clientes perdidos. Para efeitos comparativos, os autores testaram várias técnicas de *machine learning*, como as SVM e as *Random Forest*. De modo a comparar os modelos implementados foram usados neste trabalho métricas como *recall* (*sensivity)*, *specificity*, *accuracy* e *lift*. Entre os métodos testados, as *Random Forest* obtiveram o melhor resultado de *recall*, alcançando até 80.5%.

Nagaraj et al. propuseram um modelo de previsão de *customer churn* baseado no comportamento dos clientes e utilizando algoritmos de *machine learning*, dirigido ao *e-commerce*, com destaque nas empresas de telecomunicações [18]. Este estudo utilizou um *dataset* proveniente da plataforma Kaggle, contendo 20 atributos, a descrição e o tipo dos mesmos. De modo a remover o ruído nos dados do *dataset* foi feita uma seleção das variáveis relevantes. Entre os algoritmos utilizados tais como *Decision Tree*, *Random Forest*, SVM, os autores concluíram que as *Random Forest* obteveram uma boa precisão na previsão, embora esta não seja especificada. Concluindo que o uso de *machine learning* é uma abordagem eficaz para a previsão de *customer churn*.

Prabadevi et al. realizaram uma análise de *customer churn* através do uso de algoritmos de *machine learning* como *Random Forest*, *Stochastic Gradient Boosting* (SGB), *K-Nearest Neighbors* (KNN), e *Logistic Regression* [19]. Estes algoritmos foram treinados através de um *dataset* com 21 atributos obtido da plataforma Kaggle. Com este estudo foi concluído que o algoritmo SGB foi o mais adequado para a previsão de *customer churn*, obtendo o melhor desempenho com uma *precision* de 65%, seguido pelas *Random Forest* com uma *precision* de 64%, ambos os algoritmos possuindo uma *recall* de 46%.

Siddika et al. fizeram um estudo onde propuseram um sistema para prever o *customer churn* no setor das telecomunicações, através do uso de modelos de *machine learning* e *deep learning* [20]. Para treino e teste dos modelos, os autores utilizaram um *dataset* com 20 atributos descritivos do uso dos serviços de telecomunicação oferecidos a um número de 3333 clientes. Neste estudo foi realizada uma limpeza do ruido dos dados do *dataset* e uma identificação dos atributos mais relevantes para a previsão do *customer churn*. Os algoritmos testados para efeitos comparativos foram *Naïve Bayes* (NB), *Random Forest*, KNN, *Decision Tree, Logistic Regression, Multilayer Perceptron* (MLP), *Convolutional Neural Network* (CNN) e *Long Short-Term Memory* (LSTM), sendo que o algoritmo *Decision Tree* obteve o melhor resultado de *recall*, com um valor de 84%, seguido pelas *Random Forest* que obteve um resultado de 83%, mas destacando-se tendo em conta as métricas de *precision*, *accuracy* e *AUC* superiores.

Raj et al. implementaram um sistema para o setor da telecomunicação de previsão de *customer churn*, com o uso de técnicas de *resampling* e de algoritmos de classificação *ensemble* [21]. Neste trabalho foram usados algoritmos individuais *Decision Tree* e SVM, além dos algoritmos de classificação *ensemble* XGBoost, CatBoost e *Random Forest*, tendo sido estes treinados a partir de um *dataset* da IBM proveniente da plataforma Kaggle [22], com 21 atributos. Dentro dos dados existentes, os autores fizeram uma preparação dos mesmos, de modo a aprimorar a *performance* dos modelos a serem testados. Esta preparação consistiu no processamento de valores nulos e dos *outliers*, seleção de atributos, codificação dos dados categóricos e *resampling*, onde foram usados dois métodos *Synthetic Minority Oversampling Technique* (SMOTE) e *Synthetic Minority Oversampling Technique and Edited Nearest Neighbors* (SMOTE-ENN). De maneira a avaliar a *performance* dos modelos utilizados, os autores deste trabalho utilizaram as métricas *accuracy*, *recall*, *precision*, *F1-score* e *AUC*. A partir dos resultados deste estudo foi possível notar uma melhoria significativa dos resultados ao utilizar o método de *resampling* SMOTE-EEN, chegando a atingir o valor máximo *recall* de 92%, *AUC* de 99%, *accuracy* de 95% e uma *precision* de 96% com a utilização do algoritmo XGBoost. É importante referir que tanto o algoritmo SVM como as *Random Forest* obtiveram bons resultados de *performance* comparáveis com os resultados do algoritmo XGBoost.

Angelina et al. desenvolveram um trabalho onde é utilizado o algoritmo CatBoost para prever o *customer churn* no setor da telecomunicação [23]. Para treinar e testar o modelo proposto, foi feita uma divisão do *dataset* em que, 70% dos dados foram usados para treino e 30% para teste. Embora não esteja explicita a fonte, nem detalhes do *dataset* utilizado, é feita referência a atributos como total de minutos, total de minutos noturnos e estado do cliente. De modo a determinar a importância dos atributos do *dataset* os autores utilizaram o método *Permutation Feature Importance*, completando o processo de construção de árvores binárias simétricas pelo CatBoost. Por motivos comparativos, foram testados os algoritmos SVM, *Random Forest* e CatBoost. Os resultados mostram que o CatBoost obteve o melhor desempenho, sendo feita uma comparação simples através da métrica *accuracy,* alcançando este o melhor resultado de 95%.

Acero-Charaña et al. propuseram um modelo de *machine learning* para previsão do *customer churn*, utilizando um *dataset* de 21 atributos, contendo um conjunto de 3333 clientes de uma empresa do setor da telecomunicação [24]. Os modelos de ML utilizados neste trabalho foram Decision Tree, Artificial Neural Network e SVM, tendo estes sido implementados com a utilização da plataforma de analise de dados KNIME. As medidas utilizadas para comparar a *performance* dos algoritmos foram *precision*, *recall*, *accuracy*, *f1-score*, *recall* (*sensivity)* e *specificy*. Assim sendo, o algoritmo *Decision Tree* apresentou a maior *recall*, com um valor de 61.9%, a melhor *accuracy*, com um resultado de 91,7% e uma *precision* de 73,9%. O modelo SVM embora possua uma *accuracy* de 85.5%, obteve uma *recall* de 0%, o que significa que classificou todos os casos de teste

como clientes retidos, ou seja, apenas fez a previsão de casos negativos e falsos negativos.

Chang et al. fizeram um estudo onde foi proposto um sistema de previsão de *customer churn* no setor das telecomunicações [16]. Para a implementação deste sistema foram usados os modelos, *Logistic Regression*, KNN, NB, *Decision Tree* e *Random Forest*. Adicionalmente o *dataset* utilizado para o treino e teste dos modelos está disponível publicamente, sendo que possui 7043 registos com 38 atributos que descrevem os clientes subscritos a um serviço de telecomunicação. Entre os testes realizados, as *Random Forest* destacaram-se relativamente aos outros modelos testados, obtendo uma *recall* de 85.5%, uma *accuracy* de 86,9% e um valor de *AUC* de 95%. Em adição, este estudo explorou as ferramentas de visualização ilustrativas dos resultados dos modelos, *Local Interpretation Model-Agnostic Explanations* (LIME) e *SHapley Additive exPlanations* (SHAP). Através do uso destas ferramentas os autores conseguiram concluir que os atributos mais relevantes para o acontecimento do *churn*, segundo a ferramenta SHAP são "*Contrat*", "*Number of referrals*", "*Tenuer in months*", "*Monthly charge*" e "*Online security*", e segundo a ferramenta LIME são "*Contract*", "*Number of Referrals*", "*Monthly charge*", "*Online security*", "*Payment method*", "*Age*", "*City*".

Krishna et al. investigaram o uso de diferentes algoritmos de ML para a previsão de *customer churn* no setor de telecomunicações [25]. Neste trabalho são comparados os modelos *Logistic Regression*, *SVM*, *Random Forest*, *Gradient Boosting* (GB), XGBoost e *Light Gradient-Boosting Machine* (LGBM). Os modelos desenvolvidos são treinados com um *dataset* proveniente da plataforma Kaggle, com 2000 registos e 14 atributos, em que 12,69% dos registos abandonaram o serviço. Este trabalho prova a importância da métrica *precision*, embora a *recall* seja considerado uma métrica crítica neste contexto. Os resultados de *recall* foram muito positivos, atingindo valores entre os 82.4% e os 100%, destacando uma boa capacidade de identificar casos positivos verdadeiros. No entanto, os valores baixos de *precision*, cujo máximo foi 19.4%, indicam que muitos casos classificados como positivos seriam na verdade negativos, ou seja, falsos positivos. Sugerindo que os modelos se focaram em prever casos positivos, enfrentando dificuldades em identificar verdadeiros negativos.

Agasti e Satpathy implementaram um modelo de previsão de *customer churn* com o algoritmo *Naïve Bayes*, no contexto do setor de telecomunicação [26]. Para o desenvolvimento deste trabalho foi utilizado um *dataset* obtido no Kaggle de uma empresa de telecomunicações, embora este não tenha sido especificado, possui atributos como "*voice service*", "*SMS*", "*response time*", "*latency*", "*packet loss*", "*download/upload speed*" e "*line of sight*". Os dados deste *dataset* foram previamente limpos através da remoção de valores nulos, normalização e balanceamento através dos algoritmos *Random Undersampling, Random Over-Sampling* e SMOTE. De maneira a avaliar o desempenho do modelo NB, os resultados deste foram comparados com os resultados do modelo *Decision Tree* implementado com os mesmos dados. O algoritmo NB mostrou maior eficácia na

previsão do *customer churn* comparativamente ao algoritmo *Decision Tree*, obtendo uma *recall* de 97.4%, uma *accuracy* de 98,6% e uma *precision* de 97,8%.

3.2 Resumo Comparativo

Nesta secção é apresentado um resumo comparativo, no formato de uma tabela, dos melhores algoritmos para cada um dos artigos estudados, referindo quando presentes, os resultados das métricas mais relevantes.

Tabela 2 — Tabela de Análise Comparativa dos Artigos

Artigo	Melhores Algoritmos	*Recall*	*Precision*	*Accuracy*	*AUC*
1 - [17]	RF	Min - 72.1% Max – 80.5%	-	Min - 72.5% Max – 82.4%	-
2 - [18]	RF	-	-	-	-
3 - [19]	SGB	46%	65%	-	-
	RF	46%	64%	-	-
4 - [20]	DT	84%	83%	92%	84.2%
	RF	83%	94%	94.7%	91.3%
5 - [21]	SVM	Min - 82% Max – 92%	Min - 93% Max – 83%	Min - 79% Max – 93%	Min - 79% Max – 98%
	RF	Min - 83% Max – 92%	Min - 81% Max –96%	Min - 78% Max –95%	Min - 82% Max –98%
	XGBoost	Min - 84% Max – 92%	Min - 83% Max – 96%	Min - 78% Max – 95%	Min - 82% Max –99 %
6 - [23]	CatBoost	-	-	95%	-
7 - [24]	DT	61.9%	73.9%	91.7%	-
8 - [16]	RF	85.5%	-	86.9%	95%
9 - [25]					
10 - [26]	NB	97.4%	97.8%	98.6%	-

3.3 Considerações Finais

Através da análise de várias contribuições na área de previsão do *customer churn*, disponíveis na literatura, pode-se concluir que os algoritmos mais utilizados são as SVM, *Logistic Regression*, *Decision Tree*, *Naïve Bayes* e algoritmos de ensemble, como as *Random Forest* e o XGBoost.

Relativamente ao desempenho destes algoritmos, o algoritmo *Random Forest*, destacou-se, mostrando resultados bons de forma consistente, [16-17, 19, 21, 25], provando a sua capacidade de compreender relações complexas entre atributos.

No entanto, outras técnicas conseguiram demonstrar competência para fazer previsões confiáveis como o SVM e o XGBoost que em [21], obtiveram um resultado de *recall* máximo de 92%. Já o algoritmo *Naïve Bayes*, num cenário de preparação rigorosa do *dataset*, mostrou resultados competitivos no estudo [26], com uma *recall* de 97.4%.

É de notar que a qualidade dos *datasets* e preparação dos dados serão dois fatores importantes para um bom desempenho dos modelos.

Nos estudos [21] e [26], os resultados significativamente positivos, podem ser atribuídos a uma preparação adequada dos dados antes do treino dos modelos. Em [25], embora a *recall* seja alto, a *precision* foi reduzida, dando origem a uma quantidade de falsos positivos injustificável.

O uso de ferramentas como SHAP e LIME em [16], demonstra-se útil na explicabilidade dos mesmos, uma vez que identifica possíveis fatores a serem tomados em atenção, quando a criação de uma proposta para evitar casos de *customer chrun* é o objetivo principal.

É possível concluir que não existe um modelo ideal para a previsão do *customer churn*, no entanto as técnicas *Random Forest* e XGBoost demonstraram uma *performance* elevada de forma consistente. Os estudos referidos, reforçam também o benefício de *performance* ao usar técnicas de preparação de dados.

4. Tecnologias e Ferramentas Utilizadas

Para a realização deste trabalho foi necessário o uso de diversas tecnologias e ferramentas. Neste capítulo, serão detalhadas as ferramentas utilizadas, acompanhadas pela justificação da sua escolha e da maneira como cada uma contribuiu para o desenvolvimento dos objetivos propostos.

4.1 Linguagem de Programação Python

Python é uma linguagem de programação interpretada, de alto nível e orientada a objetos. Esta é uma linguagem de programação cuja curva de aprendizagem é suave, uma vez que possui uma sintaxe simples e de fácil compreensão [27].

Existe uma forte presença da comunidade de programadores, incluindo a área de *Machine Learning*, a trabalhar com Python. Uma das razoes para a sua popularidade, é a quantidade de bibliotecas e *Frameworks* desenvolvidas e disponíveis que contribuem para a produtividade e desenvolvimento eficiente [28].

Todo o código Python foi implementado em *NoteBooks* através do Jupyter NoteBook, pela capacidade de divisão dos documentos em células, tornando assim a implementação mais fácil de interpretar e estruturada [29].

Por conseguinte, na realização deste trabalho o Python foi a linguagem de programação escolhida, pela sua presença e progresso na área de ML, facilidade de compreensão e manipulação de dados. A versão utilizada do Python durante o desenvolvimento deste projeto foi a 3.12.7. Estando todo o código desenvolvido deste projeto disponível em https://github.com/FranGoncal/ProjetoLEI.

4.2 Bibliotecas de *Software*

Para a análise e manipulação de dados, o uso de bibliotecas de software é essencial para um desenvolvimento eficiente e otimizado. De seguida, são apresentadas as bibliotecas mais relevantes, utilizadas durante este projeto.

4.2.1 Pandas

Pandas é uma biblioteca *open-source* em Python que visa facilitar a manipulação de *datasets*. O recurso principal desta biblioteca é o objeto "*DataFrame*", este é uma estrutura de dados bidimensional que permite armazenar, visualizar e manipular dados, organizados em linhas e colunas [30]. Sendo esta biblioteca útil para o desenvolvimento deste projeto.

```python
import pandas as pd
# Ler o arquivo CSV
df = pd.read_csv('customer_churn_data.csv')
print(df)
```

```
     CustomerID  Age  Gender  Tenure  MonthlyCharges    ContractType  \
0             1   49    Male       4           88.35  Month-to-Month
1             2   43    Male       0           36.67  Month-to-Month
2             3   51  Female       2           63.79  Month-to-Month
3             4   60  Female       8          102.34        One-Year
4             5   42    Male      32           69.01  Month-to-Month
..          ...  ...     ...     ...             ...             ...
995         996   42    Male      41           37.14  Month-to-Month
996         997   62    Male       9           80.93  Month-to-Month
997         998   51  Female      15          111.72  Month-to-Month
998         999   39    Male      68           65.67        One-Year
999        1000   50    Male       1           56.67  Month-to-Month

    InternetService  TotalCharges TechSupport Churn
0       Fiber Optic        353.40         Yes   Yes
1       Fiber Optic          0.00         Yes   Yes
2       Fiber Optic        127.58          No   Yes
3               DSL        818.72         Yes   Yes
4               NaN       2208.32          No   Yes
..              ...           ...         ...   ...
995     Fiber Optic       1522.74         Yes   Yes
996             NaN        728.37          No   Yes
997     Fiber Optic       1675.80         Yes   Yes
998             NaN       4465.56          No   Yes
999             NaN         56.67          No   Yes

[1000 rows x 10 columns]
```

Figura 8 — Exemplo de Carregamento e Visualização de um *DataFrame* da Biblioteca Pandas

4.2.2 NumPy

NumPy é uma biblioteca considerada como fundamental para a computação científica em Python [31]. Oferece suporte ao uso de *arrays* multidimensionais e outros objetos derivados de *arrays* e matrizes. Adicionalmente, a NumPy, disponibiliza operações de alto desempenho incluindo cálculos matemáticos sobre *arrays*, manipulação de formas, ordenação, seleção, entre outras funcionalidades [31].

As bibliotecas Skit-learn e Pandas dependem da instalação do NumPy, pois ele fornece a base necessária para manipulação eficiente de *arrays* e estruturas de dados. Assim, o uso do NumPy torna-se um requisito fundamental para o trabalho.

4.2.3 Matplotlib

A visualização de dados desempenha um papel crucial no processo de análise, o Python destaca-se como uma das linguagens mais indicadas para esta finalidade. Com a biblioteca Matplotlib, é possível fazer a representação dos dados em diversos tipos de gráficos [32]. Entre os tipos de gráficos disponíveis estão os de linhas, dispersão, barras, caixa, histogramas, violino, entre outros [33].

Desta maneira a biblioteca Matplotlib é essencial para a realização da análise e interpretação dos *datasets* ao longo deste projeto.

```python
import matplotlib.pyplot as plt
# Contar as ocorrências de gênero e churn
gender_churn_counts = df.groupby(['Gender', 'Churn']).size().reset_index(name='Count')
# Criar o gráfico de bolha
plt.figure(figsize=(5, 5))
plt.scatter(
    x=gender_churn_counts['Gender'],
    y=gender_churn_counts['Churn'].map({'No': 0, 'Yes': 1}),
    s=gender_churn_counts['Count'] * 40,
    alpha=0.6,
    edgecolors='w',
)
# Ajuste dos limites
plt.ylim(-0.5, 1.5)
plt.xlim(-0.5, 1.5)
plt.title('Distribuição de Gênero e Churn')
plt.xlabel('Gênero')
plt.ylabel('Churn')
plt.grid(True)
plt.xticks(rotation=20)
plt.yticks(rotation=20)
plt.yticks([0, 1], ['Não', 'Sim'])
plt.xticks([0, 1], ['Mulher', 'Homem'])
plt.show()
```

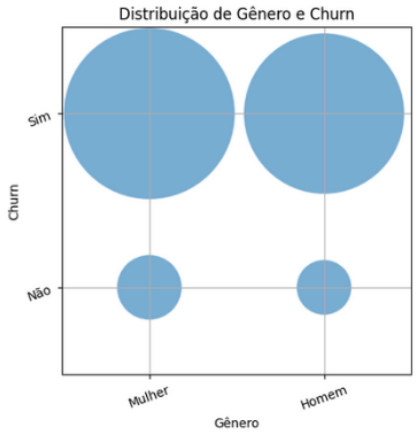

Figura 9 — Exemplo do Uso da Biblioteca Matplotlib

4.2.4 Seaborn

A biblioteca Seaborn, tal como a Matplotlib é usada para fazer a representação gráfica de dados. Esta é mais adequada para abordagem de dados estatísticos exploratórios e inferenciais, sendo desenvolvida sobre o Matplotlib, por consequência necessitando do Matplotlib para o seu funcionamento [34-35].

```python
import seaborn as sns
import matplotlib.pyplot as plt

plt.figure(figsize=(10, 6))
sns.violinplot(x='Churn', y='MonthlyCharges', data=df_IBM, palette="Set2", hue="Churn", legend=False)
plt.title('Distribuição de Cobrança Mensal por Churn')
plt.xlabel('Churn')
plt.ylabel('Cobrança Mensal')
plt.grid(True)
plt.xticks([0, 1], ['Sim', 'Não'])
plt.show()
```

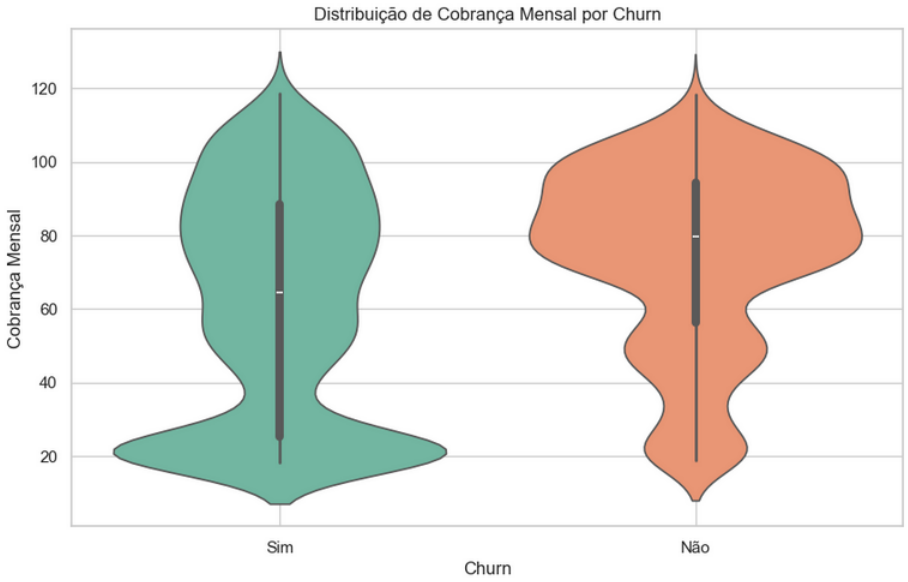

Figura 10 — Exemplo do Uso das Bibliotecas Seaborn e Matplotlib em Conjunto

4.2.5 Scikit-learn

Scikit-learn é uma biblioteca *open-source* para Python, amplamente utilizada para implementar diversos algoritmos de ML. Esta disponibiliza, não só algoritmos de ML, como também ferramentas de avaliação de modelos ML, pré-processamento de dados e visualização de algoritmos [34].

Esta biblioteca é essencial, uma vez que o objetivo principal deste trabalho envolve o treino e comparação de vários modelos de ML cujas implementações podem ser encontradas nesta biblioteca.

5. *Datasets*

Os *datasets* são a base fundamental da ciência de dados, sobre a qual é possível fazer análises de dados, treinar modelos e retirar conclusões sobre determinados problemas. A escolha e o estudo de *datasets* adequados são cruciais para garantir a validade e relevância dos resultados.

Durante a pesquisa efetuada na plataforma Kaggle, foram encontrados vários *datasets*. De modo a selecionar um *dataset* adequado, optou-se por escolher um dos *datasets* com maior relevância, bem como um número significativo de colunas e linhas, independentemente da data de publicação [22]. Paralelamente, de modo comparativo foi também selecionado um *dataset* recente, cuja publicação fosse mais atual e apresentasse uma pontuação de usabilidade equivalente a 10 na plataforma [35].

Entre os dois *datasets* selecionados o primeiro é uma amostra com 7 anos publicada pela IBM, que contém um total de 7043 registos e 21 atributos, de maneira a facilitar a distinção entre os *datasets* utilizados, este *dataset* será referido como o *dataset* proveniente da IBM [22].

A disponibilização deste *dataset* teve como objetivo permitir o estudo de métodos de retenção de clientes no setor das telecomunicações. O *dataset* encontra-se no formato XLS, não contém valores nulos e não se encontra balanceado, possuindo uma percentagem de 26.5% de clientes que fizeram *churn*. Entre os atributos deste *dataset*, o "*customerID*" é um atributo irrelevante, os restantes atributos descrevem informações dos serviços contratados por um cliente, informações da conta do cliente e informações demográficas do mesmo.

Tabela 3 — Descrição dos Atributos do *Dataset* Proveniente da IBM

Atributo	Descrição
CustomerID	Identificador único do cliente.
Gender	Atributo categórico do género masculino e feminino. (*Male, Female*)
SeniorCitizen	Atributo binário que identifica se o cliente é um cidadão sénior.
Partner	Atributo categórico que descreve se o cliente tem um parceiro. (*Yes, No*)
Dependents	Atributo categórico que descreve se o cliente tem dependentes (*Yes*, *No*)
Tenure	Atributo numérico dos meses que o cliente permaneceu na empresa.
PhoneService	Atributo categórico que descreve se o cliente tem um serviço de telefone. (*Yes, No*)
MultipleLines	Atributo categórico que descreve se o cliente um serviço de mais de uma linha telefónica associada ao contrato. (*Yes, No, No phone service*)
InternetService	Atributo categórico que descreve o contrato de internet do cliente. (*DSL, Fiber optic, No*)
OnlineSecurity	Atributo categórico que descreve se cliente está subscrito a um serviço adicional de segurança online. (*Yes, No, No internet service*)
OnlineBackup	Atributo categórico que descreve se o contrato do cliente inclui um serviço de backup de segurança online. (*Yes, No, No internet service*)
DeviceProtection	Atributo categórico que descreve se o cliente está subscrito a um plano de segurança para os equipamentos fornecidos pela empresa. (*Yes, No, No internet service*)
TechSupport	Atributo categórico que descreve se o cliente está subscrito a um serviço de suporte adicional da empresa. (*Yes, No, No internet service*)
StreamingTV	Atributo categórico que descreve se o cliente tem contratado um serviço de transmissão de canais televisivos. (*Yes, No, No internet service*)
StreamingMovies	Atributo categórico que descreve se o cliente tem contratado um serviço de transmissão de filmes. (*Yes, No, No internet service*)
Contract	Atributo categórico dos termos do contrato. (*Month-to-month, One year, Two year*)
PaperlessBilling	Atributo categórico que descreve se a cobrança ao cliente é sem papeis. (*Yes, No*)
PaymentMethod	Atributo categórico que descreve o método de pagamento do serviço. (*Electronic check, Mailed check, Bank transfer (automatic), Credit card*)
MonthlyCharges	Atributo numérico do valor pago pelos clientes mensalmente.
TotalCharges	Atributo numérico do valor total pago pelos clientes.
Churn	Atributo alvo.

Já em relação ao segundo *dataset*, este é uma amostra publicada na plataforma Kaggle em setembro de 2025, que contém um total de 1000 registos e 10 atributos. De maneira a facilitar a distinção entre os *datasets* utilizados este *dataset* será referido como o *dataset* proveniente do Kaggle [35], embora ambos constem nesta plataforma.

O *dataset* em questão foi publicado com o objetivo exploratório de investigar quais os fatores mais influentes para a ocorrência tanto do *churn* como da retenção dos clientes. Este *dataset* está no formato CSV, limpo de ruido, mas não está balanceado, com uma percentagem de 88.3% de clientes que fizeram *churn*. Os atributos existentes neste *dataset* são semelhantes aos do *dataset* proveniente da IBM, contendo informações demográficas, da conta do cliente e dos serviços contratados.

Tabela 4 — Descrição dos Atributos do *Dataset* Proveniente do Kaggle

Atributo	Descrição
CustomerID	Identificador único do cliente.
Age	Atributo numérico da idade do cliente.
Gender	Atributo categórico do género masculino e feminino. (*Male, Female*)
Tenure	Atributo numérico em meses do tempo que o cliente permaneceu na empresa.
MonthlyCharges	Atributo numérico do valor pago pelos clientes mensalmente.
ContractType	Atributo categórico do tipo de contrato. (*Month-to-month, One-year, Two-year*)
InternetService	Atributo categórico que descreve o contrato de internet do cliente. (*DSL, Fiber optic, None*)
TotalCharges	Atributo numérico do valor total pago pelos clientes. *TotalCharges = MonthlyCharges * Tenure*
TechSupport	Atributo categórico que descreve se o cliente tem suporte técnico ou não. (*Yes, No*)
Churn	Atributo alvo.

5.1 Análise Descritiva dos *Datasets*

Através da análise a **figura 11 e 12**, é possível observar que nenhum dos *datasets* está balanceado relativamente ao atributo alvo. Sendo que o *dataset* do Kaggle possui uma maior quantidade de clientes que abandonaram o serviço, pelo contrário o *dataset* da IBM possui uma maior quantidade de clientes que foram retidos.

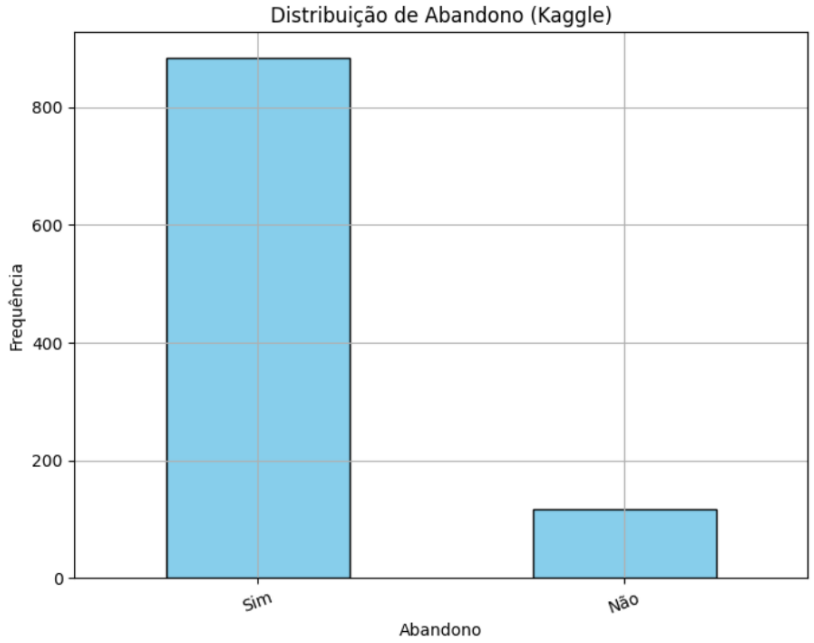

Figura 11 — Gráfico da Distribuição do Atributo *Churn* (Kaggle)

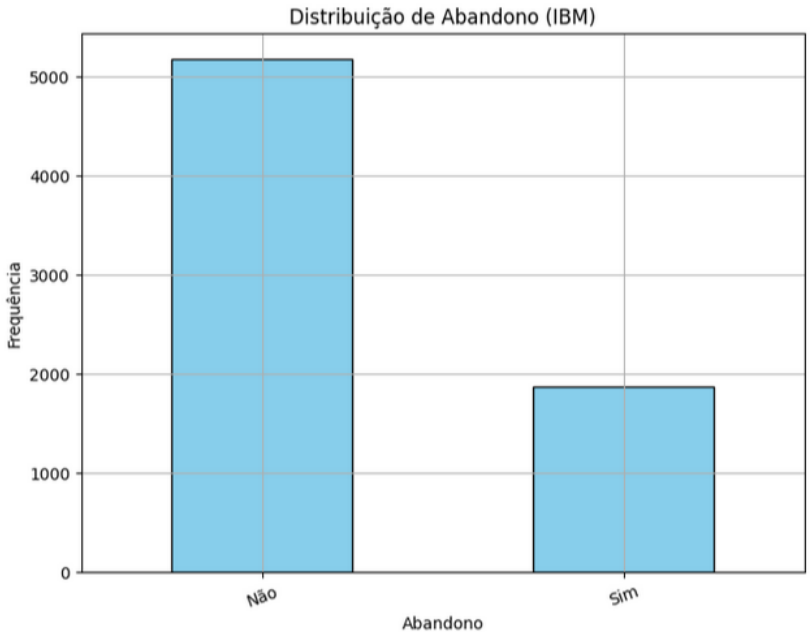

Figura 12 — Gráfico da Distribuição do Atributo *Churn* (IBM)

Nos gráficos de violino das **figuras 13 e 14**, é possível observar uma diferença entre as distribuições de valores do atributo "*Tenure*". O *dataset* da IBM possui uma distribuição mais equilibrada para os valores temporais diferentes, enquanto o *dataset* do Kaggle possui uma densidade maior em valores temporais baixos.

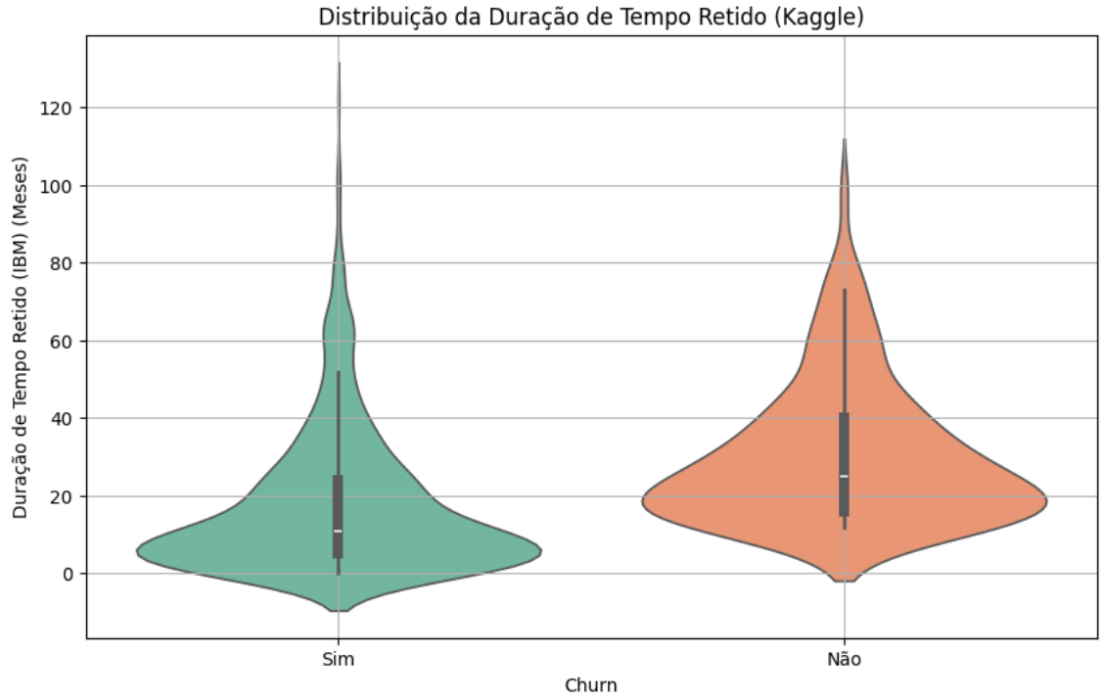

Figura 13 — Gráfico de Violino da Distribuição da *Tenure* pelo *Churn* (Kaggle)

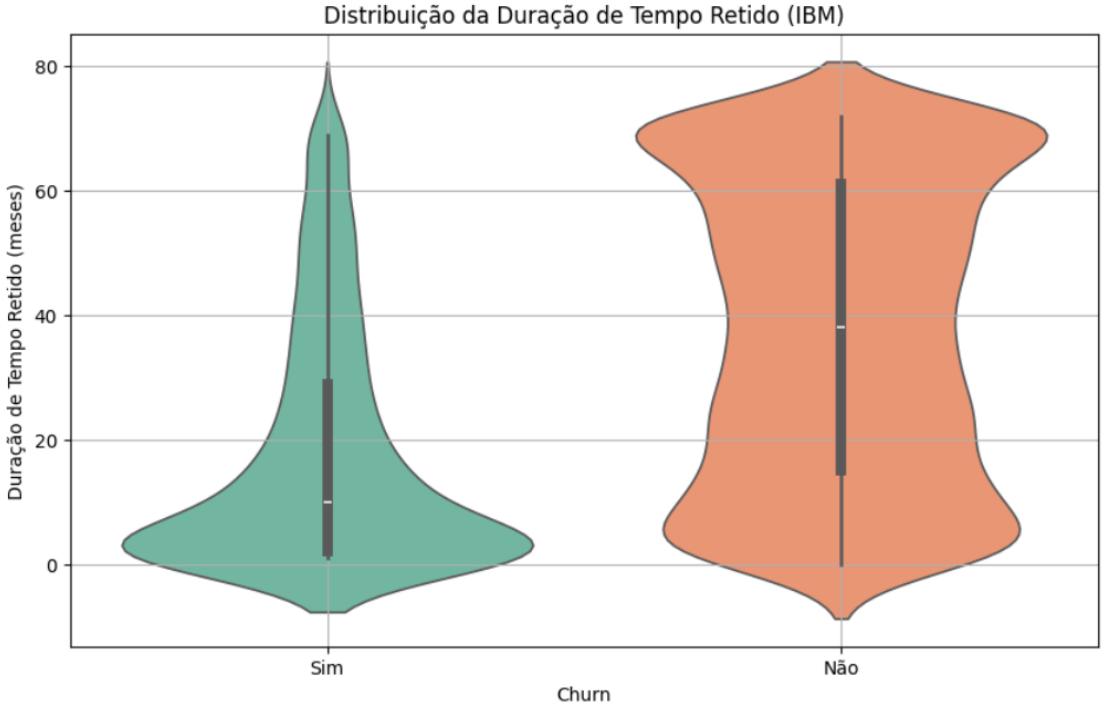

Figura 14 — Gráfico de Violino da Distribuição da *Tenure* pelo *Churn* (IBM)

O *dataset* do Kaggle apresenta uma distribuição uniforme do valor cobrado mensalmente aos clientes, enquanto o *dataset* da IBM demonstra haver uma maior concentração de clientes com valores baixos.

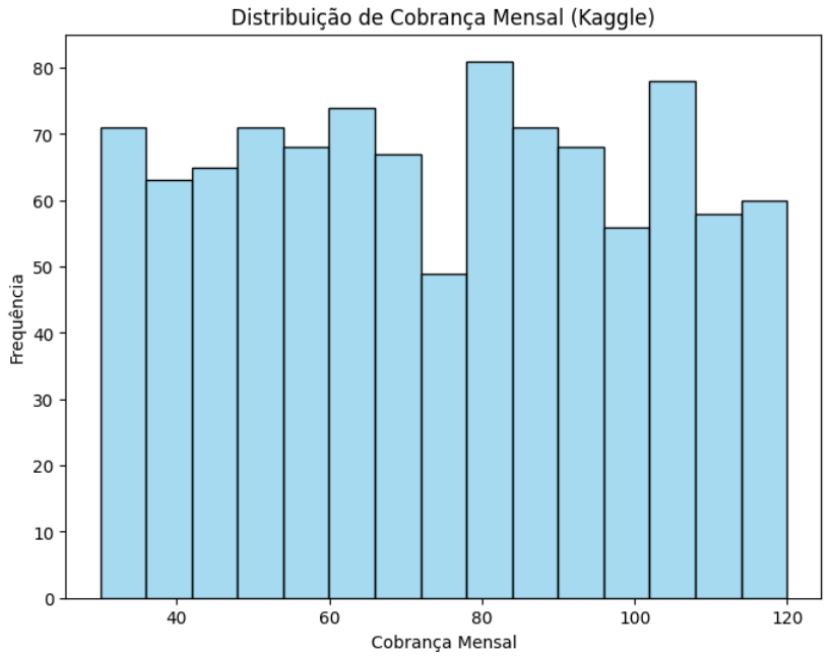

Figura 15 — Gráfico da Distribuição de Cobrança Mensal (Kaggle)

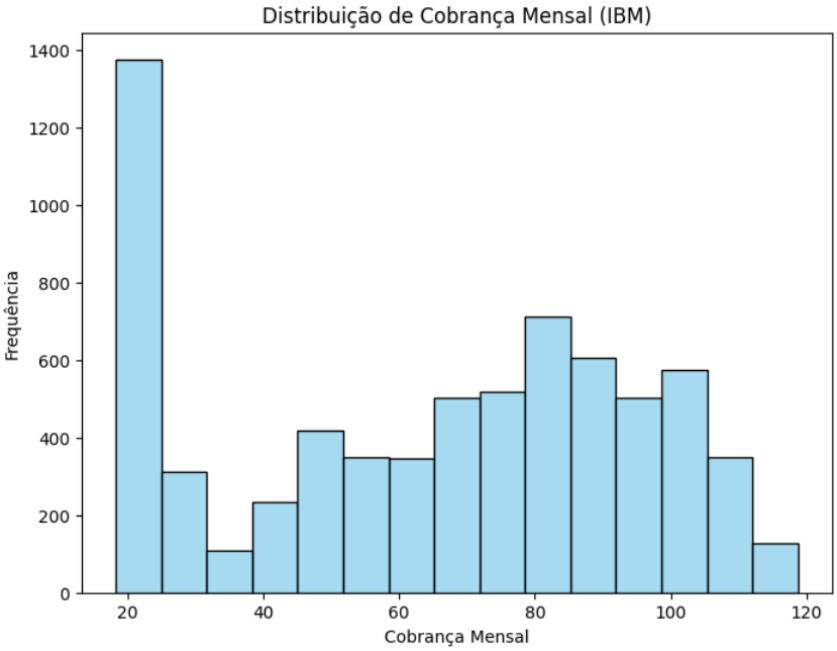

Figura 16 — Gráfico da Distribuição de Cobrança Mensal (IBM)

Pela interpretação das **figuras 17 e 18**, é interessante observar que as posições relativas da mediana de cada gráfico de violino são consistentes para ambos os *dataset*, sendo que o gráfico cuja mediana é mais baixa, é associado à retenção dos clientes.

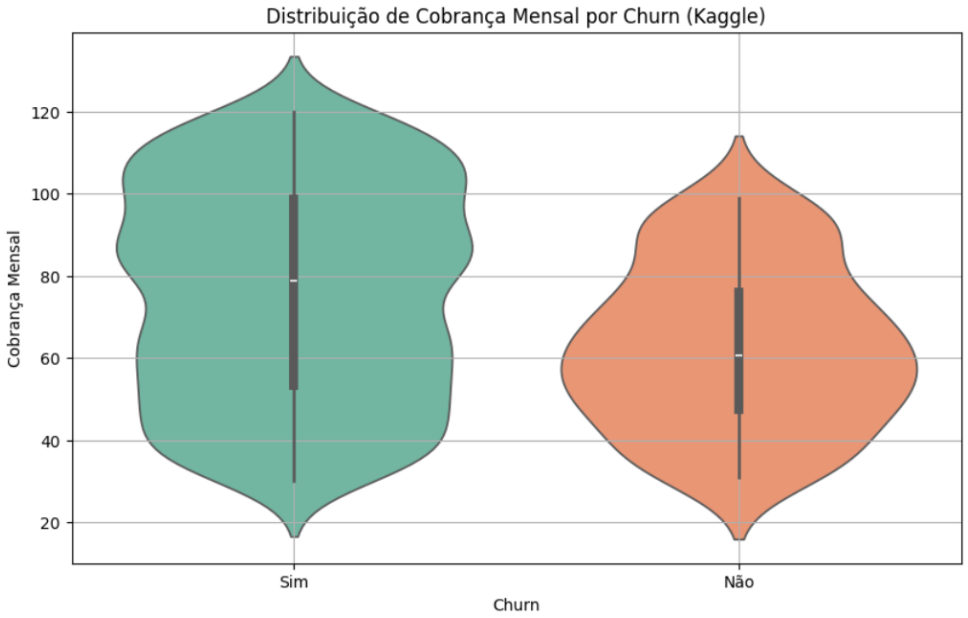

Figura 17 — Gráfico de Violino da Distribuição da Cobrança Mensal pelo *Churn* (Kaggle)

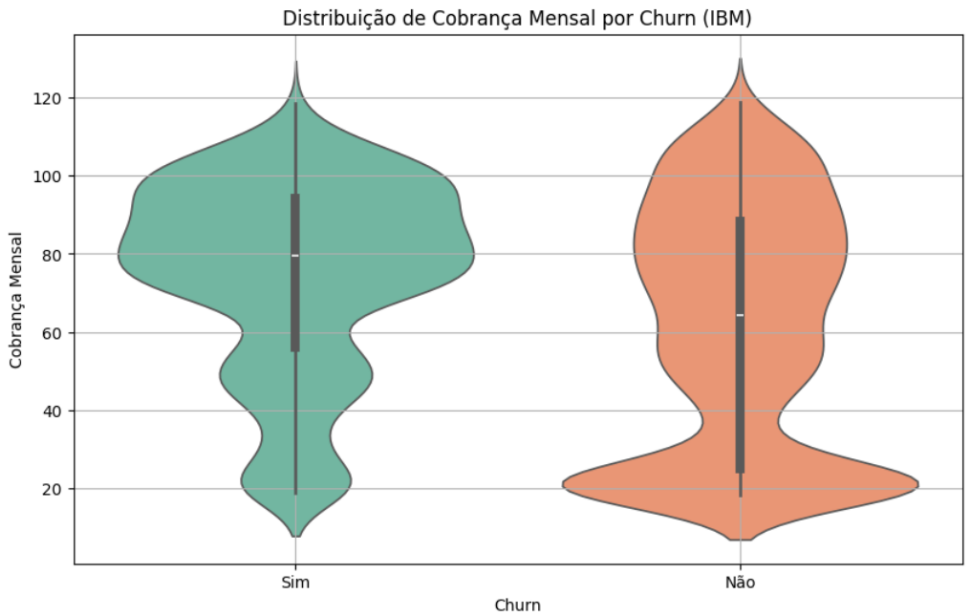

Figura 18 — Gráfico de Violino da Distribuição da Cobrança Mensal pelo *Churn* (IBM)

Os gráficos de bolha na página apresentada, demonstram um comportamento diferenciado entre *datasets* relativamente ao tipo de contratos mensais e à retenção dos clientes. Enquanto no *dataset* do Kaggle todos saíram, no *dataset* da IBM a bolha com maior densidade está inserida nesse mesmo grupo. Este fator pode ser explicado pela possível diferença entre a qualidade de serviços.

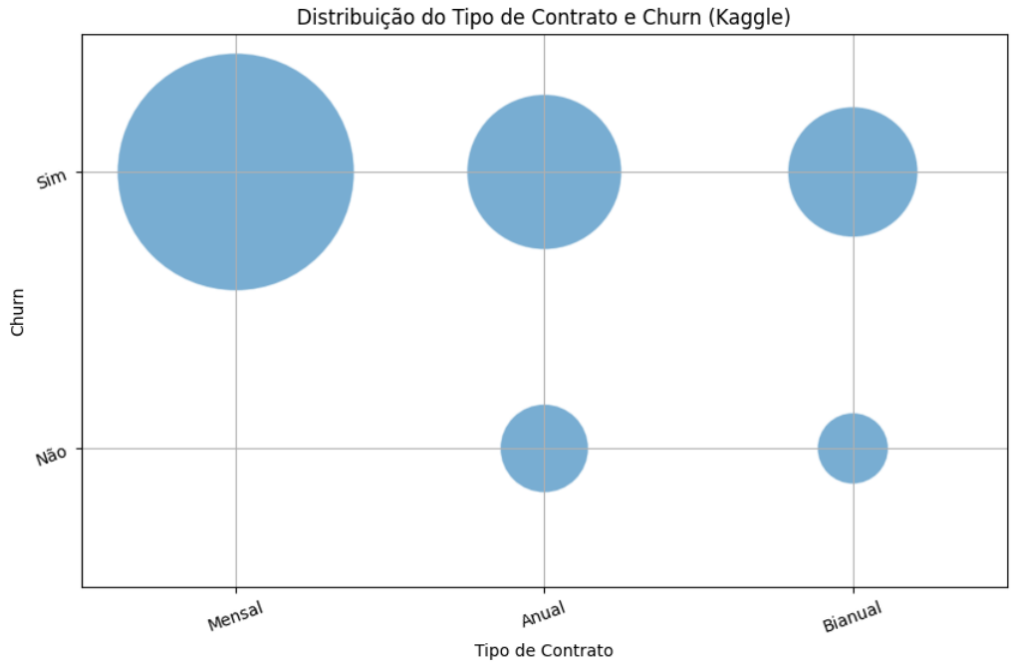

Figura 19 — Gráfico de Bolhas entre a Distribuição do Tipo de Contrato e o *Churn* (Kaggle)

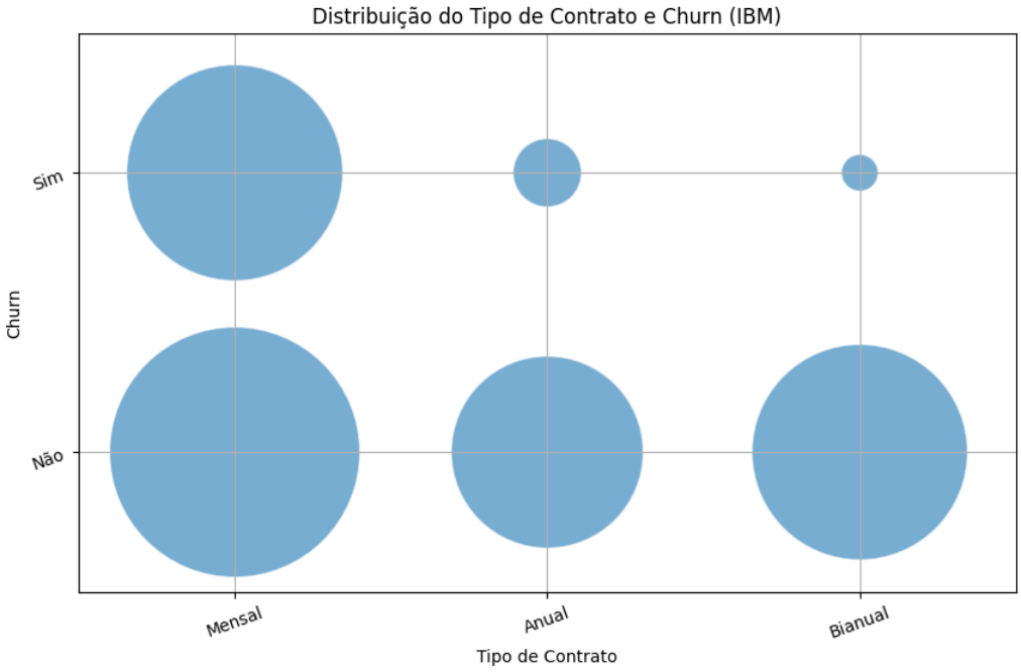

Figura 20 — Gráfico de Bolhas entre a Distribuição do Tipo de Contrato e o *Churn* (IBM)

Relativamente aos serviços de internet, no *dataset* do Kaggle o gráfico demonstra que o abandono aconteceu de forma uniforme independentemente do tipo de serviço. No *dataset* da IBM é possível observar que o serviço de internet com mais tendência a abandono foi a fibra ótica.

Adicionalmente, é contrastante que nenhum cliente do *dataset* do Kaggle sem serviço de internet foi retido, enquanto no *dataset* da IBM é apresentado um número considerável de clientes retidos.

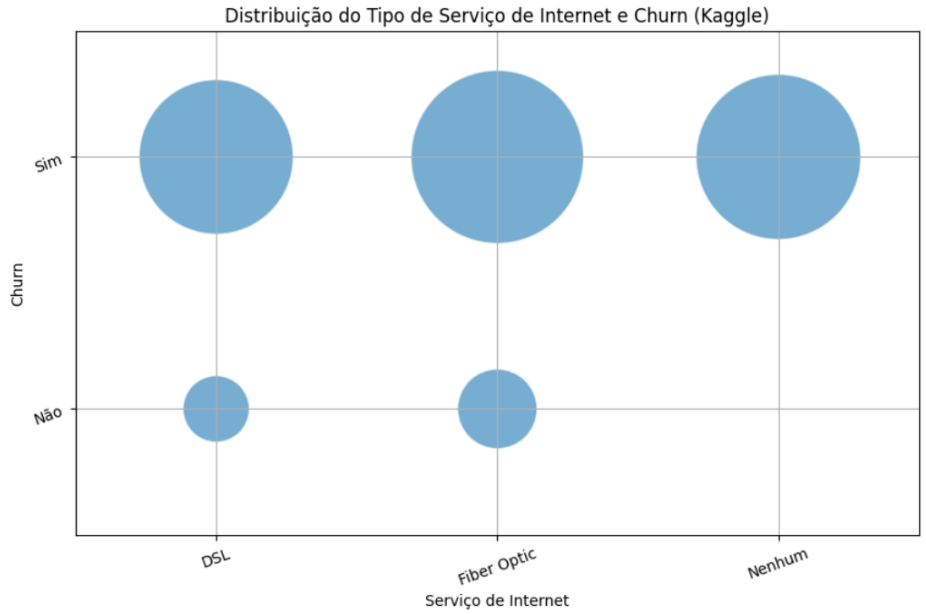

Figura 21 — Gráfico de Bolhas entre a Distribuição do Serviço de Internet e o *Churn* (Kaggle)

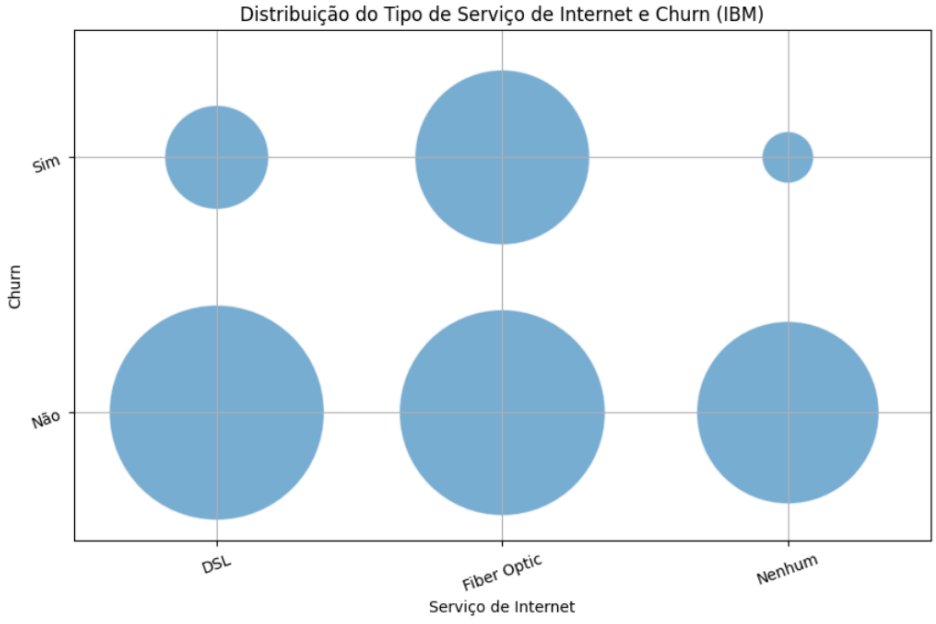

Figura 22 — Gráfico de Bolhas entre a Distribuição do Serviço de Internet e o *Churn* (IBM)

Nos gráficos de bolhas das **figuras 23 e 24**, ao analisar o gráfico referente ao *dataset* da IBM, observa-se que a maioria dos clientes retidos não possuía um serviço de suporte técnico. Por outro lado, no *dataset* do Kaggle, todos os clientes sem suporte técnico acabaram por abandonar o serviço.

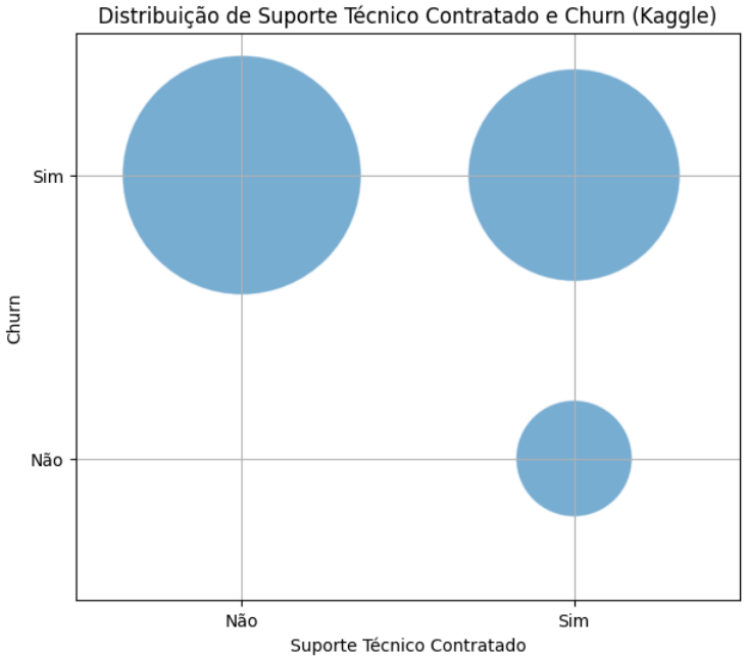

Figura 23 — Gráfico de Bolhas entre a Distribuição do Suporte Técnico Contratado e o *Churn* (Kaggle)

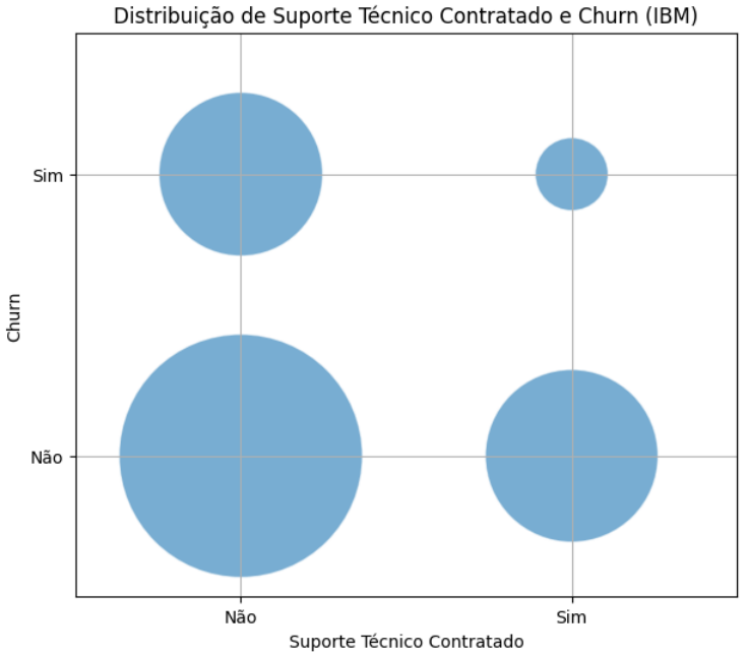

Figura 24 — Gráfico de Bolhas entre a Distribuição do Suporte Técnico Contratado e o *Churn* (IBM)

Na **figura 25** é possível observar a distribuição do *churn* entre os tipos de clientes cujo pagamento não envolve papéis, ou seja, o pagamento de forma eletrónica e os clientes cujo pagamento não é eletrónico. Esta distribuição indica uma tendência maior dos clientes cujo pagamento é eletrónico para abandonarem o serviço.

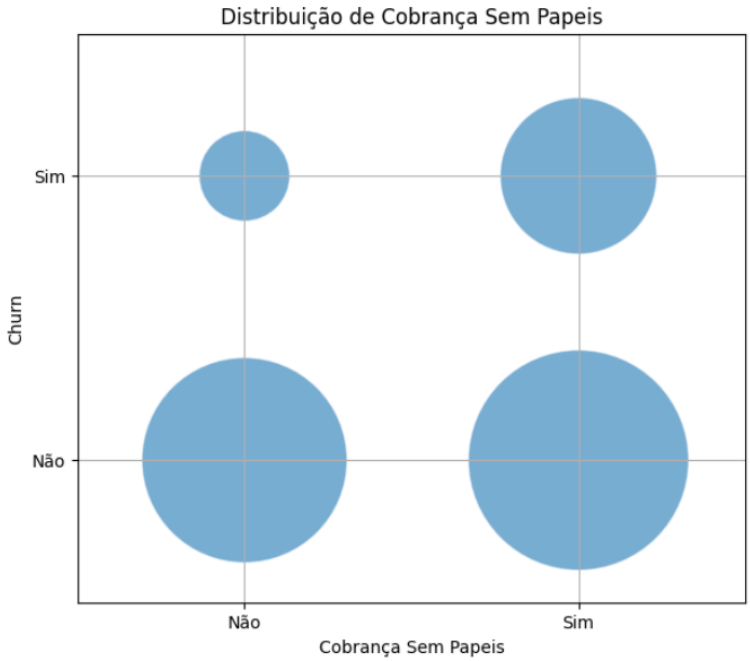

Figura 25 — Gráfico de Bolhas entre a Distribuição de Cobrança Sem Papeis e o *Churn* (IBM)

5.2 Reflexões

A analise descritiva dos *datasets* selecionados, permitiu identificar as principais diferenças das distribuições entre eles, e explorar cada atributo focando a análise apenas naqueles que foram considerados relevantes.

O *dataset* do Kaggle demonstrou uma distribuição do valor cobrado mensalmente mais equilibrada, enquanto o *dataset* da IBM possuía uma representatividade maior de valores baixos. Em contraste os valores do tempo que o cliente possuiu o serviço estão mais equilibrados no dataset da IBM, uma vez que o *dataset* do Kaggle possui poucos exemplos com períodos de retenção alargados.

Algumas diferenças na distribuição do tipo do contrato e do serviço de internet em relação ao *churn*, podem ser influenciadas por fatores como a qualidade de serviço, contexto temporal, social e cultural. Apesar dos atributos mencionados terem distribuições diferentes, estes adicionam informações importantes para a previsão do *customer churn*.

6. Treino e Avaliação

A seguinte fase do projeto, tem como objetivo treinar e avaliar modelos de ML. Estes modelos serão treinados utilizando diversos algoritmos de ML nos dois *datasets* selecionados. O treino e a avaliação dos modelos, permitirão verificar o desempenho destes na previsão do *customer churn*.

Caso os modelos de ML demonstrem um bom desempenho, viável de fazer previsões corretas, estes poderão ser futuramente implementados num website ou aplicação, auxiliando as empresas de telecomunicações na retenção dos clientes através do suporte na identificação dos clientes com maior probabilidade de abandono.

Neste capítulo, é descrito todo o processo de treino e avaliação dos algoritmos ML para o problema de classificação apresentado.

Para treinar modelos de ML, é necessário estabelecer um processo estruturado que inicia com o carregamento do *dataset* num *DataFrame* e termina com a análise dos resultados da avaliação de cada modelo. O processo mencionado pode ser dividido em 4 etapas.

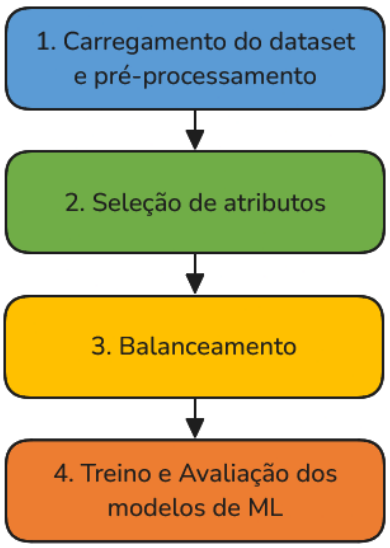

Figura 26 — Fluxo de Trabalho para Treino e Avaliação dos Modelos ML

Com o objetivo de esclarecer o fluxo do processo ilustrado, no presente capítulo, cada etapa será descrita em detalhe, incluindo explicações das decisões tomadas ao longo do desenvolvimento.

6.1 Carregamento e Pré-processamento dos *Datasets*

O processo é iniciado com o carregamento do *dataset* escolhido num *Dataframe*, que corresponde a um objeto da biblioteca Pandas utilizado para armazenar e manipular dados de forma estruturada.

De seguida no caso do d*ataset* possuir atributos categóricos, deve ser feita a sua conversão para atributos numéricos, uma vez que nem todos os algoritmos de ML suportam atributos categóricos. Para isso, neste trabalho foi utilizada a técnica *one-hot encoding*, a qual cria um atributo binário para cada categoria pertencente ao atributo original, este valor binário representa a presença da categoria no atributo original [36].

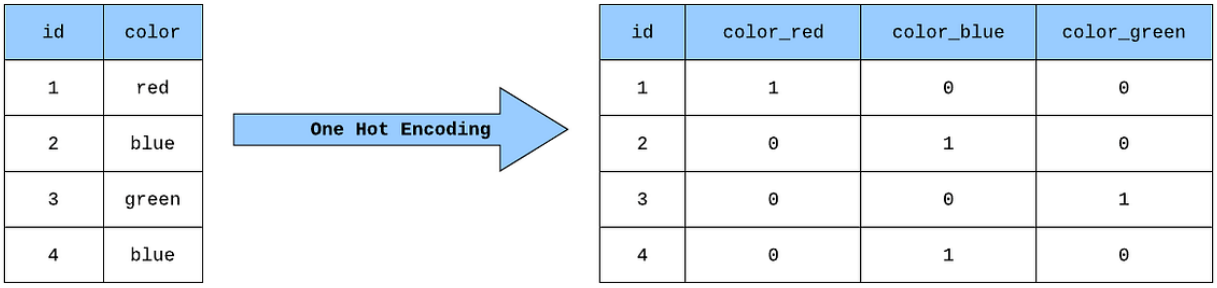

Figura 27 — Funcionamento da Técnica *One-Hot Encoding* (fonte: [37])

No *dataset* do Kaggle os atributos afetados foram: “*Gender*”, “*TechSupport*”, “*Churn*”, “*ContractType*” e “*InternetService*”. Já no dataset da IBM foram afetados os seguintes atributos: “*Contract*”, “*InternetService*”, “*PaymentMethod*”, “*Gender*”, “*TechSupport*”, “*Churn*”, “*Partner*”, “*Dependents*”, “*PhoneService*”, “*MultipleLines*”, “*OnlineBanking*”, “*DeviceProtection*”, “*StreamingTV*”, “*StreamingMovies*” e “*PaperlessBilling*”.

Por último, na existência de atributos numéricos é importante ter em conta a sua normalização, de modo a evitar que atributos com escalas diferentes tenham pesos diferentes no treino do modelo de ML. Desta maneira, neste trabalho foi utilizado o método *StrandardScaler* [38], da biblioteca Scikit-learn para garantir que todos os atributos numéricos estejam na mesma escala.

No contexto dos dois *datasets* selecionados para este trabalho os atributos numéricos influenciados foram “*Tenure*”, “*MonthlyCharges*” e “*TotalCharges*”.

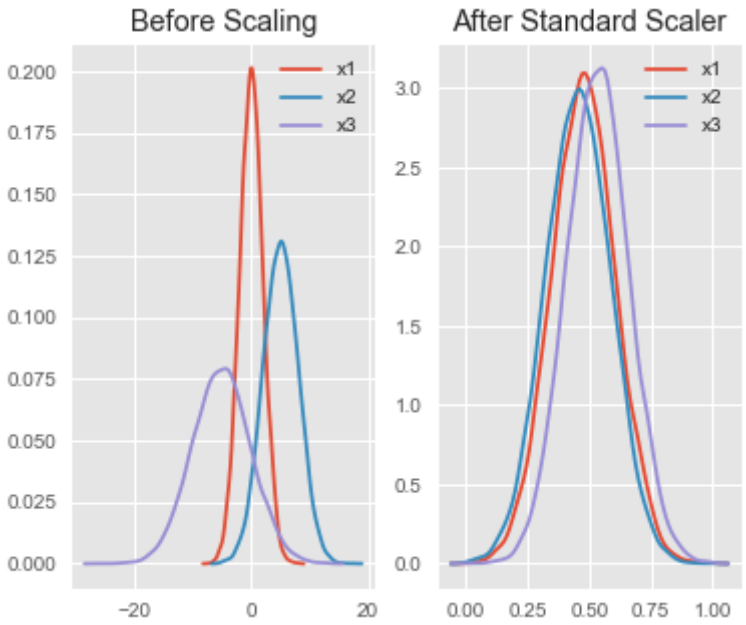

Figura 28 — Ilustração do método *StandardScaler* (fonte: [39])

6.2 Seleção de Atributos

A seleção dos atributos é uma parte essencial do processo de treino de modelos de ML, que tem como objetivo filtrar os atributos menos importantes e reduzir o número de atributos que um modelo terá como entrada.

Um número elevado de atributos pode ser algo negativo, uma vez que, à medida que o número de atributos aumenta, o volume de dados torna-se mais esparso, além disso o número de registos necessários para treinar modelos com um elevado desempenho preditivo, aumenta exponencialmente com o número de atributos preditivos. Por esta razão devemos fazer uma seleção dos atributos mais relevantes de um *dataset* de modo a beneficiar os resultados preditivos.

Existem várias maneiras de selecionar atributos, neste projeto optou-se pelo método baseado em filtros, através do uso da matriz de correlação de cada *dataset*. A matriz de correlação utiliza a métrica estatística de *correlação de Pearson* (*r*) para avaliar a relevância dos atributos, sendo possível ver as relações entre atributos, destacando-se relações fortes com o atributo alvo e idealizando relações fracas entre os atributos de entrada. Os valores de *r* variam entre -1 e 1, onde -1 indica uma correlação negativa perfeita entre dois atributos, 0 significa a não existência de correlação e 1 indica uma correlação positiva perfeita.

Para a seleção neste projeto, foi definido que qualquer atributo com um valor absoluto de *r* em relação ao atributo alvo, menor ou igual a 0.15 seria descartado, com exceção dos atributos criados durante o *one-hot encoding,* caso alguma categoria obtenha um valor válido de *r*.

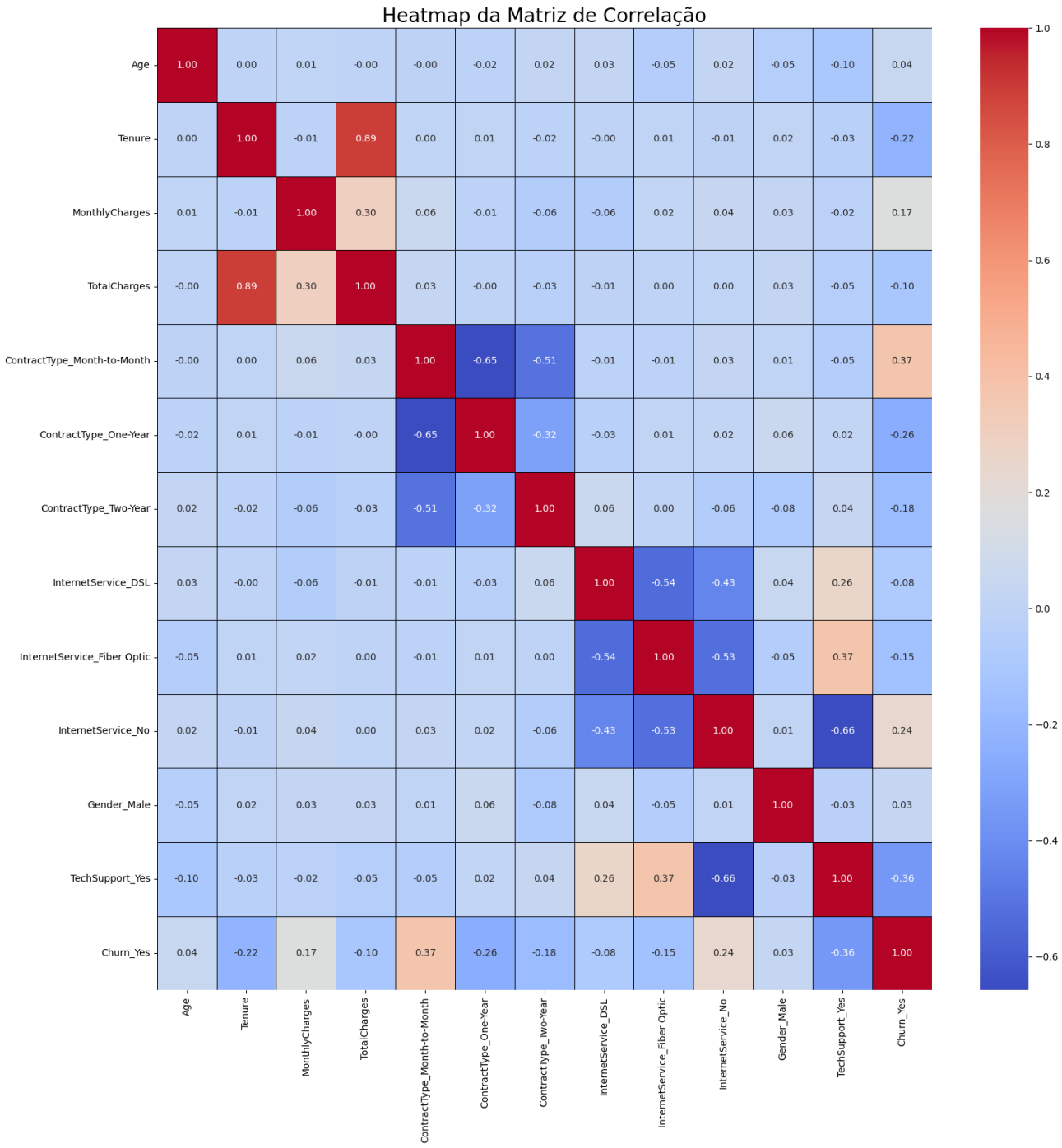

Figura 29 — Matriz de Correlação do *Dataset* Proveniente do Kaggle

No caso do *dataset* do Kaggle, os atributos “*Age*”, “*TotalCharges*” e “*Gender*” foram descartados.

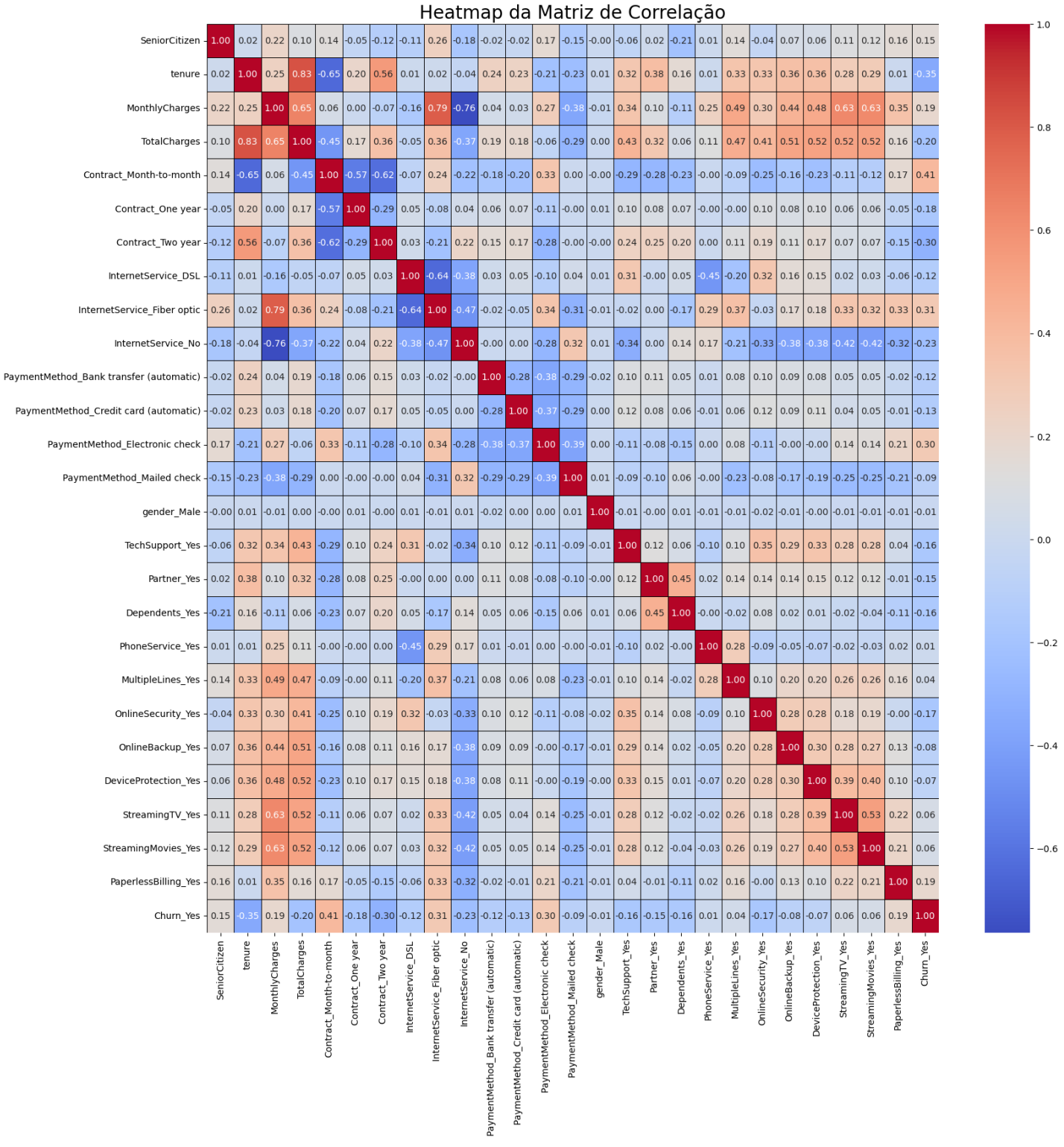

Figura 30 — Matriz de Correlação do *Dataset* Proveniente da IBM

Já no caso do *dataset* da IBM, foram descartados os atributos: "*SenioCitizen*", "*Gender*", "*Partner*", "*PhoneService*", "*MultipleLines*", "*OnlineBackup*", "*DeviceProtection*", "*StreamingTV*" e "*StreamingMovies*". O atributo "*TotalCharges*" foi adicionalmente descartado pela correlação forte com "*Tenure*" e "*MonthlyCharges*" de modo a evitar redundância de atributos.

6.3 Balanceamento

Os *datasets* escolhidos demonstraram não estar balanceados com a observação feita no **Capítulo 5.2**. A existência deste desequilíbrio do atributo alvo poderia impactar negativamente o desempenho dos modelos ML, uma vez que teriam maior facilidade em aprender a classificar a classe maioritária.

De maneira a mitigar este problema, neste projeto foi optado por combinar técnicas de *oversampling* para equilibrar as classes e de *undersampling* para melhorar a separação entre classes, removendo instâncias ambíguas ou redundantes criadas no processo.

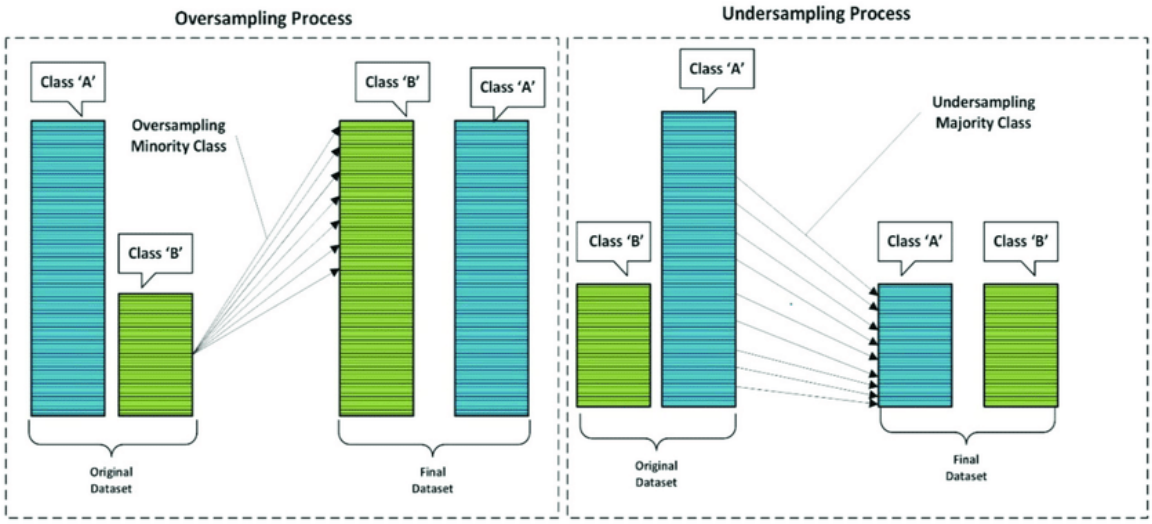

Figura 31 — Funcionamento do *Oversampling* e *Undersampling* (fonte: [40])

Em primeira instância, foi aplicada a técnica de *oversampling* SMOTE (*Synthetic Minority Oversampling Technique*). A técnica SMOTE replica exemplos da classe minoritária de modo a aumentar o número de registos da classe menos representada, alcançando o balanceamento entre as classes [41].

De seguida, foi utilizada a técnica de *undersampling* Tomek Links. Esta técnica remove instâncias muito próximas no espaço de atributos com classes opostas. Visando reduzir o ruido e ambiguidade na separação entre classes [41].

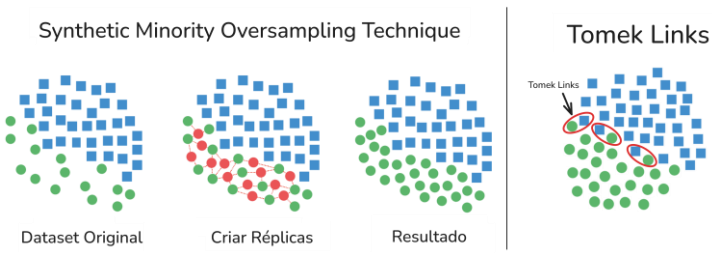

Figura 32 — Ilustração das Técnicas de Balanceamento SMOTE e Tomek Links

6.4 Treino e Avaliação dos Modelos de ML

A seleção dos modelos a serem treinados e avaliados nesta secção, foi feita através dos algoritmos mais relevantes entre os resultados da análise do estado da arte. Sendo assim, os algoritmos selecionados foram: RF (*Random Forest*), SVM (*Support Vector Machine*), XGBoost (*Extreme Gradient Boosting Classifier*), DT (*Decision Tree*), LR (*Logistic Regression*) e NB (*Naïve Bayes*).

Neste processo foi utilizada a validação cruzada configurada com o parâmetro k=10, dividindo assim os *datasets* em 10 *folds* iguais. Assim como mencionado em "**Métricas de Avaliação para Problemas de Classificação**" todos os registos irão contribuir para o teste e avaliação de cada modelo.

O treino do *dataset* proveniente do Kaggle com os atributos de entrada "*Tenure*", "*MonthlyCharges*", "*ContractType*", "*InternetService*" e "*TechSupport*", resultou nos seguintes resultados da **tabela 5**:

Tabela 5 — Avaliação dos Modelos ML com o *Dataset* do Kaggle

Algoritmos	Validação Cruzada				
	Precision	*Accuracy*	*Recall*	*F1-Score*	*AUC*
RF	**100%**	**100%**	**100%**	**100%**	**100%**
SVM	**100%**	95.3% ±1%	90.6% ±3%	95.3% ±1%	98.6% ±1%
XGBoost	**100%**	**100%**	**100%**	**100%**	**100%**
DT	**100%**	**100%**	**100%**	**100%**	**100%**
LR	**100%**	94.9% ±1%	89.8% ±3%	94.9% ±1%	98.5% ±1%
NB	**100%**	91.9% ±2%	83.9% ±4%	91.8% ±2%	98.6% ±1%

Com a observação destes resultados é possível concluir que o algoritmo *Naïve Bayes* será o menos adequado. Sendo a *recall* uma métrica importante no contexto deste problema, como o resultado da *recall* foi o pior e a sua *performance* avaliada pelas outras métricas foram as piores com exceção da *precision*, este será o algoritmo menos adequado para implementação numa aplicação do mundo real.

Relativamente aos algoritmos com as melhores *performances*, as *Random Forest*, XGBoost e as *Decision Tree*, uma vez que obtiveram resultados equivalentes em todas as métricas com o uso do *dataset* proveniente do Kaggle, acaba por ser inconclusivo, não sendo possível salientar o desempenho de um algoritmo específico.

Já o *dataset* proveniente da IBM, foi treinado com os atributos de entrada "*Tenure*", "*MonthlyCharges*", "*Contract*", "*InternetService*", "*PaymentMethod*", "*TechSupport*", "*Dependents*", "*OnlineSecurity*" e "*PaperlessBilling*". A avaliação dos modelos ML foi refletida nos seguintes resultados da **tabela 6**:

Tabela 6 — Avaliação dos Modelos ML com o *Dataset* da IBM

Algoritmos	Validação Cruzada				
	Precision	*Accuracy*	*Recall*	*F1-Score*	*AUC*
RF	**85.1% ±1%**	**86% ±1%**	**88.8% ±2%**	**86% ±1%**	**93.1% ±1%**
SVM	74.9% ±1%	77.2% ±1%	85.1% ±2%	77% ±1%	86.7% ±1%
XGBoost	82.7% ±2%	83.3% ±1%	86.3% ±2%	83.3% ±1%	91% ±1%
DT	80.2% ±2%	80.2% ±2%	81.4% ±2%	80.2% ±2%	80.4% ±2%
LR	78.4% ±1%	79.2% ±1%	83.2% ±1%	79.1% ±1%	87.1% ±1%
NB	77% ±1%	77.9% ±1%	82.4% ±2%	77.8% ±1%	85.7% ±1%

Sem margem de dúvida, o *dataset* da IBM contribuiu melhor para a diferenciação da *performance* dos vários algoritmos. Este facto pode dever-se ao seu tamanho e representatividade da população.

Na **tabela 6** é possível identificar uma preferência pelo desempenho do algoritmo *Random Forest*. As *Random Forest* obtiveram os melhores resultados, com uma *recall* de 88.8% e um desvio padrão de 2%, significando assim que tem uma boa capacidade preditiva de falsos negativos, sendo este o aspeto mais importante no contexto do problema abordado.

6.5 *Feature Importance*

Alguns dos algoritmos usados neste estudo são baseados em árvores ou modelos lineares, estes determinam a relevância de cada atributo para a classificação, sendo este termo chamado de *feature importance* [42]. Quanto maior o resultado atribuído a um atributo significa que este é mais importante para a previsão.

Sendo assim os algoritmos que contribuíram para identificação dos atributos mais importantes, foram as *Random Forest*, SVM, XGBoost, *Decision Tree* e *Logistic Regression.*

De seguida, serão apresentados os resultados gráficos da *feature importance* que cada algoritmo identificou para cada *dataset*:

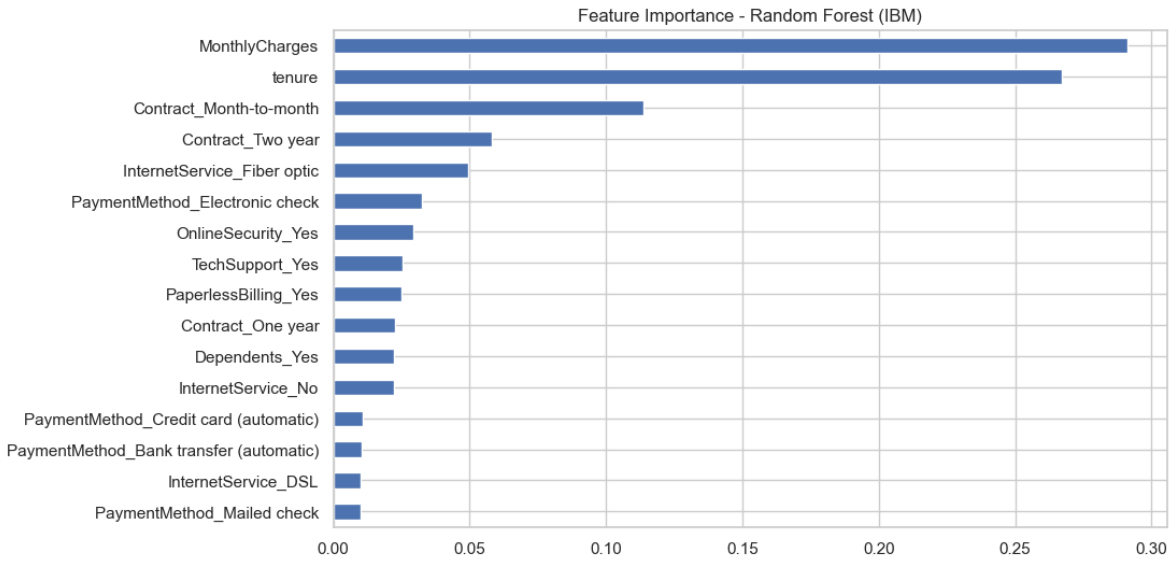

Figura 33 — *Random Forest Feature Importance* com o *Dataset* da IBM

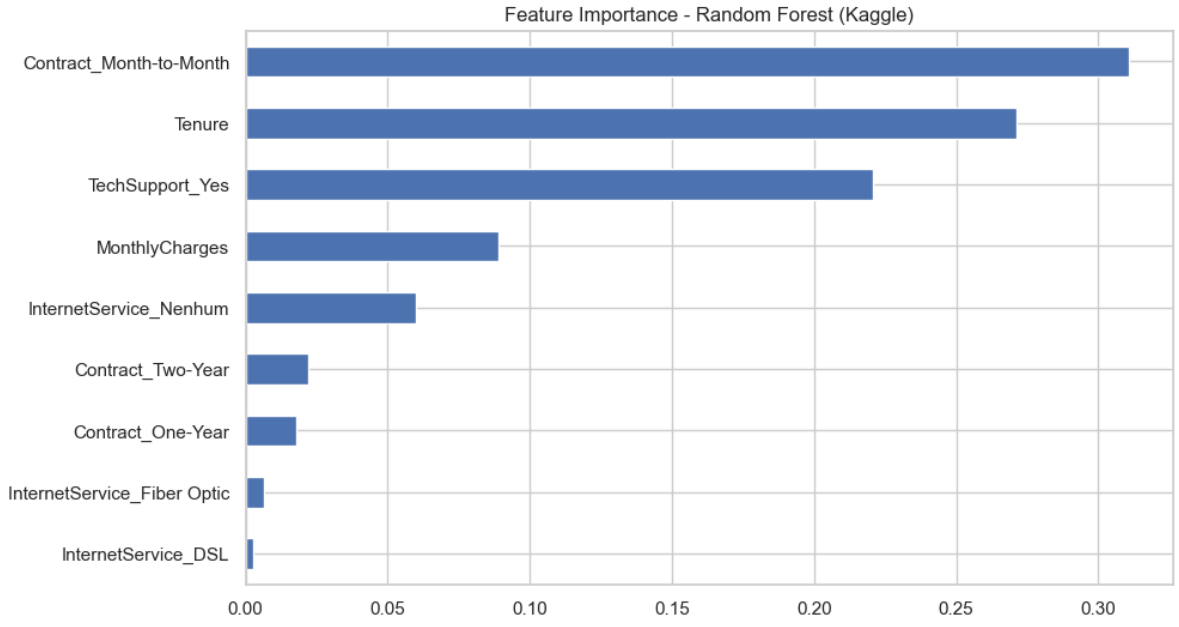

Figura 34 — *Random Forest Feature Importance* com o *Dataset* do Kaggle

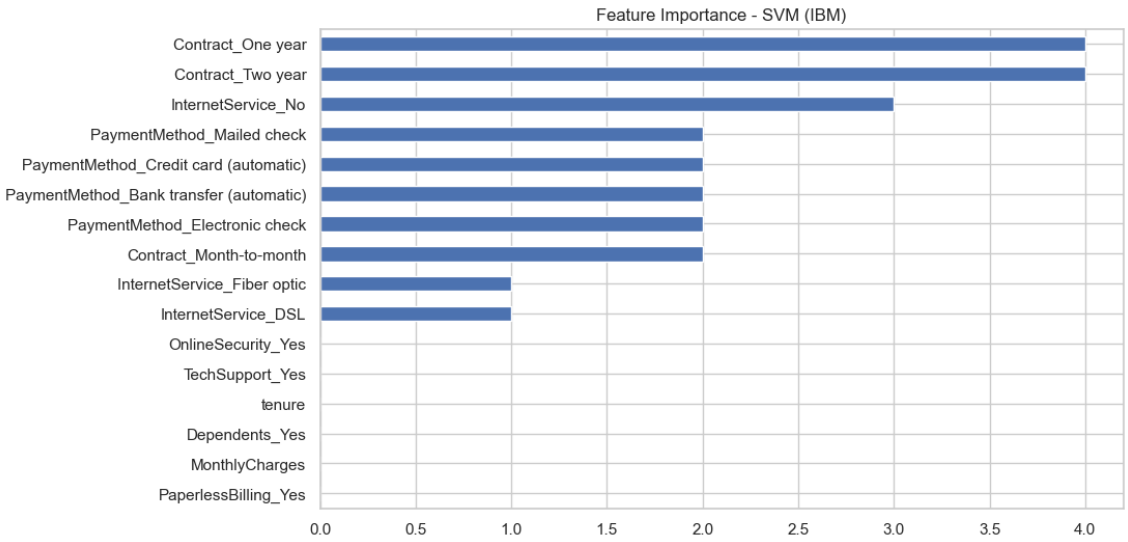

Figura 35 — SVM *Feature Importance* com o *Dataset* da IBM

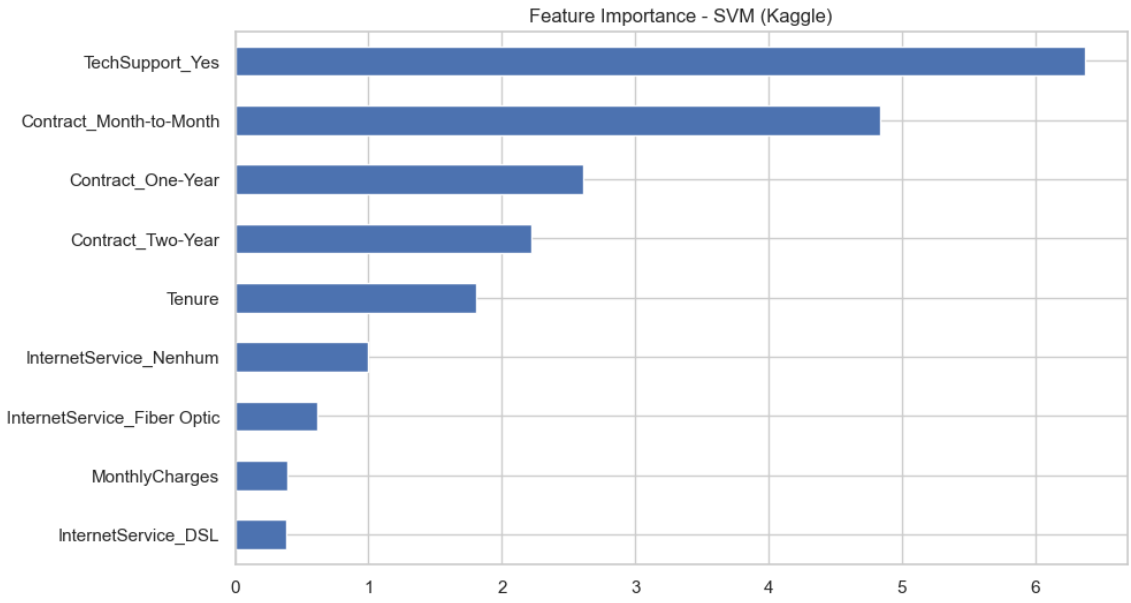

Figura 36 — SVM *Feature Importance* com o *Dataset* do Kaggle

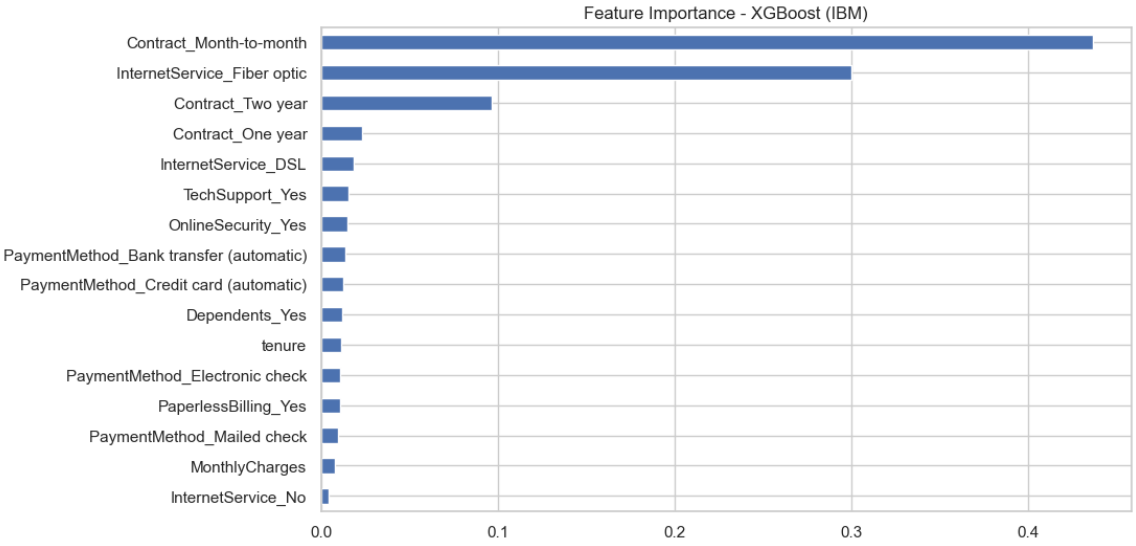

Figura 37 — XGBoost *Feature Importance* com o *Dataset* da IBM

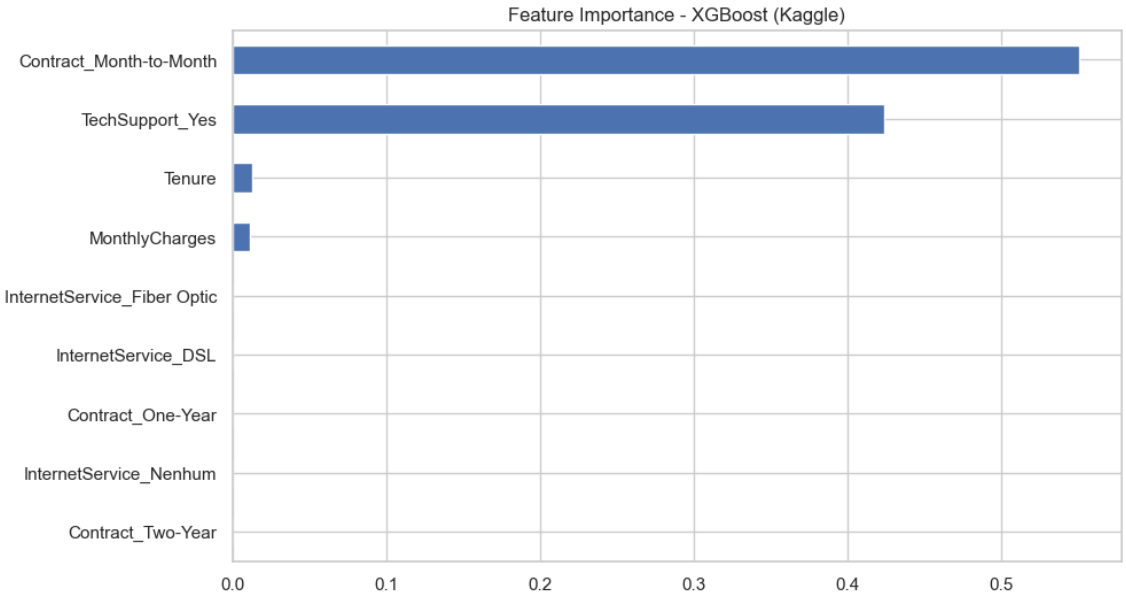

Figura 38 — XGBoost *Feature Importance* com o *Dataset* do Kaggle

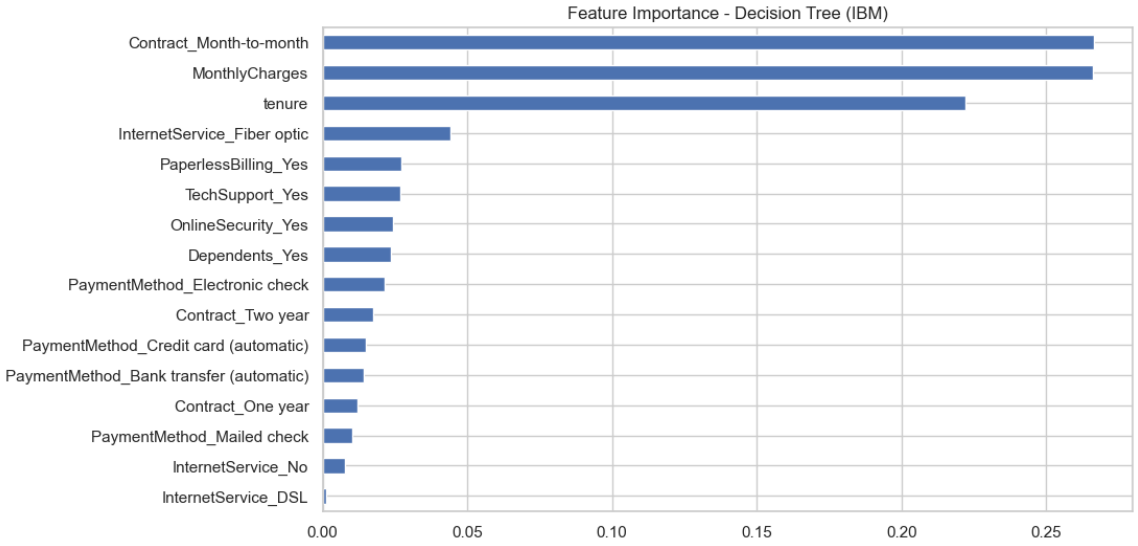

Figura 39 — *Decision Tree Importance* com o *Dataset* da IBM

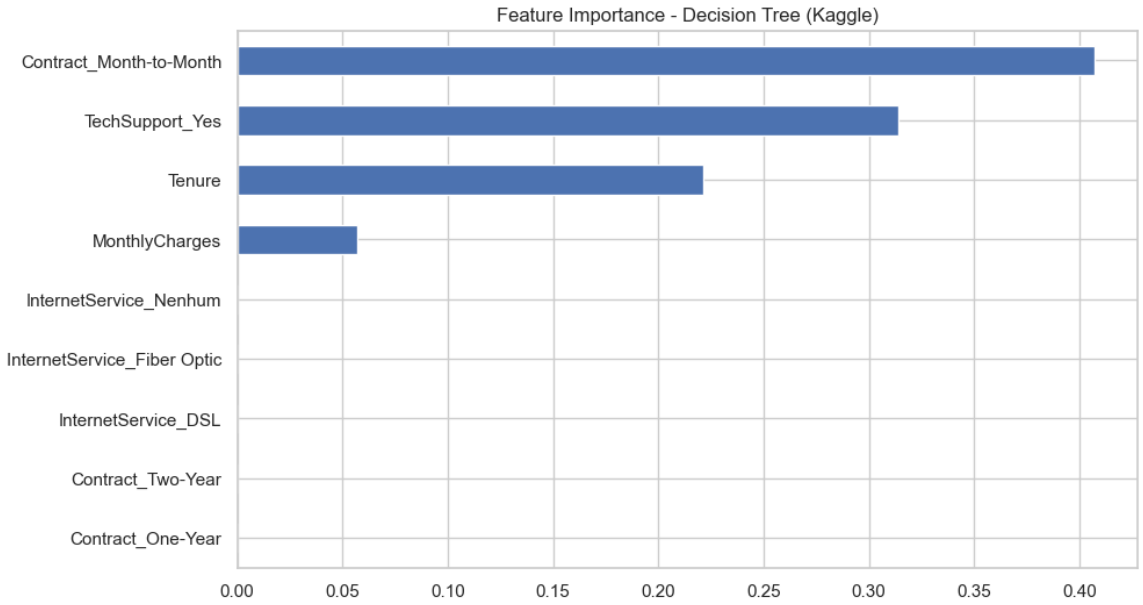

Figura 40 — *Decision Tree Feature Importance* com o *Dataset* do Kaggle

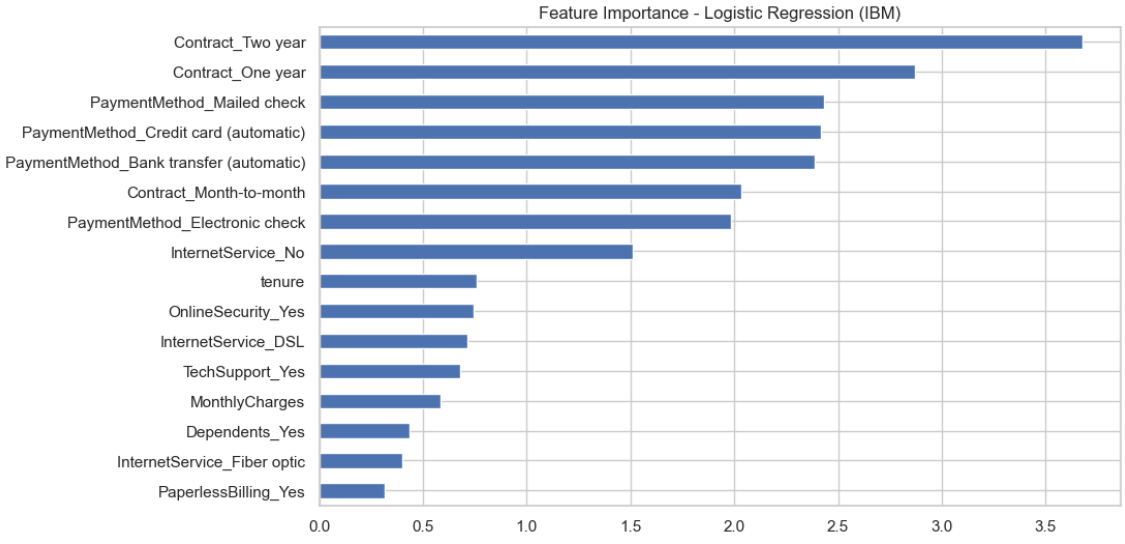

Figura 41 — *Logistic Regression Importance* com o *Dataset* da IBM

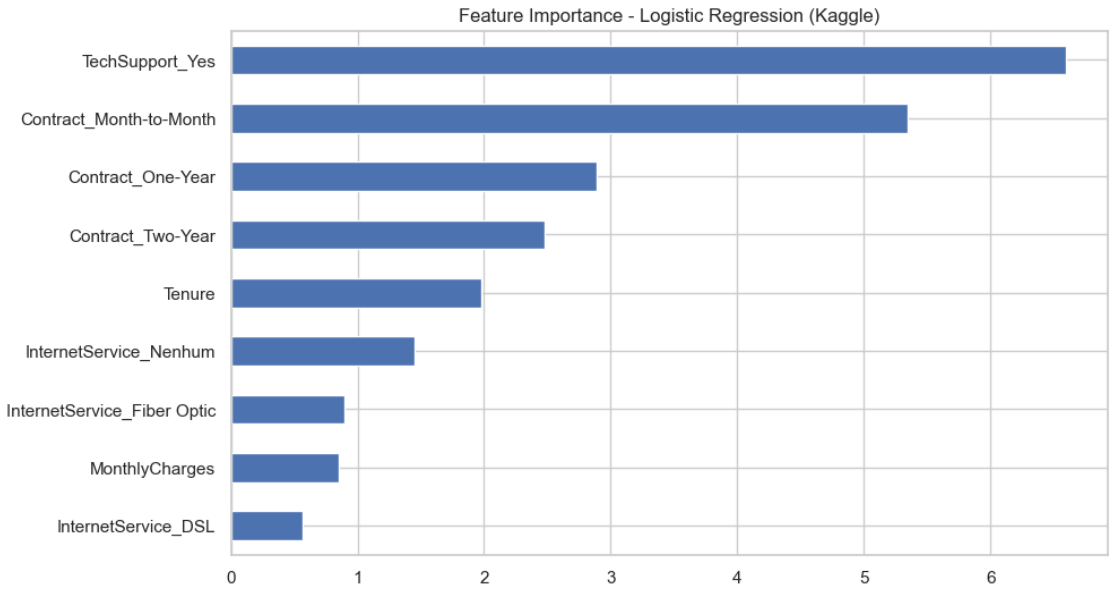

Figura 42 — *Logistic Regression Importance* com o *Dataset* do Kaggle

Entre os resultados obtidos, é possível interpretar para os diferentes modelos utilizados nos dois *datasets*, a importância de cada atributo.

As *Random Forest* deram principal importância ao tempo que o cliente possuiu o serviço, ao preço cobrado mensalmente e aos contratos de tipo mensal, de forma consistente.

O modelo SVM atribuiu muita importância à posse de um serviço de suporte quando treinado com o *dataset* do Kaggle, enquanto ao ser treinado com o *dataset* da IBM não associou qualquer relevância ao mesmo atributo. Este não valorizou muito o preço cobrado mensalmente, nem o tempo que o cliente possuiu o serviço, ao contrário das *Random Forest*. Atribuindo muita relevância aos vários tipos de contrato existentes.

O XGBoost focou-se principalmente nos atributos de contrato do tipo mensal para os dois *datasets*, posse de um serviço de suporte, apenas quando treinado com o *dataset* do Kaggle, e posse de um serviço de internet de fibra ótica quando treinado como o *dataset* da IBM.

O modelo de *Decision Tree*, valorizou principalmente o tempo que o cliente possuiu o serviço e contratos do tipo mensal. No modelo treinado com o *dataset* do Kaggle ele valorizou também a posse de um serviço de suporte. De maneira diferente o modelo treinado com o *dataset* da IBM valorizou o preço cobrado mensalmente.

A *Logistic Regression*, foi o modelo que atribuiu importância de forma mais distribuída entre os vários atributos. Valorizando principalmente os vários tipos de contrato para os dois *datasets*, o método de pagamento para o *dataset* da IBM e a posse de um serviço de suporte para o *dataset* do Kaggle.

Desta maneira ambos os *datasets* possuem características importantes para a previsão, sendo as mais relevantes os contratos de tipo mensal, o tempo que o cliente manteve o serviço, e mais especificamente no *dataset* da Kaggle a posse de um serviço de suporte, enquanto no *dataset* da IBM o atributo relevante é menos estável.

7. Validação Independente

Por motivos meramente exploratórios e de curiosidade. Decidiu-se treinar e avaliar os mesmos modelos utilizados no **Capítulo 6**, mas utilizando os dois *datasets* ao mesmo tempo, um para treino e o outro para teste. O intuito desta experiência seria testar a viabilidade do uso de dois *datasets* obtidos de contextos diferentes para fazerem a previsão um do outro, provando uma previsão generalizada.

Esta é uma tarefa complexa e desafiante, uma vez que os *datasets* provêm de contextos diferentes e principalmente *datasets* que não são balanceados, podendo acabar por representar padrões diferentes, o que dificultará o treino de um modelo genérico capaz de fazer previsões de forma correta.

7.1 Validação Independente entre *Datasets*

A experiência consiste no uso dos modelos de ML explorados anteriormente utilizando o *dataset* da IBM como treino e do *dataset* do Kaggle como validação, alterando posteriormente para o treino com o *dataset* do Kaggle e o teste com o dataste da IBM.

O processo de treino foi semelhante ao descrito no **Capítulo 6**, com a diferença de que os atributos selecionados, foram os atributos comuns a ambos os *datasets*, dentro daqueles que já tinham sido selecionados, de modo a serem compatíveis. Adicionalmente o método de validação cruzada não foi utilizado uma vez que foram utilizados conjuntos independentes.

O treino do *dataset* da IBM com os atributos de entrada "*Tenure*", "*MonthlyCharges*", "*ContractType*", "*InternetService*" e "*TechSupport*", produziu os resultados apresentados na **tabela 7**:

Tabela 7 — Avaliação dos Modelos ML com o Treino do *Dataset* da IBM

Algoritmos	*Precision*	*Accuracy*	*Recall*	*F1-Score*	*AUC*
RF	82.9%	58.2%	20.8%	51.4%	66.2%
SVM	99.4%	70%	40%	66.9%	76.8%
XGBoost	**100%**	**80.7%**	**61.4%**	**79.9%**	**77.7%**
DT	66.8%	57.2%	28.5%	53.3%	57.5%
LR	72.3%	64.9%	48.2%	63.9%	73.6%
NB	**100%**	69.5%	39.1%	66.4%	75.5%

O treino do *dataset* da Kaggle com os mesmos atributos da tabela anterior, produziu os seguintes resultados que se podem observar na **tabela 8**, quando testado no *dataset* da IBM:

Tabela 8 — Avaliação dos Modelo ML com o Treino do *Dataset* do Kaggle

Algoritmos	*Precision*	*Accuracy*	*Recall*	*F1-Score*	*AUC*
RF	56.9%	60%	97.7%	52.2	64.2
SVM	58.4%	61.8%	94.7%	56.4	74.6
XGBoost	56.9%	60%	**97.8%**	52.1	74.1
DT	56.9%	60%	**97.8%**	52.2	58.1
LR	**58.8%**	**62.3%**	94.6%	**57.4**	**78.5**
NB	58.4%	61.8%	94.6%	56.4	60.4

Numa primeira instância, os resultados da **tabela 7** aparentam não ser os melhores. Embora a precisão seja muito boa, a *recall* é insuficiente. Tendo em conta que a *recall* é uma métrica crítica foi possível interpretar que o *dataset* da IBM embora tenha o maior número de registos, não era adequado para fazer uma previsão generalizada de forma correta.

Já a **tabela 8**, apresenta resultados mais aceitáveis. A grande diferença do desempenho entre os dois testes reside na métrica *recall*. O *dataset* da IBM é o *dataset* mais completo, possuindo entre 7 a 8 vezes mais registos que o *dataset* do Kaggle, o que deveria contribuir de maneira positiva para o treino de modelos a partir deste *dataset*. Então levanta-se a questão sobre a razão dos resultados serem tão diferentes.

Uma das possíveis razões para a diferença nestes resultados é a ocorrência de *overffiting*. Sendo que o *dataset* da IBM possui um número considerável de registos, os modelos de ML poderão fixar padrões específicos que incluam ruído, sendo estes não generalizáveis e consequentemente afetando negativamente o desempenho dos modelos.

7.2 Validação Independente entre *Datasets* com *Undersampling*

De modo a verificar se a ocorrência de *overfitting* estaria a afetar os resultados, decidiu-se repetir a experiência, mas desta vez balanceando os *datasets* de maneira diferente.

Durante a primeira experiência a estratégia de balanceamento usada foi a mesma do treino e avaliação dos modelos no próprio *dataset.* De maneira a evitar a ocorrência de *overfitting*, nesta experiência foi utilizada a técnica de *random undersamping*, já que ao contrário do *oversampling*, o *undersampling* reduz o número de exemplos da classe maioritária, diminuindo assim a probabilidade de que os modelos sejam treinados em demasia, reduzindo o risco de o modelo fixar excessivamente padrões específicos da classe maioritária.

Desta maneira, repetiu-se a mesma experiência, mas utilizando unicamente a técnica *random undersampling*. O treino do *dataset* da IBM com os atributos descritos anteriormente, produziu os resultados da **tabela 9**:

Tabela 9 — Avaliação dos Modelos ML com o Treino do *Dataset* da IBM (com *undersampling*)

Algoritmos	*Precision*	*Accuracy*	*Recall*	*F1-Score*	*AUC*
RF	83.3%	62%	29.9%	57.6%	70.5%
SVM	**100%**	**74.8%**	49.6%	73.1%	**85.3%**
XGBoost	**100%**	67.9%	35.9%	64.3%	78.9%
DT	45.9%	47.0%	33.3%	46%	47%
LR	89%	74.4%	**55.6%**	**73.4%**	83.4%
NB	**100%**	**74.8%**	49.5%	73.1%	83%

Já o treino dos modelos através do *dataset* do Kaggle, com os mesmos atributos mencionados, obteve os resultados da **tabela 10**:

Tabela 10 — Avaliação dos Modelos ML com o Treino do *Dataset* do Kaggle (com *undersampling*)

Algoritmos	*Precision*	*Accuracy*	*Recall*	*F1-Score*	*AUC*
RF	54.4%	58%	**97.3%**	50.1%	72.6%
SVM	56%	60.1%	94.4%	54.7%	77%
XGBoost	55.8%	59.8%	94.5%	54.3%	73.2%
DT	54.4%	57.8%	**97.3%**	50%	57.8%
LR	**61.6%**	**67.3%**	92.2%	**65.2%**	**80%**
NB	56%	60.1%	94.4%	54.7%	60.4%

Numa análise simples, podemos ver, de uma maneira geral, que os resultados da *precision* e da *recall*, não sofreram alterações relevantes. Significando assim que mesmo ao reduzir o número de instâncias do *dataset* da IBM, ao ser utilizado para treino, não se evitou um possível *overfitting*.

Com a conclusão da segunda experiência, foi possível entender que os resultados negativos não dependeram puramente das técnicas de balanceamento usadas. Ao observarmos os resultados das duas experiências, é possível ver que o *dataset* do Kaggle possui maior facilidade em determinar os falsos negativos, uma vez que possui uma boa *recall*, mas não conseguiu prever corretamente os verdadeiros negativos, fruto da previsão de muitos falsos positivos, que influenciaram o resultado da *precision*. Assim sendo, podemos entender que os modelos treinados pelo *dataset* do Kaggle possuem uma tendência para prever casos positivos, independentemente do tipo de balanceamento utilizado.

Já no *dataset* da IBM, verifica-se o contrário. A *precision* é muito boa, o que significa que consegue distinguir bem os positivos dos falsos positivos. Por outro lado, os valores de *recall* não são os melhores, o que significa uma clara dificuldade em distinguir os casos negativos verdadeiros e os falsos. Por conseguinte, é possível entender que este modelo tem uma tendência para previsões negativas.

Cada *dataset* acabou por treinar modelos cuja tendência é a previsão da classe maioritária, sendo este resultado independente do tipo de balanceamento feito. Este efeito pode significar que não é possível utilizar um modelo de classificação generalizado para dados provenientes de contextos diferentes e que, aliás, poderão existir resultados demasiadamente otimistas noutros trabalhos relacionados, que não recorrem a validações independentes.

Apesar dos modelos treinados com o *dataset* da IBM, apresentarem *performances* que não apoiam uma boa capacidade de generalização, os modelos treinados através do *dataset* do Kaggle conseguiram resultados aceitáveis, demonstrando ser robusto e mais aproximado dos resultados pretendidos, sendo importante referir que o modelo *Logistic Regression* obteve um resultado de AUC de 80% juntamente com um valor alto de *recall* de 92.2%, o que estima um bom desempenho preditivo do modelo classificador.

8. Conclusões

O trabalho apresentado neste documento, teve como objetivo principal determinar a viabilidade da aplicação de ML para a previsão do *customer churn*, no setor das empresas de telecomunicações, sendo este um tema de estudo nos últimos anos. Identificar clientes com maior probabilidade de abandonar um serviço é fundamental para implementar medidas preventivas que garantam a sua retenção. Neste contexto, a aplicação de soluções baseadas em ML apresentam ser uma oportunidade valiosa para prever os clientes mais propensos de abandonar os serviços.

Numa primeira analise, foram investigados os tipos de aprendizagem ML existentes, concluindo que o problema abordado é um problema de classificação que requer um método de aprendizagem supervisionada. A relevância das várias métricas existentes e a sua respetiva interpretação foram fundamentais para a análise comparativa dos modelos de ML.

Durante o estudo das contribuições feitas previamente no contexto de ML para previsão do *customer churn*, foi possível identificar os algoritmos mais utilizados, a importância do pré-processamento dos datastes utilizados e os benefícios do balanceamento de *datasets* cujos atributos alvo estão em proporções desequilibradas.

Os *datasets* selecionados foram analisados, demonstrando algumas diferenças entre eles que podem ser explicadas por possíveis diferenças no contexto de cada um. Sob a perspetiva da ciência de dados, o *dataset* da IBM demonstrou ser mais robusto possuindo mais atributos úteis para a previsão e uma distribuição de períodos temporais de retenção mais uniforme.

Após o processo de pré-processamento, foram selecionados apenas os atributos mais relevantes através da matriz de correlação de cada *dataset*, de modo a beneficiar os resultados preditivos.

Na última fase, os *datasets* foram balanceados e ocorreu o treino e avaliação dos modelos de ML com base na técnica de validação cruzada. Os resultados obtidos permitiram a confirmação da viabilidade do uso de modelos ML para a previsão do *customer churn* no setor das telecomunicações. Entre os modelos treinados, o modelo baseado nas *Random Forest* apresentou os melhores resultados preditivos, confirmando a consistência dos resultados identificados no estado da arte.

Por último, é importante considerar a importância da qualidade dos *datasets* e a adequação ao contexto em que serão aplicados. Observou-se que os dois *datasets* estudados, apresentaram alguns padrões distintos, os quais poderão ter um impacto significativo no contexto deste trabalho, abrindo caminho para novos estudos.

8.1 Desafios e Trabalho Futuro

Apesar dos bons resultados com os modelos testados, seria interessante a disponibilização de um *dataset* com um maior número de atributos e registos, incluindo contexto externo como dados económicos, informações geográficas e tendências do mercado como concorrência.

A utilização das ferramentas SHAP e LIME, seriam uma adição positiva ao trabalho desenvolvido, permitindo interpretar e visualizar o impacto das variáveis de entrada nas previsões dos modelos.

O trabalho futuro, poderá também envolver a melhoria de treino de um modelo de ML, aplicando técnicas de otimização de hiperparâmetros, de modo a aperfeiçoar a capacidade de previsão do modelo.

O desenvolvimento de uma aplicação ou website que possa servir de interface para a utilização e interação com os resultados deste trabalho, seria um desafio interessante para o futuro.

9. Referências

[1] P. Senthan, R. Rathnayaka, B. Kuhaneswaran e B. Kumara, “Development of Churn Prediction Model using XGBoost - Telecommunication Industry in Sri Lanka,” *2021 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS),* pp. 1-7, 2021.

[2] O. Celik e U. O. Osmanoglu, “Comparing to techniques used in customer churn analysis,” *Journal of Multidisciplinary Developments,* pp. 30--38, 2019.

[3] I. Ullah, B. Raza, A. K. Malik, M. Imran, S. U. Islam e S. W. Kim, “A Churn Prediction Model Using Random Forest: Analysis of Machine Learning Techniques for Churn Prediction and Factor Identification in Telecom Sector,” *IEEE Access,* pp. 60134-60149, 2019.

[4] C. Stryker e E. Kavlakoglu, “What is artificial intelligence (AI)?,” IBM, 9 agosto 2024. [Online]. Available: https://www.ibm.com/think/topics/artificial-intelligence. [Acedido em 19 janeiro 2025].

[5] A. M. Turing, “COMPUTING MACHINERY AND INTELLIGENCE,” 1950.

[6] G. Press, “A Very Short History Of Artificial Intelligence (AI),” Forbes, 14 abril 2022. [Online]. Available: https://www.forbes.com/sites/gilpress/2016/12/30/a-very-short-history-of-artificial-intelligence-ai/. [Acedido em 19 janeiro 2025].

[7] “IA e machine learning,” Adobe Experience Cloud, [Online]. Available: https://business.adobe.com/pt/products/real-time-customer-data-platform/ai-vs-machine-learning.html. [Acedido em 19 janeiro 2025].

[8] S. Ben-David e S. Shalev-Shwartz, Understanding machine learning: From theory to algorithms, Cambridge university press, 2014.

[9] A. Géron, Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow, O'Reilly Media, Inc., 2022.

[10] “Quais são os tipos de aplicações de Inteligência Artificial mais comuns?,” Sprint PrograMaria Inteligência Artificial, 2 maio 2020. [Online]. Available: https://www.programaria.org/quais-sao-os-tipos-de-aplicacoes-de-inteligencia-artificial-mais-comuns/. [Acedido em 19 janeiro 2025].

[11] K. Terra, “Validação Cruzada de Modelos de Machine Learning,” 14 14 2021. [Online]. Available: https://medium.com/programacaodinamica/valida%C3%A7%C3%A3o-

cruzada-de-modelos-de-machine-learning-e89959826391. [Acedido em 19 janeiro 2025].

[12] "3.1. Cross-validation: evaluating estimator performance," sckit-learn, [Online]. Available: https://scikit-learn.org/stable/modules/cross_validation.html#cross-validation. [Acedido em 19 janeiro 2025].

[13] J. Murel e E. Kavlakoglu, "O que é uma matriz de confusão?," IBM, 19 janeiro 2024. [Online]. Available: https://www.ibm.com/br-pt/topics/confusion-matrix. [Acedido em 19 janeiro 2025].

[14] J. Saravanan, "How to Evaluate your Machine Learning Model.," 29 Maio 2021. [Online]. Available: https://medium.com/genai-io/how-to-evaluate-your-machine-learning-model-76a7671e9f2e. [Acedido em 19 janeiro 2025].

[15] "ROC Curve," DATAtab, [Online]. Available: https://datatab.net/tutorial/roc-curve. [Acedido em 19 janeiro 2025].

[16] V. Chang, K. Hall, Q. A. Xu, F. O. Amao, M. A. Ganatra e V. Benson, "Prediction of Customer Churn Behavior in the Telecommunication Industry Using Machine Learning Models," *Algorithms,* p. 231, 2024.

[17] S. E. Schaeffer e S. V. R. Sanchez, "Forecasting client retention—A machine-learning approach," *Journal of Retailing and Consumer Services,* p. 101918, 2020.

[18] P. Nagaraj, V. Muneeswaran, A. Dharanidharan, M. Aakash, K. Balananthanan e C. Rajkumar, "E-Commerce Customer Churn Prediction Scheme Based on Customer Behaviour Using Machine Learning," *2023 International Conference on Computer Communication and Informatics (ICCCI),* pp. 1--6, 2023.

[19] B. Prabadevi, R. Shalini e B. R. Kavitha, "Customer churning analysis using machine learning algorithms," *International Journal of Intelligent Networks,* pp. 145--154, 2023.

[20] A. Siddika, A. Faruque e A. K. M. Masum, "Comparative analysis of churn predictive models and factor identification in telecom industry," *2021 24th International Conference on Computer and Information Technology (ICCIT),* pp. 1--6, 2021.

[21] A. Raj e D. Vetrithangam, "Prediction of customer churn using resampling and ensemble classification algorithms on telecom dataset," *2023 11th International Conference on Emerging Trends in Engineering \& Technology-Signal and Information Processing (ICETET-SIP),* pp. 1--7,

2023.

[22] Kaggle, “Telco Customer Churn,” Kaggle, 2017. [Online]. Available: https://www.kaggle.com/datasets/blastchar/telco-customer-churn. [Acedido em 30 novembro 2024].

[23] J. J. R. Angelina, S. Subhashini, S. H. Baba, P. D. K. Reddy, P. S. K. Reddy e K. S. Khan, “A Machine Learning Model for Customer Churn Prediction using CatBoost Classifier,” *2023 7th International Conference on Intelligent Computing and Control Systems (ICICCS),* pp. 166--172, 2023.

[24] C. Acero-Charaña, E. Osco-Mamani e T. Ale-Nieto, “Model for Predicting Customer Desertion of Telephony Service using Machine Learning,” *International Journal of Advanced Computer Science and Applications,* 2021.

[25] R. Krishna, D. Jayanthi, D. S. Sam, K. Kavitha, N. K. Maurya e T. Benil, “Application of machine learning techniques for churn prediction in the telecom business,” *Results in Engineering,* p. 103165, 2024.

[26] B. R. Agasti e S. Satpathy, “Predicting customer churn in telecommunication sector using Naïve Bayes algorithm,” *Indonesian Journal of Electrical Engineering and Computer Science,* pp. 1610-1617, 2024.

[27] “What is Python? Executive Summary,” [Online]. Available: https://www.python.org/doc/essays/blurb/. [Acedido em 21 janeiro 2025].

[28] “Why Python for Machine Learning?,” [Online]. Available: https://pythonbasics.org/why-python-for-machine-learning/. [Acedido em 21 janeiro 2025].

[29] “What is Jupyter Notebook?,” [Online]. Available: https://domino.ai/data-science-dictionary/jupyter-notebook. [Acedido em 21 janeiro 2025].

[30] “About pandas,” [Online]. Available: https://pandas.pydata.org/about/. [Acedido em 21 janeiro 2025].

[31] “What is NumPy?,” [Online]. Available: https://numpy.org/doc/stable/user/whatisnumpy.html. [Acedido em 22 janeiro 2025].

[32] “What Is Matplotlib In Python? How to use it for plotting?,” [Online]. Available: https://www.activestate.com/resources/quick-reads/what-is-matplotlib-in-python-how-to-use-it-for-plotting/. [Acedido em 22 janeiro 2025].

[33] “Plot types,” [Online]. Available: https://matplotlib.org/stable/plot_types/index.

[34] “Learning Model Building in Scikit-learn,” 4 setembro 2024. [Online]. Available: https://www.geeksforgeeks.org/learning-model-building-scikit-learn-python-machine-learning-library/. [Acedido em 22 janeiro 2025].

[35] Kaggle, “ Customer Churn Prediction:Analysis,” Kaggle, setembro 2024. [Online]. Available: https://www.kaggle.com/datasets/abdullah0a/telecom-customer-churn-insights-for-analysis. [Acedido em 30 novembro 2024].

[36] “Pandas - get_dummies() method,” 3 dezembro 2024. [Online]. Available: https://www.geeksforgeeks.org/python-pandas-get_dummies-method/. [Acedido em 30 janeiro 2025].

[37] G. Novack, “Building a One Hot Encoding Layer with TensorFlow,” 7 junho 2020. [Online]. Available: https://towardsdatascience.com/building-a-one-hot-encoding-layer-with-tensorflow-f907d686bf39. [Acedido em 20 janeiro 2025].

[38] “StandardScaler,” [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html. [Acedido em 30 janeiro 2025].

[39] B. Keen, “Feature Scaling with scikit-learn,” 10 maio 2017. [Online]. Available: http://benalexkeen.com/feature-scaling-with-scikit-learn/. [Acedido em 20 janeiro 2025].

[40] V. Kumar, “Addressing Binary Classification over Class Imbalanced Clinical Datasets Using Computationally Intelligent Techniques,” *Healthcare,* vol. 10, julho 2022.

[41] M. Rutecki, “SMOTE and Tomek Links for imbalanced data,” [Online]. Available: https://www.kaggle.com/code/marcinrutecki/smote-and-tomek-links-for-imbalanced-data. [Acedido em 30 janeiro 2025].

[42] T. Shin, “Understanding Feature Importance in Machine Learning,” 7 novembro 2024. [Online]. Available: https://builtin.com/data-science/feature-importance. [Acedido em 30 janeiro 2025].

[43] “seaborn: statistical data visualization,” [Online]. Available: https://seaborn.pydata.org/. [Acedido em 22 janeiro 2025].

[44] “Matplotlib vs. seaborn vs. Plotly vs. MATLAB vs. ggplot2 vs. pandas,” [Online]. Available: https://ritza.co/articles/matplotlib-vs-seaborn-vs-plotly-vs-MATLAB-vs-ggplot2-vs-pandas/. [Acedido em 22 janeiro 2025].