INSTITUTO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DA PARAÍBA

COORDENAÇÃO DO CURSO SUPERIOR DE TECNOLOGIA EM SISTEMAS DE TELECOMUNICAÇÕES

DAMIÃO OTÁVIO DA CONCEIÇÃO

ESTUDO SOBRE RETENÇÃO DE CLIENTES EM OPERADORA DE TELECOMUNICAÇÕES UTILIZANDO INTELIGÊNCIA ARTIFICIAL

João Pessoa, março 2024

Damião Otávio da Conceição

ESTUDO SOBRE RETENÇÃO DE CLIENTES EM OPERADORA DE TELECOMUNICAÇÕES UTILIZANDO INTELIGÊNCIA ARTIFICIAL

Trabalho de Conclusão de Curso apresentado à Coordenação do Curso Superior de Tecnologia em Sistemas de Telecomunicações, do Instituto Federal da Paraíba - Campus João Pessoa, em cumprimento às exigências parciais para a obtenção do título de Tecnólogo em Telecomunicações.

Patric Lacouth da Silva
Orientador

João Pessoa, março 2024

Dados Internacionais de Catalogação na Publicação (CIP)
Biblioteca Nilo Peçanha do IFPB, *campus* João Pessoa

C744e Conceição, Damião Otávio da.

Estudo sobre retenção de clientes em operadora de telecomunicações utilizando inteligência artificial / Damião Otávio da Conceição. – 2023.

59 f. : il.

TCC (Graduação - Curso Superior de Tecnologia em Sistemas de Telecomunicações) - Instituto Federal de Educação da Paraíba / Coordenação do Curso Superior de Tecnologia em Sistemas de Telecomunicações, 2023.

Orientação : Profº D.r Patric Lacouth da Silva.

1.Inteligência artificial. 2. Telecomunicações. 3. Retenção de clientes. 4. Aprendizado de máquina. 5. Regressão logística. I. Título.

CDU 004.8:621.39(043)

Bibliotecária responsável: Lucrecia Camilo de Lima - CRB 15/132

Damião Otávio da Conceição

ESTUDO SOBRE RETENÇÃO DE CLIENTES EM OPERADORA DE TELECOMUNICAÇÕES UTILIZANDO INTELIGÊNCIA ARTIFICIAL

Trabalho de Conclusão de Curso apresentado à Coordenação do Curso Superior de Tecnologia em Sistemas de Telecomunicações, do Instituto Federal da Paraíba - Campus João Pessoa, em cumprimento às exigências parciais para a obtenção do título de Tecnólogo em Telecomunicações.

Aprovada em ____ / ____ / _____

Banca Examinadora

Documento assinado digitalmente
gov.br PATRIC LACOUTH DA SILVA
Data: 13/03/2024 09:16:17-0300
Verifique em https://validar.iti.gov.br

Profº D.r Patric Lacouth da Silva
Orientador

Documento assinado digitalmente
gov.br GUSTAVO ARAUJO CAVALCANTE
Data: 13/03/2024 09:21:10-0300
Verifique em https://validar.iti.gov.br

Profº D.r Gustavo Araújo Cavalcante
Examinador

Documento assinado digitalmente
gov.br LINCOLN MACHADO DE ARAUJO
Data: 17/03/2024 15:48:08-0300
Verifique em https://validar.iti.gov.br

Profº D.r Lincoln Machado de Araújo
Examinador

João Pessoa, março 2024

Dedico este trabalho, primeiramente, a Deus. À minha querida mãe, Eronilda Maria da Conceição, cuja força, determinação e fé são fontes inesgotáveis de inspiração. À minha amada esposa, Karolyne Thais Silva, dedico cada página deste trabalho como uma expressão do amor que compartilhamos, das superações que enfrentamos juntos e das conquistas que celebramos como equipe.

Com gratidão e amor,

Damião Otávio da Conceição

AGRADECIMENTOS

Quero expressar minha profunda gratidão a todos aqueles que foram fundamentais nesta jornada incrível de aprendizado. Primeiramente, agradeço a Deus por me guiar e me proporcionar força e perseverança ao longo dessa trajetória.

À minha mãe, minha fonte inesgotável de amor e apoio, não tenho palavras suficientes para agradecer por sua presença constante e por ser meu pilar nos momentos desafiadores.

À minha esposa, minha companheira de vida, agradeço por seu amor incondicional, compreensão e por ser minha maior incentivadora. Sua presença torna cada conquista ainda mais significativa.

Ao meu orientador, que com paciência e sabedoria, guiou-me pelos caminhos do conhecimento, sou grato por sua dedicação e orientação valiosa.

A todos os professores do curso de Tecnologia em Sistemas de Telecomunicações do IFPB, meu reconhecimento pelo compartilhamento generoso de conhecimento e pela inspiração constante. Cada um de vocês contribuiu significativamente para minha formação e crescimento profissional.

Agradeço a todos os amigos e colegas que tornaram essa jornada memorável, repleta de desafios e realizações. Este é um capítulo que levo com carinho em meu coração.

Obrigado a cada um de vocês por fazerem parte desta jornada extraordinária.

Com gratidão,
Damião Otávio da Conceição

“O Código da Inovação.”

Autor Elon Musk

RESUMO

Este trabalho tem como objetivo implementar modelos de aprendizado de máquina, utilizando o método da Regressão Logística e a Árvore de Decisão. Para atingir o propósito deste trabalho, os modelos desenvolvidos foram aplicados a uma base de dados de uma operadora de telecomunicações, obtida de um repositório público. Métricas de avaliação como acurácia, precisão, sensibilidade e matriz de confusão foram utilizadas para analisar os modelos treinados e ajudar na decisão dos melhores modelos. O modelo de Regressão Logística, apresentou maior precisão como métrica relevante. No entanto, a Árvore de Decisão demonstrou superioridade em acurácia e sensibilidade. Considerando a precisão como métrica mais relevante na classificação da solução proposta, dada sua relação direta com a classe 1 (abandono), o modelo de Regressão foi selecionado. Finalmente, uma tabela com os resultados de todas as métricas é apresentada, mostrando que os resultados obtidos com o uso da inteligência artificial na retenção de clientes são satisfatórios. O modelo desenvolvido neste trabalho pode ser adaptado para resolver problemas de evasão em outras empresas prestadoras de serviços, destacando também a possibilidade de intervenções nas áreas da saúde e educação.

Palavras-Chave: Inteligência Artificial. Retenção. Aprendizado de Máquina. Regressão.

ABSTRACT

This work aims to implement machine learning models using the Logistic Regression and Decision Tree methods. To achieve the purpose of this study, the developed models were applied to a telecommunications operator's dataset obtained from a public repository. Evaluation metrics such as accuracy, precision, sensitivity, and confusion matrix were used to analyze the trained models and assist in the selection of the best-performing models. The Logistic Regression model showed higher precision as a relevant metric. However, the Decision Tree demonstrated superiority in accuracy and sensitivity. Considering precision as the most relevant metric in classifying the proposed solution, given its direct relation to class 1 (churn), the Regression model was selected. Finally, a table with the results of all metrics is presented, showing that the use of artificial intelligence in customer retention yields satisfactory results. The model developed in this study can be adapted to address churn issues in other service-providing companies, highlighting the potential for interventions in the fields of health and education as well.

Keywords: Artificial Intelligence. Retention. Machine Learning. Regression."

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURA E SIGLAS

ARPU	Receita média por usuário
IA	Inteligência artificial
FP	Falso Positivo
FN	Falso Negativo
GINI	Índice de impureza
VP	Verdadeiros Positivos
VN	Verdadeiro Negativo
α	Combinação linear das variáveis e seus coeficientes

SUMÁRIO

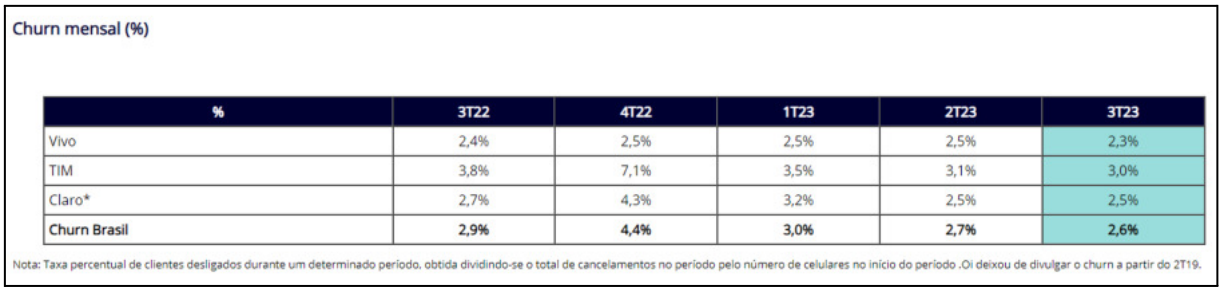
1 INTRODUÇÃO

No atual panorama altamente competitivo das operadoras de telecomunicações, a retenção de clientes representa um desafio crucial para garantir a estabilidade e o crescimento sustentável das empresas do setor. A rápida evolução tecnológica, aliada às crescentes expectativas dos consumidores, demanda abordagens inovadoras para antecipar e atender às necessidades individuais dos clientes, visando a fidelização. Nesse contexto, a aplicação de métodos de aprendizado de máquina supervisionado, como Regressão Logística e Árvore de Decisão, emerge como uma estratégia promissora para otimizar a gestão do relacionamento com o cliente (SERPA, 2023).

Este estudo propõe uma investigação abrangente sobre a retenção de clientes em operadoras de telecomunicações, com ênfase na implementação de modelos preditivos baseados em inteligência artificial. Utilizando dados obtidos do https://www.*kaggle*.com/, exploramos a eficácia de dois métodos de aprendizado supervisionado, Regressão Logística e Árvore de Decisão, na predição de churn (cancelamento do serviço).

A retenção de clientes impacta diretamente a receita das operadoras de telecomunicações no Brasil. Como pode ser visto nos dados da Figura 1, a média de *churn* pelas algumas das principais operadoras do Brasil. No ano de 2022 quarto trimestre (‘4T22’) foi de 4,4%, representando a maior taxa em comparação com os outros trimestres da pesquisa. Ao analisarmos o mesmo período na Figura 2, constatamos uma menor arrecadação (TELECO, 2023). Dessa forma, observamos que, na Figura 1, o menor valor da taxa de cancelamento está associado a uma arrecadação maior.

Figura 1 - Cancelamentos de Assinaturas no Brasil



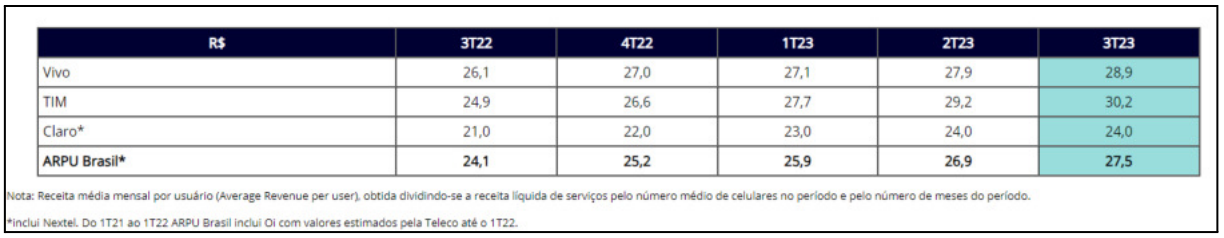
Churn mensal (%)

%	3T22	4T22	1T23	2T23	3T23
Vivo	2,4%	2,5%	2,5%	2,5%	2,3%
TIM	3,8%	7,1%	3,5%	3,1%	3,0%
Claro*	2,7%	4,3%	3,2%	2,5%	2,5%
Churn Brasil	**2,9%**	**4,4%**	**3,0%**	**2,7%**	**2,6%**

Nota: Taxa percentual de clientes desligados durante um determinado período, obtida dividindo-se o total de cancelamentos no período pelo número de celulares no início do período .Oi deixou de divulgar o churn a partir do 2T19.

Fonte: adaptado de https://www.teleco.com.br/ (2023)

Figura 2 - Receita Média Mensal pelas Operadoras no Brasil



R$	3T22	4T22	1T23	2T23	3T23
Vivo	26,1	27,0	27,1	27,9	28,9
TIM	24,9	26,6	27,7	29,2	30,2
Claro*	21,0	22,0	23,0	24,0	24,0
ARPU Brasil*	**24,1**	**25,2**	**25,9**	**26,9**	**27,5**

Nota: Receita média mensal por usuário (Average Revenue per user), obtida dividindo-se a receita líquida de serviços pelo número médio de celulares no período e pelo número de meses do período.

*Inclui Nextel. Do 1T21 ao 1T22 ARPU Brasil inclui Oi com valores estimados pela Teleco até o 1T22.

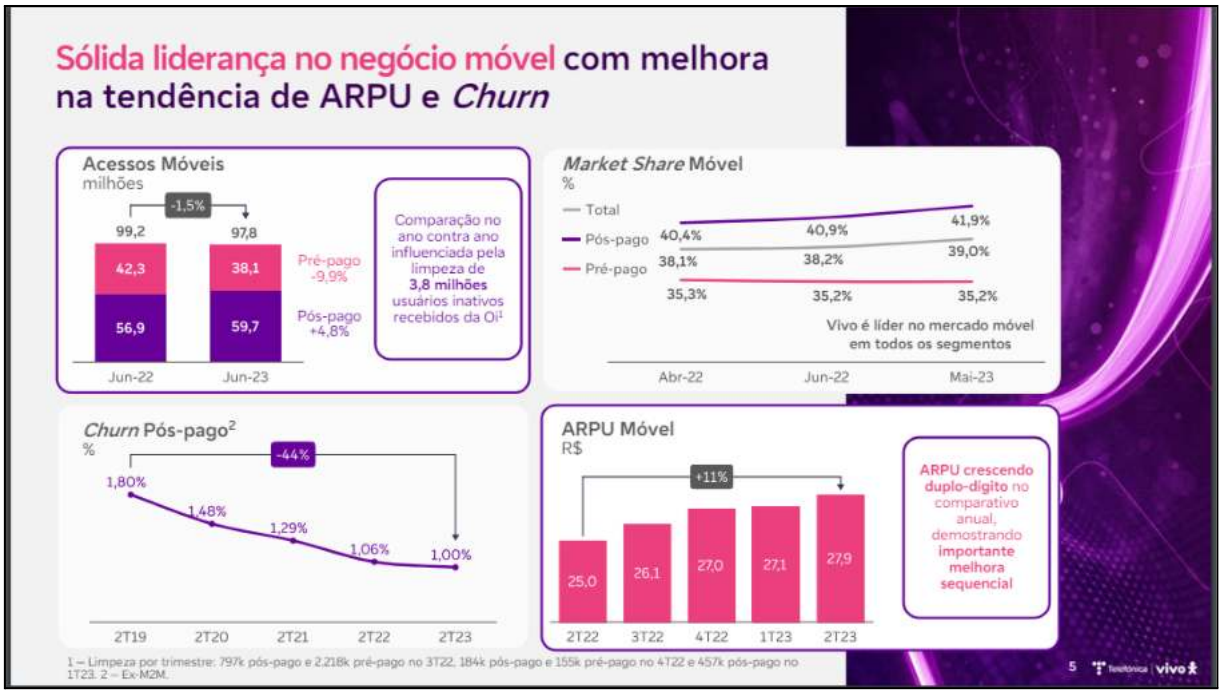
Fonte: adaptado de https://www.teleco.com.br/ (2023)

Para compreender a importância da retenção para as operadoras de telecomunicações, analisaremos os dados disponibilizados pela empresa Vivo na Figura 3, onde podemos observar a relação entre o *churn* e o ARPU (receita média por usuário ou *average revenue per user*).

O churn representa a relação entre o número de cancelamentos e a média de clientes no período. Em um mercado competitivo, esse dado é crucial para medir a fidelidade dos clientes. Já o ARPU é a receita de serviços, líquida de impostos, dividida pelo número médio de usuários. Essa informação é apresentada com a subdivisão para clientes pré-pagos e clientes pós-pagos, sendo o valor total do ARPU conhecido como ARPU *blended*.

Ao analisarmos o gráfico do *churn* versus ARPU na Figura 3, podemos constatar como a retenção efetivamente impacta as operadoras. Quanto menor e mais estável for essa taxa, melhor será o ganho e a estabilidade financeira da empresa. Isso ocorre porque, à medida que a retenção da organização é baixa, sua instabilidade e saúde financeira também são reduzidas.

Figura 3 - Churn vs URPU



Fonte: adaptado de https://ri.telefonica.com.br/ (2023).

A média ideal para taxa de cancelamento do serviço é de 0,5% ao mês, sem ultrapassar 7% ao ano (WISNIEWSKI, 2020). Uma taxa alta alerta para a perda de receita, indicando a necessidade de medidas corretivas (CORDOVEZ, 2023). Por isso essa medida é tão importante para as empresas prestadoras de serviços.

Ao finalizar este estudo, espera-se que os resultados gerados não apenas enriqueçam o conhecimento acadêmico sobre a interseção entre aprendizado de máquina e retenção de clientes, mas também forneçam orientações práticas para operadoras brasileiras, capacitando-as a desenvolver estratégias mais eficazes e adaptadas ao contexto específico do mercado nacional.

1.1 Objetivos

Os objetivos para executar o estudo estão classificados em objetivo geral e objetivos específicos.

1.2 Objetivo Geral

- Entender como o uso da inteligência artificial pode ajudar empresas prestadoras de serviços por assinatura na retenção de clientes .

1.3 Objetivos Específicos

- Desenvolver um modelo de Aprendizado de Máquina para prever o abandono de clientes em uma empresa de telecomunicações, utilizando uma base de dados.
- Aplicar mais de um método de predição para comparar os resultados obtidos.
- Escolher o melhor método para solução do problema.

1.4 Organização do Trabalho

Com o objetivo de alcançar os propósitos delineados neste trabalho, a organização adotou a seguinte estrutura: composto por cinco capítulos. No Capítulo 2, realizou-se uma revisão bibliográfica abrangente na qual foram analisados trabalhos e estudos relevantes sobre o tema. Esta revisão teve como propósito proporcionar uma compreensão aprofundada de como a inteligência artificial é aplicada na área em questão. No Capítulo 3, apresentam-se detalhes sobre a metodologia utilizada na seleção dos estudos revisados, incluindo critérios de inclusão e exclusão, além das fontes consultadas, com o intuito de oferecer transparência ao processo. No Capítulo 4, são apresentadas as etapas de desenvolvimento, os modelos desenvolvidos foram aplicados aos dados, e os resultados de cada modelo foram comparados. No Capítulo 5, são dispostas as conclusões finais do trabalho.

2 REFERENCIAL TEÓRICO

Este segmento tem como objetivo explorar certos princípios discutidos na literatura sobre Retenção de Clientes, Inteligência artificial, Fundamentos de Aprendizado de Máquina .

2.1 Retenção de Clientes

O foco primordial em qualquer organização é o relacionamento com os clientes, conforme destacado na missão empresarial (FERREIRA, 2012). Reter clientes é crucial, dada a significativa diferença de custos entre conquistar novos e manter os atuais (Gnoatto, 2023). Para empresas de serviços, uma orientação forte para o cliente resulta em ligações duradouras, influenciando a lealdade e proporcionando lucros sustentáveis (FERREIRA, 2012).

A retenção está intrinsecamente ligada à satisfação, mantendo a preferência do cliente ao longo do tempo (MILAN, TONI, 2012). Em serviços, a retenção concentra-se em satisfazer clientes existentes, visando conquistá-los para o longo prazo (ECKERT, MILAN, MECCA, NUNES, 2013).

A qualidade do serviço influencia diretamente a permanência do cliente, estabelecendo confiança desde as primeiras experiências (Gnoatto, 2023). Num mercado competitivo e dinâmico, torna-se desafiador reter clientes e evitar o abandono, especialmente com o avanço tecnológico e maior acesso à informação (SERPA, 2023).

2.2 Rotatividade de Clientes

O custo de manter um cliente existente é mais vantajoso do que o custo de atrair um novo cliente (MILAN, TONI, 2012). Portanto, a retenção está diretamente associada à taxa de rotatividade de clientes. Diversos autores oferecem definições pertinentes a esse conceito:

A taxa de rotatividade, também conhecida como desistência ou cancelamento de clientes, ocorre quando um cliente encerra seu relacionamento com uma empresa, possivelmente migrando para uma organização concorrente (SILVEIRA, 2022).

Esta métrica, conforme destacado por SERPA (2023), desempenha o papel crucial de identificar a propensão dos clientes a encerrarem sua relação com a instituição.

GNOATTO (2023) complementa que a rotatividade envolve a evasão de clientes e a previsão daqueles que apresentam alguma possibilidade de abandonar o serviço, tornando-se um elemento fundamental na gestão de clientes.

2.3 *Churn*

Churn termo essencial no empreendedorismo, especialmente para serviços por assinatura, é definido como '*churn*' ou '*churn rate*,' também conhecido como Taxa de *Churn*, sendo a métrica para compreender a perda de clientes pagantes (ABEL, 2017). Essa métrica é crucial para entender a curva de perda de clientes, permitindo à organização tomar medidas para evitar e recuperar clientes (CORDOVEZ, 2023).

CORDOVEZ (2023) define '*churn*' como a métrica que mostra o número de clientes que cancelaram um serviço em um período específico. WISNIEWSKI (2020) a caracteriza como a taxa de rotatividade de clientes em uma empresa.

GOMES e BRAGA (2017, p.73) descrevem 'churn' como a 'quantidade de clientes ou assinantes que cortam laços com serviços ou empresas durante determinado período de tempo.' Essa métrica é crucial para analisar a relação com o cliente, capacidade de retenção e evasão (ECKERT, MILAN, MECCA, NUNES, 2013).

O cálculo da *Taxa de Churn* é simples: total de cancelamentos no período dividido pelo total de clientes no período escolhido.

- Definir um período, dividir o número de clientes perdidos até o fim desse período pelo total de clientes que tinha no início do período.

$$Taxa\ Churn = \frac{Total\ de\ cancelamentos\ no\ período}{Total\ de\ clientes\ no\ período\ escolhido} \quad (1)$$

Exemplo:

Período = 30 dias

Total de clientes no período = 500

Total de cancelamentos no período = 15

$$Taxa\ Churn = \frac{15}{500} = 0{,}03\ \%$$

A recomendação é manter a taxa de rotatividade (churn) em média 0,5% ao mês, sem ultrapassar 7% ao ano (WISNIEWSKI, 2020). Uma taxa alta alerta para a perda de receita, indicando a necessidade de medidas corretivas (CORDOVEZ, 2023).

"As empresas têm percebido que atrair e reter clientes depende fundamentalmente de fornecer a eles uma experiência personalizada. Por essa razão, mais empresas estão buscando soluções de Big Data e uma visão analítica aprimorada para entender seus clientes de forma mais profunda" (GOMES, BRAGA, 2017, p.75)."

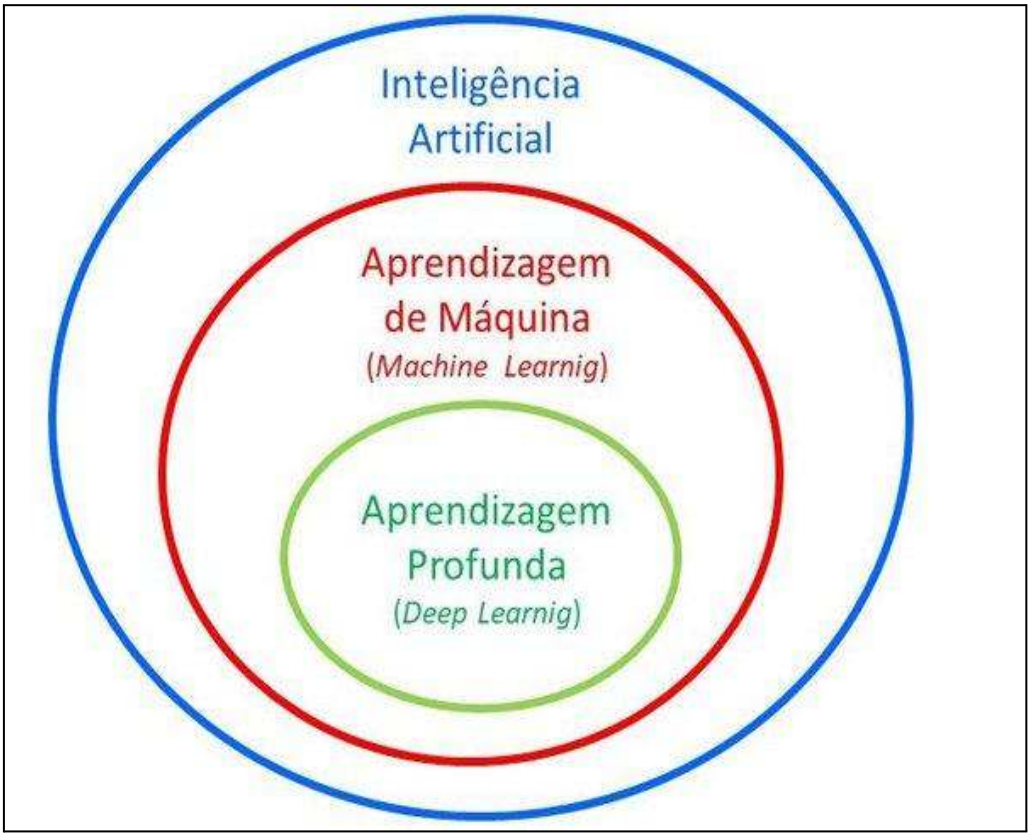
2.4 Inteligência Artificial

Quando buscamos uma definição na literatura, não encontramos algo exato que define a Inteligência Artificial (IA), muito por conta da indefinição do próprio termo "inteligência". No entanto, podemos entender que se trata da capacidade de fazer máquinas interagirem ou até mesmo tomarem decisões com base em dados fornecidos a elas. Isso ocorre graças a complexos algoritmos que viabilizam este conceito (GANOALTO, 2023).

Mas podemos definir a Inteligência Artificial (IA) como a subárea da Ciência da Computação responsável por pesquisar e propor a elaboração de dispositivos computacionais capazes de simular aspectos do intelecto humano, como a capacidade de raciocinar, perceber, tomar decisões e resolver problemas (RÔMULO, 2013, p. 1).

Segundo DAMACENO, S. S. e VASCONCELOS, R. O. (2018), a Inteligência Artificial (IA) pode ser subdividida em camadas ou em partes que a compõem, conforme ilustrado na Figura 4.

Figura 4 - Divisões da Inteligência Artificial.



Fonte: Modificado de Taurion (2019, p.4).

- *Machine Learning*, ou aprendizado de máquina, é o processo de aprendizado contínuo de uma máquina. Consiste basicamente em fornecer dados de entrada, permitindo assim que a máquina aprenda com esses dados e elabore saídas que atendam à situação-problema. Algoritmos de *Machine Learning* são estruturados com equações pré-definidas para organizar e processar os dados (DAMACENO, S. S., & VASCONCELOS, R. O. 2018)."

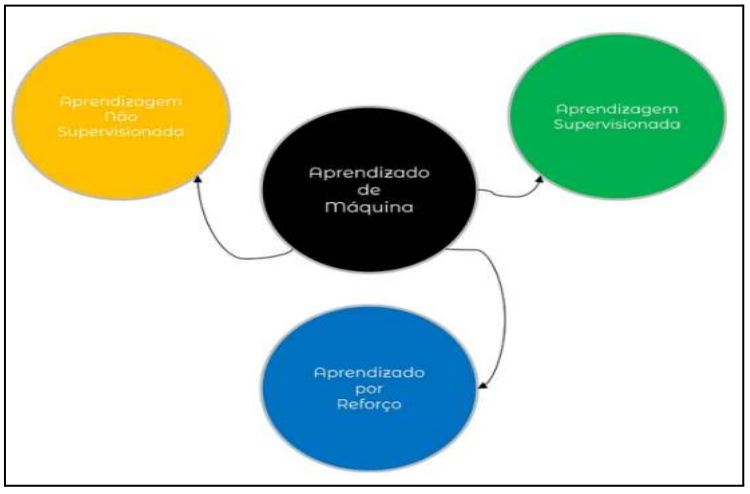
- *Deep Learning* (Aprendizado Profundo) é um tipo de aprendizado de máquina que executa tarefas mais complexas, como reconhecimento de voz, identificação de imagens e previsões. Essa camada da inteligência artificial estabelece parâmetros fundamentais sobre esses dados e treina o computador para aprender autonomamente, utilizando várias camadas de processamento no reconhecimento de padrões (DAMACENO, S. S., & VASCONCELOS, R. O. 2018).

Desta forma, conforme ilustrado na Figura 4, na área de inteligência artificial, essas duas camadas de aprendizado de máquina e aprendizado profundo andam lado a lado, podendo ser categorizadas em esferas, com a inteligência artificial abrangendo ambas as tecnologias. A camada da Inteligência Artificial (IA), na qual este estudo está focado, é a área de *Machine Learning* (Aprendizado de Máquina). Nessa camada da inteligência artificial, por meio de algoritmos, a IA é capaz de aprender, executar tarefas e tomar decisões de forma autônoma. Através do treinamento, esses sistemas computacionais adquirem conhecimento (DAMACENO, S. S., & VASCONCELOS, R. O. 2018).

2.5 Tipos de Aprendizagem de Máquina.

O aprendizado de máquina consiste em diversas metodologias e aplicações, sendo dividido em diferentes áreas, variando de acordo com os dados de estudo. Uma maneira de compreender isso é realizando a distinção entre os tipos de aprendizado e sua aplicabilidade em diferentes situações e objetivos. Como ilustrado na Figura 5.

Figura 5 - Tipos de Aprendizagem de Máquina.



Fonte: Modificado de Medium (2022, p.6).

Aprendizado supervisionado

Nesse método, afirmamos que o aprendizado ocorre de maneira supervisionada quando temos acesso a uma amostra do problema contendo informações sobre um determinado objeto de estudo, denominadas dados de entrada, juntamente com o resultado

obtido a partir dessas informações, chamado de dado de saída (IGNACIO, 2021). Isso ocorre quando há uma variável de resposta estabelecida como objetivo.

O processo de treinamento do algoritmo funciona da seguinte forma: recebe os valores de entrada, nos quais podemos ajustar os parâmetros para corrigir erros na saída, dividindo os dados de entrada em dois conjuntos: dados de treino e dados de teste. O algoritmo deve ser executado utilizando o conjunto de dados de treino com a finalidade de obter os parâmetros do modelo estudado. Após esta etapa, para testar a capacidade de predição, ele é executado nos dados de teste (IGNACIO, 2021). Deste modo, verificamos a tomada de decisão do algoritmo, testamos sua acurácia e precisão em relação aos dados de teste.

O aprendizado supervisionado é geralmente aplicado para solucionar problemas em casos de mineração de dados, podendo ser dividido em dois tipos: classificação e regressão.

- Na regressão, nesse contexto, a variável de saída é um valor real, e a variável de entrada é mapeada para alguma função contínua (Medium, 2022, p.7). Nesse caso, o método de regressão utiliza algoritmos para entender a relação entre variáveis dependentes e independentes. Os modelos de regressão são úteis para prever valores numéricos, como projeções de receita de vendas para um determinado negócio. Alguns algoritmos de regressão populares incluem regressão linear, regressão logística e regressão polinomial (JAIME, 2022).

- Na classificação, em problemas desse tipo, as variáveis de interesse são categóricas e discretas. Dessa forma, assumem valores binários (0 ou 1) representando sim ou não. Contudo, também existe a classificação multiclasse, na qual os dados podem ter mais do que duas categorias (Medium, 2022).

Aprendizado não supervisionado

Contrariamente ao aprendizado supervisionado, o aprendizado não supervisionado ocorre quando não temos acesso a um conjunto de amostras que contenha a relação entre os pontos de entrada e saída. O objetivo do aprendizado não supervisionado é extrair padrões dos dados disponíveis (IGNACIO, 2021). Nesse caso, o algoritmo é executado sem orientação sobre os dados, realizando agrupamentos para encontrar padrões naturais (Medium, 2022). Esse tipo de aprendizado é utilizado em tarefas como agrupamento, associação e redução de dimensionalidade:

- O Agrupamento visa unir dados sem informações prévias com base em suas semelhanças ou diferenças.
- Associação é outro método de aprendizado não supervisionado que utiliza regras para identificar relacionamentos entre variáveis em um conjunto de dados específico.
- A redução de dimensionalidade é uma técnica de aprendizado aplicada quando o conjunto de dados é extenso. Essa técnica reduz o número de entradas de dados para torná-lo mais gerenciável, preservando a integridade dos dados mesmo com a diminuição das dimensões.

Aprendizado por reforço

Neste tipo de aprendizado, ele é baseado no processo de reforço, consistindo em ensinar um agente a interagir com um ambiente por meio de um conjunto finito de ações para que ele atinja um objetivo definido. A parte do reforço se dá porque o processo de treinamento envolve a aplicação de um valor de punição para cada ação realizada pelo agente, de modo que o agente busca encontrar ações que minimizem a punição recebida (IGNACIO, 2021).

Um bom exemplo de aplicação desse tipo de aprendizado é em jogos de videogame. Em termos gerais, assemelha-se muito à forma como os humanos aprendem por tentativa e erro. No entanto, a aprendizagem por reforço é conceitualmente similar, mas ocorre por meio da realização de ações com a minimização do erro (Medium, 2022).

3 METODOLOGIA

Um problema de classificação binária consiste em determinar, a partir de um conjunto de dados, um dado que assume dois valores possíveis (0 ou 1), como "sim" ou "não". Mesmo que o dado não seja inicialmente numérico, ele é frequentemente transformado, uma vez que envolve duas classes distintas. Um exemplo prático disso é a classificação de transações com cartão de crédito para determinar se são ou não fraudulentas, prever se um cliente pagará ou não um empréstimo, ou antecipar o churn de clientes.

Nesse tipo de modelo, a resposta geralmente varia entre esses dois valores, associando a ocorrência do evento ao valor 1 e o valor 0 à não ocorrência do evento. Métodos estatísticos e de aprendizado de máquina amplamente utilizados para predizer essas classificações incluem regressão logística, árvores de decisão, florestas aleatórias (*Random Forest*), KNN (K-vizinhos mais próximos), *Support Vector Machine* (SVM, ou Máquina de Vetores de Suporte) e redes neurais artificiais. Esses métodos são amplamente empregados em diversas áreas, conforme destacado por SERPA (2023).

3.1 Regressão Logística e Árvores de Decisão

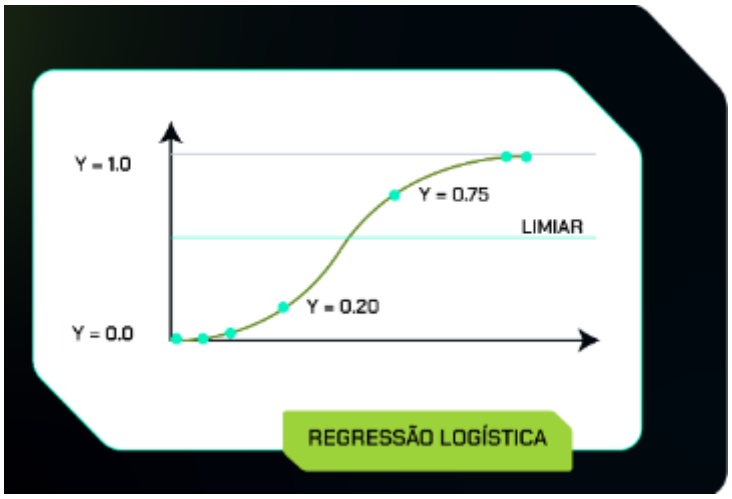
Regressão logística

A Regressão Logística é um método estatístico utilizado para prever classes binárias, em que a variável alvo apresenta natureza dicotômica, indicando a presença de apenas duas possíveis classes. Esta técnica é aplicável a diversas situações, tais como detecção de câncer, identificação de fraudes em compras, previsão da probabilidade de um cliente honrar ou não um empréstimo, entre outras aplicações. A Regressão Logística é empregada na modelagem desses cenários ao calcular a probabilidade de ocorrência de um determinado evento (GONZALES, 2018).

Apesar do nome estar associado à regressão, essa técnica estatística é amplamente utilizada para modelar a relação entre uma variável binária dependente e uma ou mais variáveis independentes. Seu algoritmo é empregado na resolução de problemas de

classificação, mas seu funcionamento guarda grande semelhança com o algoritmo de Regressão Linear. A Regressão Logística é utilizada para estimar valores discretos de classes binárias, tais como 0/1, sim/não, verdadeiro/falso, com base em um conjunto de variáveis independentes. Como se pode observar na Figura 6.

Figura 6 - Gráfico da Regressão Logística.



Fonte: Modificado de Alura (2024, p.20).

Internamente, a Regressão Logística calcula a probabilidade de ocorrência de um evento, ajustando os dados a uma função *logit*, a qual mapeia a saída em valores entre 0 e 1 (ESCOVEDO, 2020).

A variável dependente Y na regressão logística é frequentemente binária. Portanto, segue a distribuição de Bernoulli, tendo uma probabilidade desconhecida p (ESCOVEDO, 2020).

Na área de teoria das probabilidades e estatística, a distribuição de Bernoulli, nomeada em homenagem ao cientista suíço Jakob Bernoulli, é uma distribuição discreta no espaço amostral {0, 1}. Ela assume o valor 1 com a probabilidade de sucesso, e o valor 0 com a probabilidade de falha (DISTRIBUIÇÃO DE BERNOULLI. In: WIKIPÉDIA, 2023).

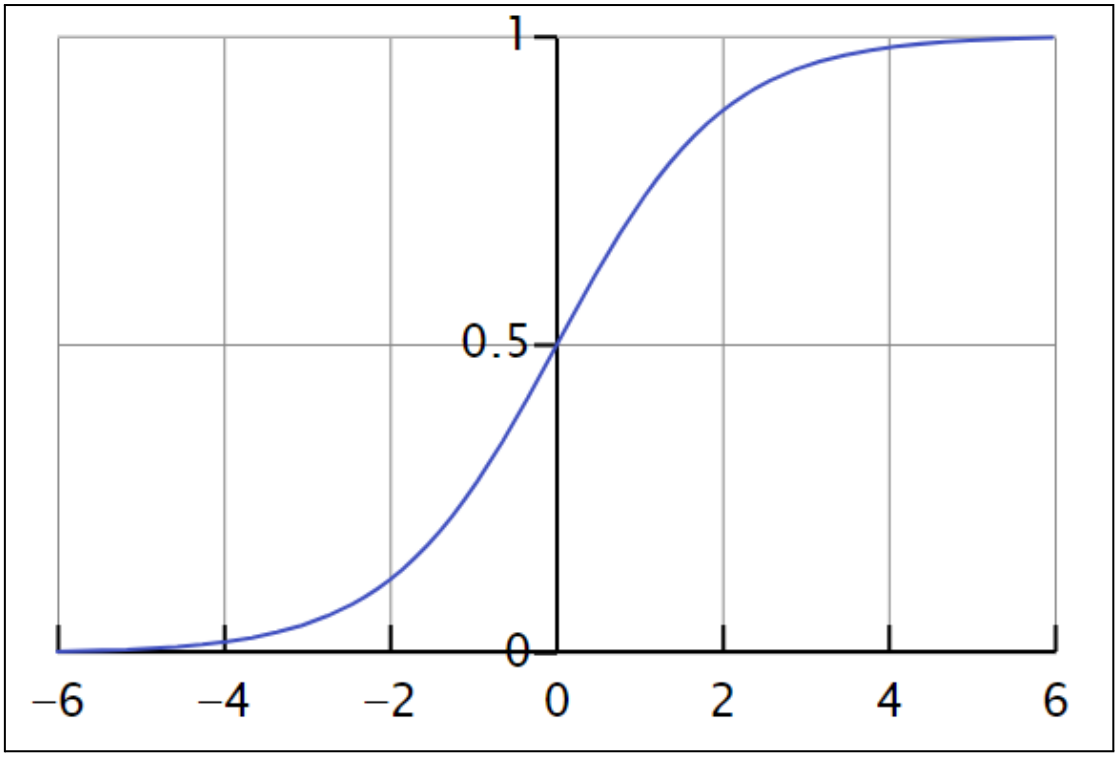
De forma que Y é igual a 1 se ocorrer sucesso, 0 se ocorrer fracasso. A probabilidade de sucesso é $0 \leq p \leq 1$ e a probabilidade de fracasso q = 1 - p. Como a variável dependente segue a distribuição de Bernoulli é preciso que as variáveis independentes x também siga essa mesma distribuição, essa relação é chamada de Logit. Entretanto na regressão logística não conhecemos a probabilidade p como é padrão nos problemas de distribuição Bernoulli. Deste modo, o modelo logístico serve para estimar a probabilidade de p.

$$logit^{-1}(\alpha) = \frac{1}{1+e^{-\alpha}} = \frac{e^{\alpha}}{1+e^{\alpha}} \quad (2)$$

Adaptado de (ESCOVEDO, 2020)

No modelo de regressão logística, α, será a combinação linear das variáveis e seus coeficientes. De modo que a função logit retorna a probabilidade da variável dependente Y ser igual a igual a 1 .

Figura 7 - Gráfico da função Logít.



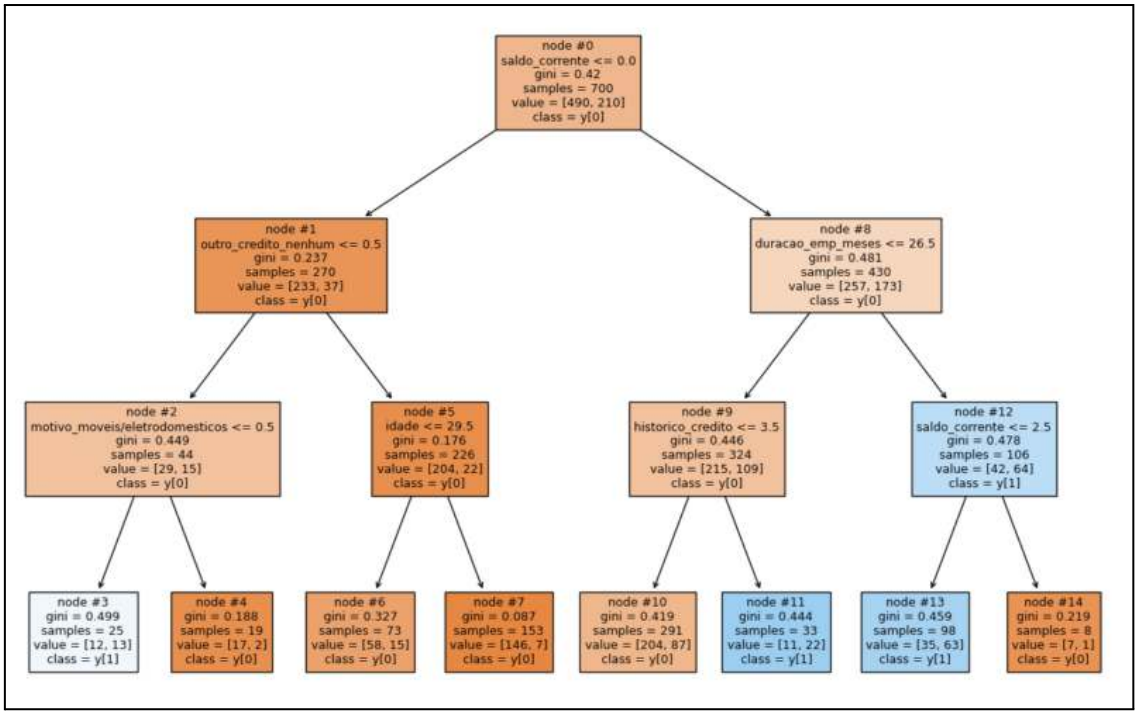
Fonte: Modificado de Wikipédia (2023, p.1).

Como se pode observar na Figura 7, a função logística, por se tratar de uma probabilidade, apresenta resultados sempre limitados entre 0 e 1. A curva assume a forma de um "S", indo de menos infinito até mais infinito (Rodrigues, 2020). Dentro do modelo, é estimada uma média entre os limites 0 e 1. Dependendo do valor, se estiver acima dessa média, é considerado como 1; se estiver abaixo, é considerado como 0.

Árvore de decisão

O método de árvore de decisão é empregado para resolver problemas de classificação. Este modelo é fundamentado em regras de hierarquia de separação, representadas em um desenho no formato de árvore, com ramificações partindo do nó principal, conforme ilustrado na Figura 8.

Figura 8 - Árvore de Decisão



Fonte: Modificado de Brains.dev (2023, p.1)

Uma árvore de decisão é um mapa dos possíveis resultados de uma série de escolhas relacionadas. Geralmente, ela começa com um único nó que se divide em possíveis resultados, conforme ilustrado na Figura 8. Cada um desses resultados leva a nós adicionais, que se ramificam em outras possibilidades, assim criando o modelo de árvore. Dessa forma, para cada saída da árvore, o algoritmo escolhe o melhor caminho.

Para a definição dos nós e ramos que compõem a árvore de decisão, é calculado o parâmetro GINI, que representa o índice de impureza e é dado pela fórmula:

$$G = \sum_{k=1}^{k} pk(1 - pk) \quad (3)$$

Fonte: Adaptado de MARIN (2012).

O parâmetro GINI calcula a impureza de um nó. Quanto mais puro o nó, mais transações da mesma classe ele possui. Na fórmula, os parâmetros pk representam as probabilidades de cada classe no nó, e k é o número total de classes. É importante observar que o algoritmo realiza esse cálculo do índice GINI separadamente para cada ramo da árvore e, em seguida, calcula o índice geral da árvore. O valor do índice GINI indica que, se for 0%, o nó é totalmente puro, enquanto 100% indica impureza máxima. O ideal é que o valor esteja o mais próximo possível de 0. Com base nesses cálculos, a estrutura da árvore é definida, colocando no topo as variáveis mais relacionadas à variável resposta (SERPA, 2023).

3.2 Etapas da Análise e Métricas de Avaliação

Para problemas de classificação que utilizam aprendizado de máquina supervisionado, onde temos informações predefinidas sobre a saída esperada, é comum que o conjunto de dados seja dividido em duas partes: dados de treino (usados para ajustar e treinar o modelo) e dados de teste (parcelas dos dados usadas para testar e validar o modelo com correção na predição). Nos modelos utilizados neste trabalho, considerou-se a proporção de 75/25, de modo que 75% dos dados são destinados ao treino do modelo e 25% para teste.

O modelo de Regressão Logística foi ajustado inicialmente utilizando apenas algumas variáveis da base de dados. Com base nos resultados, foram acrescentadas as demais variáveis para obter uma melhor otimização. Posteriormente, utilizaram-se as mesmas técnicas de modelagem com o modelo de Árvore de Decisão com o objetivo de comparar os resultados obtidos pelos dois modelos e verificar qual se adaptou melhor ao problema.

Outra etapa muito importante deste trabalho foi a utilização de métricas para comparar a eficiência e capacidade preditiva dos modelos. Foram consideradas 4 dessas métricas: acurácia, precisão e sensibilidade.

Acurácia

A métrica de acurácia é definida como a distância total entre os valores estimados e os valores reais; ou seja, ela mede a assertividade do modelo com base nos valores preditos. Ela fornece a porcentagem de classificações corretas em relação ao total de predições, conforme demonstrado na equação (3).

$$\text{Acurácia} = \frac{VP+VN}{total\ de\ predições} \quad (4)$$

De acordo com a equação (3), a acurácia será igual à soma dos verdadeiros positivos (VP) e dos verdadeiros negativos (VN), dividida pelo total de resultados. Essa soma inclui os verdadeiros positivos mais os falsos positivos, somados aos verdadeiros negativos mais os falsos negativos. O resultado representa o total de valores classificados e previstos corretamente.

Precisão

Precisão é outra métrica muito importante, pois nos indica quantos foram classificados como verdadeiros positivos (VP) pelo modelo, conforme a equação (4).

$$\text{Precisão} = \frac{VP}{VP + FP} \quad (5)$$

Essa métrica avalia a relação entre o total de classificações corretas de churns (verdadeiros positivos) dentro do total de classificações positivas (churns identificados) feitas pelo modelo. Ou seja, representa a precisão com que o modelo acertou o resultado.

Sensibilidade

A sensibilidade mede a porcentagem de verdadeiros positivos (VP) que o modelo classifica corretamente, demonstrando quão eficaz ele é em identificar resultados verdadeiros positivos (VP), conforme a equação (5).

$$\text{Sensibilidade} = \frac{VP}{VP + FN} \quad (6)$$

Essa métrica enfatiza mais os erros conhecidos como falsos negativos (FN), que ocorrem quando o modelo classifica erroneamente um churn observado como não churn (falsos negativos) - verdadeiros positivos.

3.3 Software Utilizado

Para a construção do modelo de predição, foi utilizada uma ferramenta gratuita do Google chamada Google Colaboratory. Este é um ambiente de notebooks Jupyter que não requer nenhuma configuração, e é de livre acesso, bastando ter uma conta de e-mail no Google. Outro ponto positivo para sua escolha foi que ele é executado em uma máquina na nuvem do próprio Google, onde é possível salvar e compartilhar, e já vem com bibliotecas pré-instaladas, sendo assim uma ferramenta muito poderosa. A linguagem de programação usada foi o Python, por ser uma linguagem de alto nível, orientada a objetos, de tipagem dinâmica, forte e de fácil implementação.

4 RESULTADOS E DISCUSSÃO

Base de Dados

O site www.kaggle.com é uma plataforma online popular entre cientistas de dados, engenheiros de aprendizado de máquina e entusiastas da análise de dados, onde empresas e organizações lançam desafios para a comunidade. A base de dados do presente estudo foi escolhida devido às suas características ideais para a aplicação de aprendizado de máquina. Esses dados fornecem informações sobre clientes do setor de serviços de telecomunicações e foram disponibilizados com o objetivo de desenvolver um modelo preditivo capaz de prever o *churn* com base nas informações dos clientes. O dataset contém 3150 registros para cada uma das 16 variáveis disponíveis, conforme descritas na Tabela 1.

Tabela 1 - Descrição e tipos de das variáveis da base de dados

Variável	Tipo	Descrição em Portugues
Call Failure	Int64	Falha na chamada
Complains	Int64	reclamação
Subscription Length	Int64	Duração da assinatura
Charge Amount	Int64	Quantidade de carga
Seconds of Use	Int64	Segundos de Uso
Frequency of use	Int64	Frequência de uso
Distinct Called Numbers	Int64	Números chamados distintos
Age Group	Int64	Grupo de idade
Tariff Plan	Int64	Plano tarifário
Status	Int64	Status
age	Int64	idade
Customer Value	Float64	Valor para o cliente
FN	Float64	Falso Negativo
FP	Float64	Falso Positivo
Churn	Int64	(1 = Sim; 0 = Não)

Fonte: Elaborado pelo autor.

As variáveis FN e FP são colunas que indicam um resultado de Falso Negativo (não churn, mas que era um churn), Falso Positivo (predito como churn, mas que era um não churn) e para fins de criar o modelo de previsão foi consideradas irrelevantes, portanto não foram consideradas neste estudo.

Foram então utilizadas as variáveis restantes, nas quais temos a variável dependente Churn e nas variáveis independentes, temos onze variáveis numéricas discretas e contínuas . A variável *Call Failure* (indica quantas vezes aquele cliente teve falha no serviço), a variável Complains (quantidade de reclamações registradas), a variável *Subscription Length* (contém a duração do plano contratado), *Charge Amount* (quantidade de recarga), *Seconds of Use* (segundos de uso do serviço), *Frequency of use* (indica a frequência de uso pelo cliente), *Distinct Called Numbers* (indica ligações feitas e recebidas distintas), *Age Group* (grupo de

idade do cliente), *Tariff Plan* (valor do serviço contratado), *Status* (indica se o cliente está ou não ativo) e por fim, a variável *Customer Value* (valor para o cliente).

Tratamento dos Dados

A primeira coisa feita no dataset (base de dados) na Figura 9, foi fazer o tratamento. Nessa primeira etapa da criação do modelo, é preciso analisar e entender os tipos de variáveis contidas. Para fazer isso foi utilizado a biblioteca Pandas do Python. Pandas é uma biblioteca para Ciência de dados de código aberto que proporciona uma abordagem rápida e simples, com estruturas robustas para analisar e modelar os dados, e também tem uma ótima interação com outras bibliotecas usadas neste trabalho como: Numpy, *Scikit-Learn,* Matplotlib entre outras.

Figura 9 - Visualização do Dataset

```
dados.head(10)
```

	Call Failure	Complains	Subscription Length	Charge Amount	Seconds of Use	Frequency of use	Frequency of SMS	Distinct Called Numbers	Age Group	Tariff Plan	Status	Age	Customer Value	FN	FP	Churn
0	8	0	38	0	4370	71	5	17	3	1	1	30	197.640	177.8760	69.7640	0
1	0	0	39	0	318	5	7	4	2	1	2	25	46.035	41.4315	60.0000	0
2	10	0	37	0	2453	60	359	24	3	1	1	30	1536.520	1382.8680	203.6520	0
3	10	0	38	0	4198	66	1	35	1	1	1	15	240.020	216.0180	74.0020	0
4	3	0	38	0	2393	58	2	33	1	1	1	15	145.805	131.2245	64.5805	0
5	11	0	38	1	3775	82	32	28	3	1	1	30	282.280	254.0520	78.2280	0
6	4	0	38	0	2360	39	285	18	3	1	1	30	1235.960	1112.3640	173.5960	0
7	13	0	37	2	9115	121	144	43	3	1	1	30	945.440	850.8960	144.5440	0
8	7	0	38	0	13773	169	0	44	3	1	1	30	557.680	501.9120	105.7680	0
9	7	0	38	1	4515	83	2	25	3	1	1	30	191.920	172.7280	69.1920	0

Fonte: elaborado pelo autor

De acordo com a Figura 9, como o dataset é muito grande, o modo de visualização foi reduzido. Na primeira análise foi removido as variáveis FN e FP pois como explicado anteriormente pode atrapalhar a construção do modelo. Outro ponto bem visível quando olhamos para os dados é que todos os dados são numéricos e não nulos, conforme figura 10.

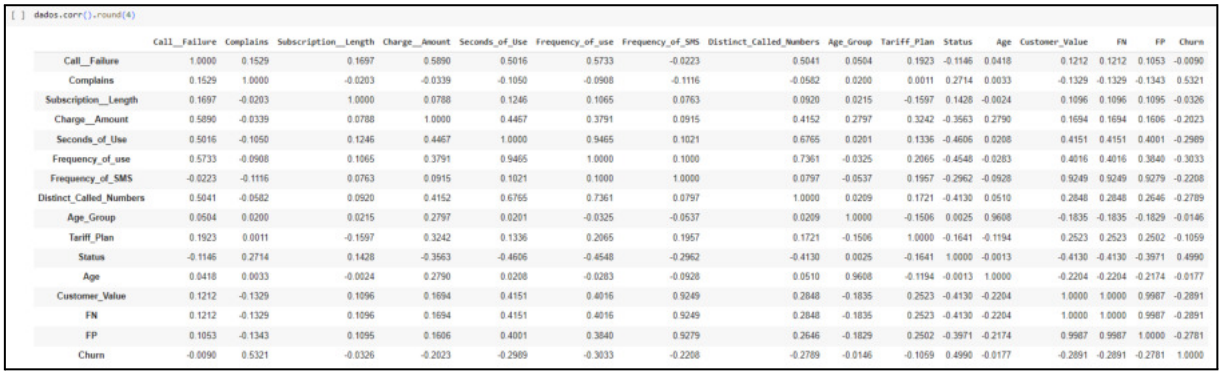
Figura 10 - Visualização descritiva dos dados

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3150 entries, 0 to 3149
Data columns (total 16 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   Call__Failure           3150 non-null   int64
 1   Complains               3150 non-null   int64
 2   Subscription__Length    3150 non-null   int64
 3   Charge__Amount          3150 non-null   int64
 4   Seconds_of_Use          3150 non-null   int64
 5   Frequency_of_use        3150 non-null   int64
 6   Frequency_of_SMS        3150 non-null   int64
 7   Distinct_Called_Numbers 3150 non-null   int64
 8   Age_Group               3150 non-null   int64
 9   Tariff_Plan             3150 non-null   int64
 10  Status                  3150 non-null   int64
 11  Age                     3150 non-null   int64
 12  Customer_Value          3150 non-null   float64
 13  FN                      3150 non-null   float64
 14  FP                      3150 non-null   float64
 15  Churn                   3150 non-null   int64
dtypes: float64(3), int64(13)
memory usage: 393.9 KB
```

Fonte: elaborado pelo autor

Entender como as variáveis independentes se relacionam com a variável dependente é outro ponto importante, para a escolha das variáveis independentes corretas. Como mostrado na figura 11, usando uma função da biblioteca Pandas é possível montar a matriz de correlação. Através dessa tabela podemos visualizar o coeficiente de correlação, que é uma medida de associação linear entre duas variáveis e situa-se entre -1 e +1 sendo que -1 indica associação negativa perfeita e +1 indica associação positiva perfeita. Quando o coeficiente de correlação está entre -1 e +1, isso indica que temos uma relação entre as variáveis e por aquele ponto está passando uma reta.

Figura 11 - Matriz de correlação



```
[ ] dados.corr().round(4)
```

	Call__Failure	Complains	Subscription__Length	Charge__Amount	Seconds_of_Use	Frequency_of_use	Frequency_of_SMS	Distinct_Called_Numbers	Age_Group	Tariff_Plan	Status	Age	Customer_Value	FN	FP	Churn
Call__Failure	1.0000	0.1529	0.1697	0.5890	0.5016	0.5733	-0.0223	0.5041	0.0504	0.1923	-0.1146	0.0418	0.1212	0.1212	0.1053	-0.0090
Complains	0.1529	1.0000	-0.0203	-0.0339	-0.1050	-0.0908	-0.1116	-0.0582	0.0200	0.0011	0.2714	0.0033	-0.1329	-0.1329	-0.1343	0.5321
Subscription__Length	0.1697	-0.0203	1.0000	0.0788	0.1246	0.1065	0.0763	0.0920	0.0215	-0.1597	0.1428	-0.0024	0.1096	0.1096	0.1095	-0.0326
Charge__Amount	0.5890	-0.0339	0.0788	1.0000	0.4467	0.3791	0.0915	0.4152	0.2797	0.3242	-0.3563	0.2790	0.1694	0.1694	0.1606	-0.2023
Seconds_of_Use	0.5016	-0.1050	0.1246	0.4467	1.0000	0.9465	0.1021	0.6765	0.0201	0.1336	-0.4606	0.0208	0.4151	0.4151	0.4001	-0.2989
Frequency_of_use	0.5733	-0.0908	0.1065	0.3791	0.9465	1.0000	0.1000	0.7361	-0.0325	0.2065	-0.4548	-0.0283	0.4016	0.4016	0.3840	-0.3033
Frequency_of_SMS	-0.0223	-0.1116	0.0763	0.0915	0.1021	0.1000	1.0000	0.0797	-0.0537	0.1957	-0.2962	-0.0928	0.9249	0.9249	0.9279	-0.2208
Distinct_Called_Numbers	0.5041	-0.0582	0.0920	0.4152	0.6765	0.7361	0.0797	1.0000	0.0209	0.1721	-0.4130	0.0510	0.2848	0.2848	0.2646	-0.2789
Age_Group	0.0504	0.0200	0.0215	0.2797	0.0201	-0.0325	-0.0537	0.0209	1.0000	-0.1506	0.0025	0.9608	-0.1835	-0.1835	-0.1829	-0.0146
Tariff_Plan	0.1923	0.0011	-0.1597	0.3242	0.1336	0.2065	0.1957	0.1721	-0.1506	1.0000	-0.1641	-0.1194	0.2523	0.2523	0.2502	-0.1059
Status	-0.1146	0.2714	0.1428	-0.3563	-0.4606	-0.4548	-0.2962	-0.4130	0.0025	-0.1641	1.0000	-0.0013	-0.4130	-0.4130	-0.3971	0.4990
Age	0.0418	0.0033	-0.0024	0.2790	0.0208	-0.0283	-0.0928	0.0510	0.9608	-0.1194	-0.0013	1.0000	-0.2204	-0.2204	-0.2174	-0.0177
Customer_Value	0.1212	-0.1329	0.1096	0.1694	0.4151	0.4016	0.9249	0.2848	-0.1835	0.2523	-0.4130	-0.2204	1.0000	1.0000	0.9987	-0.2891
FN	0.1212	-0.1329	0.1096	0.1694	0.4151	0.4016	0.9249	0.2848	-0.1835	0.2523	-0.4130	-0.2204	1.0000	1.0000	0.9987	-0.2891
FP	0.1053	-0.1343	0.1095	0.1606	0.4001	0.3840	0.9279	0.2646	-0.1829	0.2502	-0.3971	-0.2174	0.9987	0.9987	1.0000	-0.2781
Churn	-0.0090	0.5321	-0.0326	-0.2023	-0.2989	-0.3033	-0.2208	-0.2789	-0.0146	-0.1059	0.4990	-0.0177	-0.2891	-0.2891	-0.2781	1.0000

Fonte: elaborado pelo autor

Depois de ter feito todos os ajustes e modelagem dos dados, na teoria poderíamos utilizar todas as variáveis que não tem um coeficiente de correlação igual a ‘0’ no modelo, que quando é ‘0 indica que não temos reta naquele ponto. Entretanto foi testado com todas as variáveis independentes, adicionando uma por vez , e as variáveis independentes que apresentaram o melhor resultado com o método da Regressão Logística, foram as variáveis conforme na Figura 12.

Figura 12 - Visualização dos dados após ajustes e modelage

	Call__Failure	Complains	Subscription__Length	Charge__Amount	Frequency_of_use	Age_Group
0	8	0	38	0	71	3
1	0	0	39	0	5	2
2	10	0	37	0	60	3
3	10	0	38	0	66	1
4	3	0	38	0	58	1

Fonte: elaborado pelo o autor.

Scikit-Learn é uma biblioteca construída na linguagem Python de código aberto, utilizada para aprendizado de máquina e nela encontramos variáveis funções e métodos já prontos, como *LogisticRegression* (Regressão Logística) , *tree.DecisionTreeClassifier* (Árvore de Decisão) entre outros que vamos utilizar neste trabalho.

Para construção do modelo para predição *churn*. É considerando uma boa prática separar o dataset em duas partes , sendo 75 % dos dados para treino e 25 % para teste. Como mostrado nas Figuras 13 e 14 a seguir.

Figura 13 - Separação dos dados

```
[14] y = dados["Churn"]

[15] x = dados[['Complains','Charge__Amount','Seconds_of_Use','Frequency_of_use','Frequency_of_SMS','Status','Customer_Value']]
```

Fonte: elaborado pelo o autor.

Na Figura 13 é mostrado como carregamos e separamos as variáveis independentes X da dependente Y para fazer a manipulação. A biblioteca Pandas ajuda muito nessa parte, facilitando o trabalho de manipulação com grandes volumes de dados.

Figura 14 - Modelo *LogisticRegression*

```
[ ]  from sklearn.model_selection import train_test_split
     from sklearn.linear_model import LogisticRegression
     from sklearn.metrics import  accuracy_score

     X_train, X_test, y_train, y_test  = train_test_split(x, y,random_state=12, test_size= 0.25, stratify = y)
     modelo = LogisticRegression(random_state=12)
     modelo.fit(X_train, y_train)
     previsoes = modelo.predict(X_test)
```

Fonte: elaborado pelo autor.

É notável na figura 14 como a linguagem Python é poderosa, em poucas linhas de código o modelo preditivo é construído. De acordo com a imagem da figura 14, primeiro é feito a importação da biblioteca sklearn e seus métodos : *'train_test_split'* (esse método é usado para fazer a divisão do dataset em dados de treino e dados de teste), *'LogisticRegression'* (esse é o método da regressão logística, usado como preditor), *accuracy_score* (É usado para medir a assertividade do modelo, quanto a sua capacidade de predição).

Inicialmente, chamamos a função *'train_test_split'* para fazer a divisão dos dados. Passando como parâmetro a variável 'x' (contendo as variáveis independentes), a Variável 'y' (Nossa dependente com a coluna *churn*), *'rondom_state=12'* (é usado para controlar a aleatoriedade durante a divisão dos dados em conjuntos de treino e teste, quando executamos o código mais vezes). O parâmetro *'test_size=0,25'* (serve para setar o tamanho dos dados de teste, ou seja 75% para treino 25% para teste) , *'stratify = y'* (é utilizado para garantir que a distribuição das classes no conjunto de dados seja mantida nas divisões de treino e teste. Isso é particularmente útil em conjuntos de dados desequilibrados, nos quais uma ou mais classes têm uma presença significativamente menor do que outras, é uma forma de balancear os dados.).

As variáveis *"x_train, x_test e y_train, y_test"*, são carregados com os dados treino x, dados teste x, dados treino y, dados teste y, respectivamente.

Em seguida, chamamos o modelo *'LogisticRegression(rondom_state=12)'* com o mesmo parâmetro de aleatoriedade, pelo mesmo motivo utilizado no método *'train_test_split'*. Logo após chamamos o método *'modelo.fit(x_train, y_train)'* (Onde carrega os dados e faz o treinamento do modelo, onde o modelo aprende a relação entre os recursos *(x_train)* e os rótulos *(y_train)*. Durante o treinamento, os parâmetros internos do modelo são ajustados para minimizar a diferença entre as previsões do modelo e os rótulos

reais.). Depois que o modelo é treinado com método *'fit'*. Usamos o *'previsoes.predict(x_test)'*, passando os dados de teste *(x_test)* como argumento. Ele retorna as previsões do modelo para esses dados de teste.

Agora temos as previsões feitas pelo modelo *('previsoes')*. No próximo passo faremos aplicação como os dados de *'y_test'*, dados reservados para testar o modelo. Para isso usaremos as métricas.

A matriz de confusão é uma ferramenta muito utilizada para avaliar modelos de aprendizado de Máquina de classificação. Ela consiste em uma matriz em que as linhas representam os valores reais e as colunas representam os valores preditos. Cada espaço da matriz passa a ser um diagnóstico. A ideia geral é contabilizar a quantidade de vezes que um determinado A é classificado como valor B. As siglas associadas são as seguintes:

- Falso Negativo (FN) representa os clientes que abandonaram o serviço (*churn* - classe 1), mas o modelo errou ao classificá-los como não *churn* (classe 0), indicando incorretamente que não abandonaram o serviço.

- Verdadeiro Negativo (VN) representa os clientes que não abandonaram o serviço (não *churn* - classe 0), e o modelo acertou ao classificá-los como não *churn* (classe 0), indicando que não abandonaram o serviço.

- Verdadeiro Positivo (VP) representa os clientes que abandonaram o serviço (*churn* - classe 1), e o modelo acertou ao classificá-los como *churn* (classe 1), indicando que realmente abandonaram o serviço.

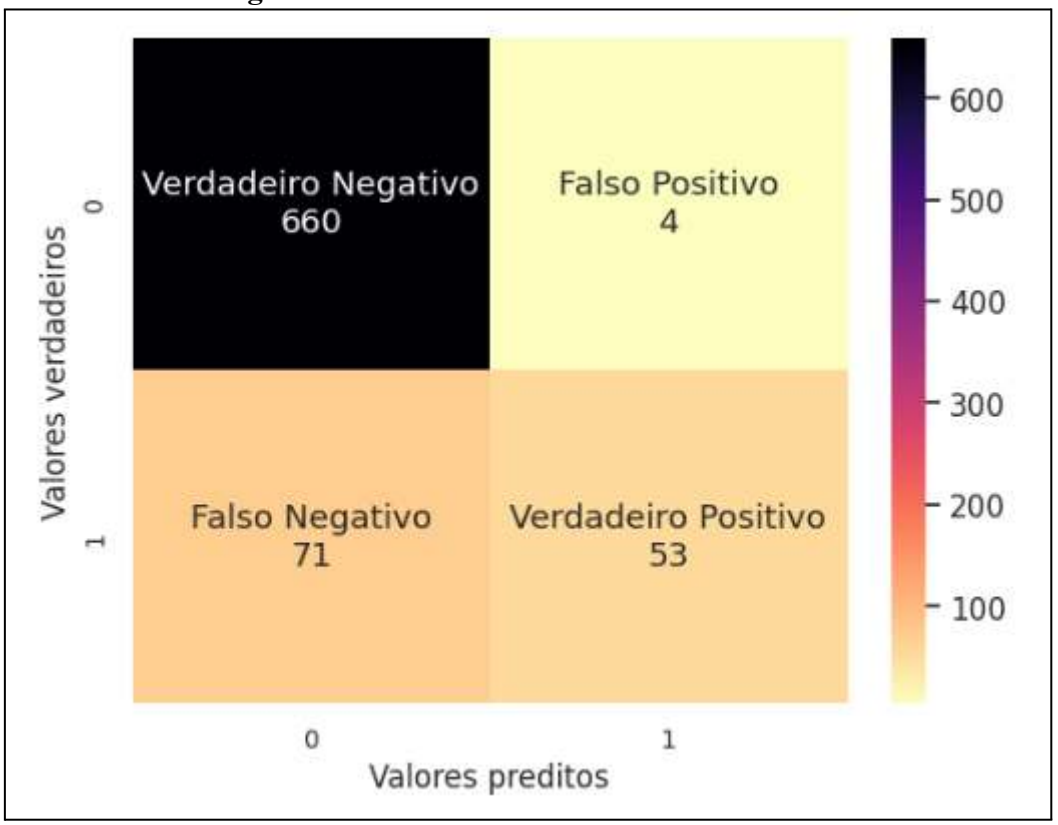
- Falso Positivo (FP) representa os clientes que não abandonaram o serviço (não *churn* - classe 0), mas o modelo errou ao classificá-los como *churn* (classe 1), indicando incorretamente que abandonaram o serviço.

Figura 15 - Matriz de Confusão

```
[124] from sklearn.metrics import confusion_matrix

[125] mc = confusion_matrix(y_test, previsoes)  # Matriz de confusão
      mc

      array([[660,   4],
             [ 71,  53]])
```

Fonte: elaborado pelo autor.

Figura 16 - Gráfico da Matriz de Confusão Gerada.



Fonte: elaborado pelo autor.

Os valores da matriz de confusão na figura 15 e 16, são são distribuídos da seguinte forma:

- Como podemos ver o valor 660 representa os clientes que o modelo acertou, mas que não churn [classe 0]
- O valor 53 representa os clientes que o modelo acerto, e que são churn realmente [classe 1]

- Como podemos ver o valor 4 representa os clientes que o modelo errou mas que não são churn nesse [classe 0]
- O valor 71 representa os clientes que o modelo errou e que são churn nesse [classe 1]

A partir da matriz de confusão, poderemos compreender melhor os resultados das outras métricas utilizadas neste trabalho.

Acurácia

Figura 17- Acurácia

```
[128] acuracia = accuracy_score(y_test, previsoes) * 100
      print("A acurácia foi %.2f%%" % acuracia)

      A acurácia foi 90.48%
```

Fonte: elaborado pelo autor.

De acordo com a figura 17, podemos afirmar que a assertividade do modelo foi 90.48 %, que é um valor muito excelente. Para entender esse valor temos que olhar para a matriz de confusão da Figura 16.

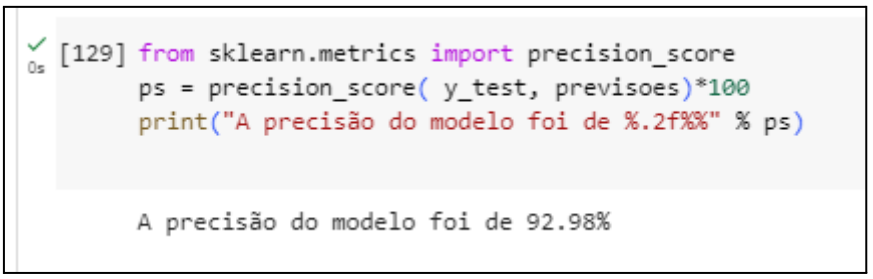
$$\text{Acurácia} = \frac{VP+VN}{total\ de\ predições} = \frac{53+660}{660+4+53+71} = 0,90482233.... \ ou\ 90,48\%$$

A matriz de confusão nos ajuda a compreender a métrica da acurácia. Nesse contexto, a acurácia é definida como a soma dos verdadeiros positivos e dos verdadeiros negativos, dividida pelo total de resultados.

Precisão

A precisão pode ser considerada a métrica mais importante para este trabalho, pois essa métrica nos diz a precisão que o modelo tem em prever o abandono a classe 1, como ilustrado na Figura 18 a seguir:

Figura 18 - Precisão



Fonte:elaborado pelo autor

A precisão foi de 92,98%, o que representa um valor excelente. Isso significa que o modelo possui uma precisão de 92,98% na previsão da classe 1, ou seja, na identificação do abandono do cliente. No entanto, para compreender como esse valor foi calculado, é necessário analisar a matriz de confusão nas Figuras 14 e 15.

$$\text{Precisão} = \frac{VP}{VP + FP} = \frac{53}{53+4} = \ 0,929824\ldots \ ou\ 92.98\%$$

Sensibilidade

A sensibilidade, frequentemente confundida com a precisão devido à semelhança de suas fórmulas, é, no entanto, uma medida que avalia quão bem o modelo é capaz de classificar os resultados verdadeiramente positivos. Isso é determinado pelo cálculo, conforme demonstrado na equação (5). A sensibilidade é calculada como a razão entre o número de verdadeiros positivos e a soma dos verdadeiros positivos e falsos negativos. A diferença está em vez de usar os falsos positivos, como na precisão, são utilizados os falsos negativos, que intuitivamente representam os verdadeiros positivos.

Figura 19 - Sensibilidade

```
from sklearn.metrics import recall_score

sensibilidade = recall_score(y_test, previsoes)*100
print("A sensibilidade do modelo foi de %.2f%%" % sensibilidade)

A sensibilidade do modelo foi de 42.74%
```

Fonte: elaborado pelo autor.

De acordo com o resultado apresentado na Figura 19, a sensibilidade foi de 42,74%. Embora não tenha sido muito alta, também não foi considerada ruim, como pode ser observado no cálculo da sensibilidade a seguir.

$$\text{Sensibilidade} = \frac{VP}{VP + FN} = \frac{53}{53+71} = 0,4274193.. \; ou \; 42.74\%$$

Os dados foram submetidos a testes utilizando o modelo conhecido como Árvore de Decisão, com o objetivo de comparar os resultados obtidos com dois modelos distintos. Os parâmetros empregados foram os mesmos utilizados no modelo de Regressão Logística. Isso foi feito para garantir que um modelo não apresentasse vantagem sobre o outro nos resultados.

Figura 20 - Dados do da daset

```
y = dados["Churn"]

x = dados[['Complains','Charge__Amount','Seconds_of_Use','Frequency_of_use','Frequency_of_SMS','Status','Customer_Value']]
```

Fonte: elaborado autor.

Conforme a Figura 20, utilizaremos as mesmas variáveis que foram empregadas no modelo anterior. Uma vez que os dados foram previamente analisados e tratados, não é necessário repetir esse processo.

Figura 21 - Modelo Árvore de decisão.

```
from sklearn.model_selection import train_test_split
from sklearn import tree
from sklearn.metrics import  accuracy_score

X_train, X_test, y_train, y_test  = train_test_split(x, y,random_state=12, test_size= 0.25, stratify = y)
modelo_arvore = tree.DecisionTreeClassifier(max_depth=10 , random_state=12)
modelo_arvore.fit(X_train, y_train)
previsoes_arvore = modelo_arvore.predict(X_test)
```

Fonte: elaborado pelo autor.

Um novo módulo da biblioteca scikit-learn que utilizaremos para a construção da Árvore de Decisão é o tree. O módulo tree do scikit-learn contém classes e funções relacionadas à construção de árvores de decisão, necessárias para a criação do modelo, conforme mostrado na Figura 21.

No módulo tree, encontramos o método *tree.DecisionTreeClassifier* (árvore de decisão). A estruturação do modelo é semelhante à do modelo usado anteriormente, mudando apenas o método, como pode ser observado. Foi utilizada a mesma divisão de parâmetros, incluindo o fator de aleatoriedade. No entanto, um parâmetro adicional que incorporamos ao modelo de árvore de decisão é *'max_depth=10'*, onde definimos a profundidade máxima da árvore.

Figura 22 - Gerando a figura da árvore.

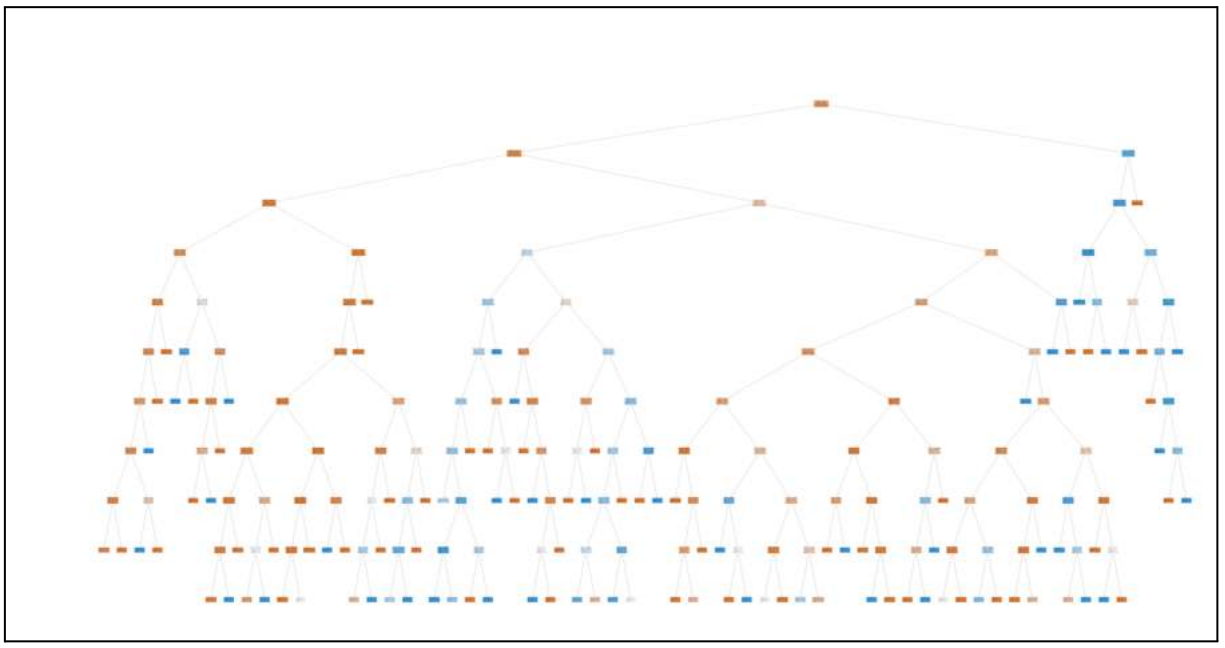
```
[ ]  import matplotlib.pyplot as plt

     def salvar_arvore(classificador, nome):
       plt.figure(figsize=(200,100))
       tree.plot_tree(classificador, filled=True, fontsize=14)
       plt.savefig(nome)
       plt.close()
```

```
#criacao da figura da arvore de decisao
salvar_arvore(modelo_arvore, "arvore_decisao_tcc.png")
```

Fonte: elaborado pelo autor.

Na Figura 22, o trecho de código gera uma imagem da árvore criada pelo modelo, como demonstrado na Figura 23 a seguir:

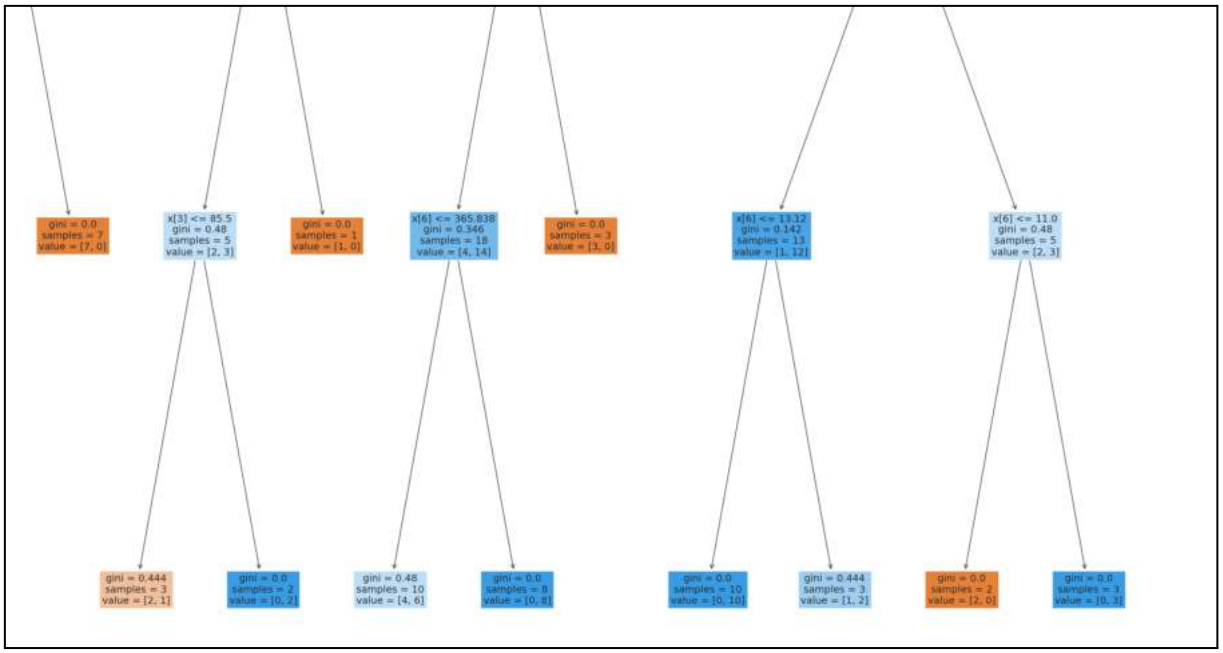
Figura 23 - Árvore Gerada



Fonte: elaborado pelo autor.

Como a árvore gerada é muito extensa, a visualização torna-se difícil. É necessário aplicar zoom na figura. No entanto, é possível observar como a árvore é construída pelo modelo e como ele toma decisões em cada ramo utilizado. Na Figura 23 a seguir, aumentaremos a resolução para uma melhor visualização.

Figura 24 - Árvore para metro GINI



Fonte: elaborado pelo autor.

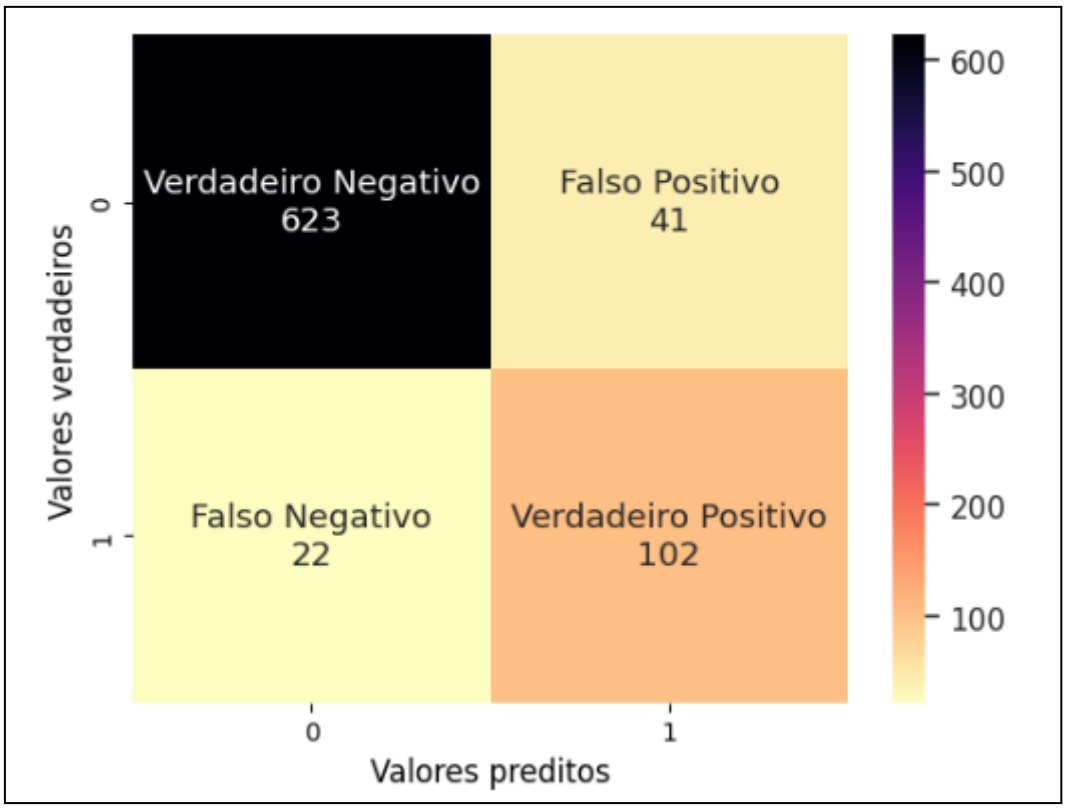
Quando ampliamos o zoom na Figura 24, podemos observar o parâmetro Gini. Gini é uma medida de impureza utilizada em algoritmos de árvore de decisão. O objetivo principal de uma árvore de decisão é dividir os dados em sub-ramos o mais puro possível, ou seja, em sub-ramos nos quais os exemplos de uma mesma classe estão agrupados juntos. O índice Gini mede a probabilidade de classificar erroneamente um elemento escolhido aleatoriamente. Quanto menor o valor do índice Gini, maior é a pureza do nó.

Figura 25 - Matriz de Confusão Árvore de Decisão.

```
[19] from sklearn.metrics import confusion_matrix

[20] mc = confusion_matrix(y_test,previsoes_arvore)  # Matriz de confusão
     mc

     array([[623,  41],
            [ 22, 102]])
```

Fonte: elaborado pelo autor.

Figura 26 - Gráfico Matriz de Confusão.



Fonte: elaborado pelo autor.

Com base na matriz de confusão gerada, será aplicada também às demais validações, conforme realizado no modelo anterior, para comparar os resultados. Ao analisar a matriz de confusão, já podemos notar que os resultados foram diferentes em comparação com o modelo anterior.

Figura 27 - Acurácia da Árvore de Decisão.

```
acuracia_arvore = accuracy_score(y_test, previsoes_arvore) * 100
print("A acurácia foi %.2f%%" % acuracia_arvore)

A acurácia foi 92.01%
```

Fonte: elaborado pelo autor.

$$\text{Acurácia} = \frac{VP+VN}{total\ de\ predições} = \frac{102+623}{623+41+102+22} = \ 0,9200507\ldots\ ou\ 92,01\%$$

Figura 28 - Precisão Árvore de Decisão.

```
[25] from sklearn.metrics import precision_score

     ps_arvore_decisao = precision_score(y_test, previsoes_arvore)*100
     print("A precisão do modelo foi de %.2f%%" % ps_arvore_decisao)

     A precisão do modelo foi de 71.33%
```

Fonte: elaborado pelo autor.

$$\text{Precisão} = \frac{VP}{VP + FP} = \frac{102}{102+41} = 0,71328671\ldots\ ou\ 71.33\%$$

Figura 29 - Sensibilidade da Árvore de Decisão.

```
[26] from sklearn.metrics import recall_score

[27] sensibilidade_arvore = recall_score(y_test, previsoes_arvore)*100
     print("A sensibilidade do modelo foi de %.2f%%" % sensibilidade_arvore)

     A sensibilidade do modelo foi de 82.26%
```

Fonte: elaborado pelo autor.

$$\text{Sensibilidade} = \frac{VP}{VP + FN} = \frac{102}{102+22} = 0,822580..\ ou\ 82.26\%$$

Como podemos observar na Figura 27, a acurácia da Árvore de Decisão foi superior à da Regressão Logística. Na Figura 29, a sensibilidade apresentou um resultado consideravelmente melhor. Entretanto, na Figura 28, mesmo obtendo um ótimo resultado de 71,33%, ainda assim ficou aquém do desempenho alcançado pela Regressão Logística nesta métrica.

4.1 Comparação dos Resultados

Para a escolha do modelo de aprendizado de máquina que obteve melhor desempenho na predição do churn, vamos comparar os resultados obtidos por meio das métricas de validação. Isso determinará a melhor opção para esse problema. Neste trabalho, apresento um estudo detalhado. A seguir, temos a Tabela 2 com os resultados de cada modelo.

Tabela 2 - Resultados Regressão Logística versus Árvore de Decisão

Regressão Logística	Árvore de Decisão
Acurácia $0,90482233....\ ou\ 90,48\%$	Acurácia $0,9200507....\ ou\ 92,01\%$
Precisão $0,929824...\ ou\ 92.98\%$	Precisão $0,71328671...\ ou\ 71.33\%$
Sensibilidade 0,4274193.. ou 42.74%	Sensibilidade $0,822580..\ ou\ 82.26\%$

Com base nos resultados obtidos, o modelo escolhido como o melhor para a predição de churn neste trabalho foi a Regressão Logística. Isso se deve ao fato de que, mesmo perdendo em acurácia e sensibilidade, apresentou uma precisão significativamente maior em comparação com o modelo de Árvore de Decisão. Dado que o objetivo deste trabalho está centrado na definição para retenção de clientes, optamos pela precisão, pois esta métrica está relacionada à classe 1 (abandono).

Essa escolha permite que a empresa tome decisões e medidas preventivas antes que o cliente abandone o serviço. Ao ter a probabilidade de que um determinado cliente possa deixar o serviço, é possível analisar os fatores causais e, assim, mitigar o abandono por meio de ações proativas. O código completo dos modelos utilizados está nos Apêndice A e B.

5 CONCLUSÃO

Neste trabalho, foram desenvolvidos dois modelos de aprendizado de máquina: Regressão Logística e Árvore de Decisão, com o objetivo de prever o abandono de clientes em uma operadora de telecomunicações, utilizando aprendizado supervisionado. A intenção é permitir que a empresa adote medidas proativas com antecedência, com base nos resultados da previsão de churn do cliente, evitando assim o cancelamento do serviço.

Na etapa inicial do desenvolvimento, realizou-se uma revisão bibliográfica sobre trabalhos já desenvolvidos na área, optando-se pela escolha dos métodos. Na segunda etapa, aplicaram-se os modelos preditivos a uma base de dados de uma empresa de telecomunicações obtida do repositório *Kaggle*. Na avaliação dos critérios para ambos os métodos, os resultados foram satisfatórios. No entanto, ao considerar a métrica mais relevante para a escolha do modelo, a precisão, a Regressão Logística apresentou melhor desempenho na previsão de churn do cliente, alcançando 92.98% de assertividade, em comparação com os 71.33% da Árvore de Decisão. Portanto, para este trabalho, conclui-se que o problema de retenção é mais bem abordado pelo modelo de Regressão Logística. Dentre as variáveis contidas na base de dados, a que teve maior influência para o cancelamento do serviço foi a reclamação.

Devido aos resultados obtidos com o uso da inteligência artificial (IA) e reconhecendo que a previsão de resultados contribui para a retenção de clientes em operadoras de telecomunicações, pode-se afirmar que a inteligência artificial (IA) é uma ferramenta poderosa para a retenção de clientes. O autor acredita que o modelo de aprendizado de máquina desenvolvido pode ser adaptável para auxiliar na retenção em outras empresas prestadoras de serviços com modelos de assinatura. Além disso, na visão do autor, o modelo pode ser aplicado em outras áreas, como saúde, para detecção de doenças, e na área educacional, para prever possíveis evasões de alunos nas instituições de ensino.

REFERÊNCIAS

ABEL, Carol. O que é churn. **MINDMINERS**, 6 de Nov. 2017. Disponível em: <https://mindminers.com/blog/o-que-e-churn/L>. Acesso em: 02, Nov. e 2023.

ALURA. **Direto ao ponto: o que é Machine Learning com exemplos reais.** 19 de Jan. 2024. Il.color. Disponível em: https://www.alura.com.br/artigos/machine-learning . Acesso em: 24 de Fev. 2024.

CORDOVEZ, Diego. Entenda o que é churn e o que fazer para combatê-lo. **meetme**, 6 de Fev. 2023. Disponível em: https://meetime.com.br/blog/vendas/o-que-e-churn. Acesso em: . 02, Nov. de 2023.

ECKERT, Alex. MILAN, Gabriel Sperandio Sperandio.MECCA, Marlei Salete. NUNES, Grazieli Porto. Fatores determinantes para a retenção de clientes em escritórios de contabilidade: um estudo multicaso realizado em uma cidade da Serra Gaúcha. **Revista Eletrônica de Estratégia & Negócios**, Florianópolis, v.6,n.3,set./dez.2013. Disponível em: https://doi.org/10.19177/reen.v6e3201350-78. Acesso em: 02 de Nov. 2023

ESCOVEDO, Tatiana. Machine Learning: Conceitos e Modelos — Parte I: Aprendizado Supervisionado. **Medium.**, 28 de Jun. 2020. Disponível em: https://tatianaesc.medium.com/machine-learning-conceitos-e-modelos-f0373bf4f445#:~:text=O%20Na%C3%AFve%20Bayes%20(Bayes%20Ing%C3%AAnuo,n%C3%BAmero%20de%20atributos%20(caracter%C3%ADsticas). Acesso em: 15 Nov. de 2023.

DAMACENO, S. S.; VASCONCELOS, R. O. **INTELIGÊNCIA ARTIFICIAL: UMA BREVE ABORDAGEM SOBRE SEU CONCEITO REAL E O CONHECIMENTO POPULAR.** Caderno de Graduação - Ciências Exatas e Tecnológicas - UNIT - SERGIPE, [S.l.], v.5 ,n.1, p.11,2018. Disponível em: https://periodicos.grupotiradentes.com/cadernoexatas/article/view/5729. Acesso em: 6 nov. 2023.

DISTRIBUIÇÃO DE BERNOULLI. In: **WIKIPÉDIA**, a enciclopédia livre. Flórida: Wikimedia Foundation, 2021. Disponível em: <https://pt.wikipedia.org/w/index.php?title=Distribui%C3%A7%C3%A3o_de_Bernoulli&oldid=62003948>. Acesso em: 21 nov.. 2023.

GOMES, Elisabeth Braz Pereira; BRAGA, Fabiana dos Reis. I**nteligência competitiva em tempos de Big Data: Analisando informações e identificando tendências em tempo real.** Rio de Janeiro: ALTA BOOKS, 2017.

FERREIRA, Célia Marina Costa. **Um estudo sobre fidelização e retenção de clientes na área do fitness**. Dissertação de Mestrado, [s. l.], 2012. Disponível em: https://repositorio.ipcb.pt/handle/10400.11/1701. Acesso em: 19 out. 2023.

JAIME. Diferenças entre aprendizado supervisionado e não supervisionado. **DIO.**, 30 de set. 2022. Disponível em: <https://www.dio.me/articles/diferencas-entre-aprendizado-supervisionado-e-nao-supervisionado>. Acesso em: 11 Nov. de 2023.

GNOATTO, Ana Cristina. **Análise do desempenho de hiperparâmetros de aprendizagem de máquina aplicando na previsão da taxa de rotatividade de clientes.** 2023 f. Curso de Sistemas de informação - Universidade do Vale do Taquari-Univates,Taquari-Univates, 2023.

GARCIA, Silva. Ética e Inteligência Artificial. **SBCOPENLIB**. 11 de Nov, 2020. Disponível em: https://sol.sbc.org.br/journals/index.php/comp-br/article/view/1791. Acesso em: 2 Nov. 2023.

GONZALES, Leandro de Azevedo. **Regressão Logística e suas Aplicações.** Curso de Graduação em Ciência da Computação - Centro Ciências Exatas e Tecnológicas Monografia (Graduação) - Universidade Federal do Maranhão - São Luís, 2018. Disponível em: https://monografias.ufma.br/jspui/bitstream/123456789/3572/1/LEANDRO-GONZALEZ.pdf. Acesso em: 17 nov. 2023.

IGNACIO, Lucas França Ferreira. **Aprendizado de máquina: da teoria à aplicação.** 2021. 80f. Trabalho de Conclusão de Curso (Graduação em Matemática) - Instituto de Ciências Exatas, Universidade Federal Fluminense, Volta Redonda, 2021.

LOPES, André. **Prática: Árvore de Decisão**. 17 de abril. 2023. Il.color. Disponível em: https://brains.dev/2023/pratica-arvores-de-decisao/ . Acesso em: 27 nov. 2023.

MARIN, Maikon Aloan. **Introdução de Árvore de Decisão para a inferência de Redes Gênicas**.11 de 2013. Relatório de pesquisa programa de Iniciação científica - Tecnologia em Análise e Desenvolvimento de Sistemas - Universidade Tecnológica do Paraná. Disponível em:http://paginapessoal.utfpr.edu.br/fabricio/fabricio-martins-lopes/pesquisa/orientacoes/relatorio-pibic-2013-maikon-marin.pdf . Acesso em: 30 nov. 2023.

MILAN,Gabriel Sperandio. TONI,Deonir De. A construção de um modelo sobre a retenção de clientes e seus antecedentes em um ambiente de serviços.REAd. **Rev. eletrôn. adm.** Porto Alegre,Edição 72 - N° 2 – maio/agosto 2012 – p. 433-467 . Disponível em: https://doi.org/10.1590/S1413-23112012000200006. Acesso em: 02 de Nov. 2023 .

O que é Inteligência Artificial e Aprendizado de Máquina. **Medium**, 2022. Disponível em: <https://hlima.me/o-que-%C3%A9-intelig%C3%AAncia-artificial-e-aprendizado-de-m%C3%A1quina-92c0903ee7ea>. Acesso em: 08 nov. de 2023.

SERPA, Maria Luiza Rabelo. **Random Forest aplicado na análise de Churn:comparação do ajuste com dados completos versus ajustes em estratos definidos por variável categórica** . 2023. 9,10 f. Pós-Graduação - Universidade Federal de Minas Gerais, Minas Gerais, 2023.

SILVEIRA, Caio Cesar Vieira Trinta da. **Revisão e aplicação de métodos de aprendizado de máquina para a predição de Churn**. Dissertação de Mestrado, [s. l.], 2022. Disponível em:http://hdl.handle.net/11422/18441. Acesso em: 19 out. 2023.

RODRIGUES, Daniel Fredo. **Um estudo sobre o gerenciamento de churn e a fidelização de clientes em uma empresa de telecomunicação** . 2020. 12, f. Departamento de Ciências Administrativas, Escolas de Administração - Universidade Federal do Rio Grande do Sul, Rio Grande do Sul, 2020.

ROMULO, Silva. Enciclopédia da Conscienciologia: Inteligência Artificial. , **DSPACE**. 13 de Jan, 2013. Disponível em: http://reposicons.org/jspui/handle/123456789/3737. Acesso em: 2 Nov. 2023.

RI.TELEFONIA. **Telefonica Brasil Apresentação 2T23**. 25 de Jul. 2023. Il.color. Disponível em: https://ri.telefonica.com.br/pt/documentos/2929-Telefonica-Brasil-Apresentacao-2T23.pdf . Acesso em: 16 dez. 2023

TAURION, Cezar. **O potencial (e as limitações) da inteligência artificial**. 21 de nov. 2012. Il.color. Disponível em: https://neofeed.com . Acesso em: 05 nov. 2023.

TELECO. **Operadoras de celular no Brasil**. 08 de nov. 2023. Il.color. Disponível em: https://www.teleco.com.br/opcelular.asp . Acesso em: 13 dez. 2023.

WIKIPÉDIA**, Função Logística.** Il.color. Disponível em: https://Wiki.com . Acesso em: 17 nov. 2023

.

APÊNDICE A - Regressão Logística


```
import matplotlib.pyplot as plt
%matplotlib inline

import pandas as pd
import numpy as np

import warnings
warnings.filterwarnings('ignore') # warnings.filterwarnings(action'once')

dados = pd.read_csv('Customer Churn.csv',sep = ',')
```

```
dados.head(10)
```

	Call Failure	Complains	Subscription Length	Charge Amount	Seconds of Use	Frequency of use	Frequency of SMS	Distinct Called Numbers	Age Group	Tariff Plan	Status	Age	Customer Value	FN
0	8	0	38	0	4370	71	5	17	3	1	1	30	197.640	177.8760
1	0	0	39	0	318	5	7	4	2	1	2	25	46.035	41.4315
2	10	0	37	0	2453	60	359	24	3	1	1	30	1536.520	1382.8680
3	10	0	38	0	4198	66	1	35	1	1	1	15	240.020	216.0180
4	3	0	38	0	2393	58	2	33	1	1	1	15	145.805	131.2245
5	11	0	38	1	3775	82	32	28	3	1	1	30	282.280	254.0520
6	4	0	38	0	2360	39	285	18	3	1	1	30	1235.960	1112.3640
7	13	0	37	2	9115	121	144	43	3	1	1	30	945.440	850.8960
8	7	0	38	0	13773	169	0	44	3	1	1	30	557.680	501.9120

```
dados.columns = dados.columns.str.replace(' ', '_')
```

```
dados.head(10)
```

	Call__Failure	Complains	Subscription__Length	Charge__Amount	Seconds_of_Use	Frequency_of_use	Frequency_of_SMS	Distinct_Call
0	8	0	38	0	4370	71	5	
1	0	0	39	0	318	5	7	
2	10	0	37	0	2453	60	359	
3	10	0	38	0	4198	66	1	
4	3	0	38	0	2393	58	2	
5	11	0	38	1	3775	82	32	
6	4	0	38	0	2360	39	285	
7	13	0	37	2	9115	121	144	
8	7	0	38	0	13773	169	0	
9	7	0	38	1	4515	83	2	

```
dados.shape
```

```
(3150, 16)
```

```
type(dados)
```

```
pandas.core.frame.DataFrame
```

```
# Obtendo mais inforçoes com a função "info()"
dados.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3150 entries, 0 to 3149
Data columns (total 16 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   Call__Failure           3150 non-null   int64
 1   Complains               3150 non-null   int64
 2   Subscription__Length    3150 non-null   int64
 3   Charge__Amount          3150 non-null   int64
```

```
 4   Seconds_of_Use           3150 non-null   int64
 5   Frequency_of_use         3150 non-null   int64
 6   Frequency_of_SMS         3150 non-null   int64
 7   Distinct_Called_Numbers  3150 non-null   int64
 8   Age_Group                3150 non-null   int64
 9   Tariff_Plan              3150 non-null   int64
 10  Status                   3150 non-null   int64
 11  Age                      3150 non-null   int64
 12  Customer_Value           3150 non-null   float64
 13  FN                       3150 non-null   float64
 14  FP                       3150 non-null   float64
 15  Churn                    3150 non-null   int64
dtypes: float64(3), int64(13)
memory usage: 393.9 KB
```

```
# Qunatidade  de dados contidos no dataset
dados.shape[0]
```

```
3150
```

```
dados.shape
```

```
(3150, 16)
```

```
print('A base de dados apresenta {} registros e {} variáriavies'.format(dados.shape[0],dados.shape[1]))
```

```
A base de dados apresenta 3150 registros e 16 variáriavies
```

```
dados.corr().round(4)
```

	Call__Failure	Complains	Subscription__Length	Charge__Amount	Seconds_of_Use	Frequency_of_use	Frequency_
Call__Failure	1.0000	0.1529	0.1697	0.5890	0.5016	0.5733	
Complains	0.1529	1.0000	-0.0203	-0.0339	-0.1050	-0.0908	
Subscription__Length	0.1697	-0.0203	1.0000	0.0788	0.1246	0.1065	
Charge__Amount	0.5890	-0.0339	0.0788	1.0000	0.4467	0.3791	
Seconds_of_Use	0.5016	-0.1050	0.1246	0.4467	1.0000	0.9465	
Frequency_of_use	0.5733	-0.0908	0.1065	0.3791	0.9465	1.0000	
Frequency_of_SMS	-0.0223	-0.1116	0.0763	0.0915	0.1021	0.1000	
Distinct_Called_Numbers	0.5041	-0.0582	0.0920	0.4152	0.6765	0.7361	
Age_Group	0.0504	0.0200	0.0215	0.2797	0.0201	-0.0325	
Tariff_Plan	0.1923	0.0011	-0.1597	0.3242	0.1336	0.2065	
Status	-0.1146	0.2714	0.1428	-0.3563	-0.4606	-0.4548	
Age	0.0418	0.0033	-0.0024	0.2790	0.0208	-0.0283	
Customer_Value	0.1212	-0.1329	0.1096	0.1694	0.4151	0.4016	
FN	0.1212	-0.1329	0.1096	0.1694	0.4151	0.4016	
FP	0.1053	-0.1343	0.1095	0.1606	0.4001	0.3840	
Churn	-0.0090	0.5321	-0.0326	-0.2023	-0.2989	-0.3033	

```
#dados = dados.drop(columns = ["FN", "FP"], axis=1)
#dados.head(10)
```

```
y = dados["Churn"]
```

```
x = dados[['Complains','Charge__Amount','Seconds_of_Use','Frequency_of_use','Frequency_of_SMS','Status','Customer_Value']]
```

```
x.head()
```

	Complains	Charge__Amount	Seconds_of_Use	Frequency_of_use	Frequency_of_SMS	Sta
0	0	0	4370	71	5	
1	0	0	318	5	7	
2	0	0	2453	60	359	
3	0	0	4198	66	1	
4	0	0	2393	58	2	

```
y.head(100)
```

```
0     0
1     0
2     0
3     0
4     0
     ..
95    0
96    0
97    0
98    0
99    1
Name: Churn, Length: 100, dtype: int64
```

```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import  accuracy_score

X_train, X_test, y_train, y_test  = train_test_split(x, y,random_state=12, test_size= 0.25, stratify = y)
modelo = LogisticRegression(random_state=12)
modelo.fit(X_train, y_train)
previsoes = modelo.predict(X_test)


from sklearn.metrics import confusion_matrix


mc = confusion_matrix(y_test, previsoes)  # Matriz de confusão
mc
```

```
array([[660,   4],
       [ 71,  53]])
```

```
# Função para gerar a matriz de confusão como está no para saber mais.

import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np

def gerando_matriz(matriz_confusao, labels, categorias, cmap="viridis"):
    group_counts = [f"{value}" for value in matriz_confusao.flatten()]
    labels = [f"{v1}" for v1 in labels]
    lb = [f"{v1}\n{v2}" for v1, v2 in zip(labels, group_counts)]
    lb = np.asarray(lb).reshape(matriz_confusao.shape)

    ax = plt.subplot()
    sns.set(font_scale=1.1)
    sns.heatmap(matriz_confusao, annot=lb, ax=ax, cmap=cmap, fmt="", xticklabels=categorias,yticklabels=categorias)

    label_font = {'size':'12'}
    ax.set_xlabel('Valores preditos', fontdict=label_font);
    ax.set_ylabel('Valores verdadeiros', fontdict=label_font);

    ax.tick_params(axis='both', which='major', labelsize=10)


categorias = ["0", "1"]
labels = ['Verdadeiro Negativo','Falso Positivo',
          'Falso Negativo','Verdadeiro Positivo']

gerando_matriz(mc, labels, categorias, cmap="magma_r")
```


```
acuracia = accuracy_score(y_test, previsoes) * 100
print("A acurácia foi %.2f%%" % acuracia)
```

```
A acurácia foi 90.48%
```

```
from sklearn.metrics import precision_score
ps = precision_score( y_test, previsoes)*100
print("A precisão do modelo foi de %.2f%%" % ps)
```

```
A precisão do modelo foi de 92.98%
```

```
from sklearn.metrics import recall_score

sensibilidade = recall_score(y_test, previsoes)*100
print("A sensibilidade do modelo foi de %.2f%%" % sensibilidade)
```

```
A sensibilidade do modelo foi de 42.74%
```

```
from sklearn.metrics import f1_score

media_harmonica = f1_score(y_test, previsoes)* 100
print("A média harmônica entre o recall e a precisão do modelo foi de %.2f%%" % media_harmonica)
```

```
A média harmônica entre o recall e a precisão do modelo foi de 58.56%
```

APÊNDICE B - Árvore de Decisão


```
import matplotlib.pyplot as plt
%matplotlib inline

import pandas as pd
import numpy as np


import warnings
warnings.filterwarnings('ignore') # warnings.filterwarnings(action'once')


dados = pd.read_csv('Customer Churn.csv',sep = ',')


dados.head(10)
```

	Call Failure	Complains	Subscription Length	Charge Amount	Seconds of Use	Frequency of use	Frequency of SMS	Distinct Called Numbers
0	8	0	38	0	4370	71	5	17
1	0	0	39	0	318	5	7	4
2	10	0	37	0	2453	60	359	24
3	10	0	38	0	4198	66	1	35
4	3	0	38	0	2393	58	2	33
5	11	0	38	1	3775	82	32	28
6	4	0	38	0	2360	39	285	18
7	13	0	37	2	9115	121	144	43
8	7	0	38	0	13773	169	0	44

```
dados.columns = dados.columns.str.replace(' ', '_')


dados.head(10)
```

	Call__Failure	Complains	Subscription__Length	Charge__Amount	Seconds_of_Use	Fr
0	8	0	38	0	4370	
1	0	0	39	0	318	
2	10	0	37	0	2453	
3	10	0	38	0	4198	
4	3	0	38	0	2393	
5	11	0	38	1	3775	
6	4	0	38	0	2360	
7	13	0	37	2	9115	
8	7	0	38	0	13773	
9	7	0	38	1	4515	

```
dados.shape
```

```
(3150, 16)
```

```
type(dados)
```

```
pandas.core.frame.DataFrame
```

```
# Obtendo mais inforçoes com a função "info()"
dados.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3150 entries, 0 to 3149
Data columns (total 16 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   Call__Failure           3150 non-null   int64
 1   Complains               3150 non-null   int64
 2   Subscription__Length    3150 non-null   int64
 3   Charge__Amount          3150 non-null   int64
```

```
 4   Seconds_of_Use           3150 non-null   int64
 5   Frequency_of_use         3150 non-null   int64
 6   Frequency_of_SMS         3150 non-null   int64
 7   Distinct_Called_Numbers  3150 non-null   int64
 8   Age_Group                3150 non-null   int64
 9   Tariff_Plan              3150 non-null   int64
 10  Status                   3150 non-null   int64
 11  Age                      3150 non-null   int64
 12  Customer_Value           3150 non-null   float64
 13  FN                       3150 non-null   float64
 14  FP                       3150 non-null   float64
 15  Churn                    3150 non-null   int64
dtypes: float64(3), int64(13)
memory usage: 393.9 KB
```

```
# Qunatidade  de dados contidos no dataset
dados.shape[0]
```

```
3150
```

```
dados.shape
```

```
(3150, 16)
```

```
print('A base de dados apresenta {} registros e {} variáriavies'.format(dados.shape[0],dados.shape[1]))
```

```
A base de dados apresenta 3150 registros e 16 variáriavies
```

```
dados.corr().round(4)
```

	Call__Failure	Complains	Subscription__Length	Charge__Amou
Call__Failure	1.0000	0.1529	0.1697	0.58
Complains	0.1529	1.0000	-0.0203	-0.03
Subscription__Length	0.1697	-0.0203	1.0000	0.07
Charge__Amount	0.5890	-0.0339	0.0788	1.00
Seconds_of_Use	0.5016	-0.1050	0.1246	0.44
Frequency_of_use	0.5733	-0.0908	0.1065	0.37
Frequency_of_SMS	-0.0223	-0.1116	0.0763	0.09
Distinct_Called_Numbers	0.5041	-0.0582	0.0920	0.41
Age_Group	0.0504	0.0200	0.0215	0.27
Tariff_Plan	0.1923	0.0011	-0.1597	0.32
Status	-0.1146	0.2714	0.1428	-0.35
Age	0.0418	0.0033	-0.0024	0.27
Customer_Value	0.1212	-0.1329	0.1096	0.16
FN	0.1212	-0.1329	0.1096	0.16
FP	0.1053	-0.1343	0.1095	0.16
Churn	-0.0090	0.5321	-0.0326	-0.20

```
y = dados["Churn"]
```

```
x = dados[['Complains','Charge__Amount','Seconds_of_Use','Frequency_of_use','Frequency_of_SMS','Status','Customer_Value']]
```

```
x.head()
```

	Complains	Charge__Amount	Seconds_of_Use	Frequency_of_use	Frequency_of_SMS	Sta
0	0	0	4370	71	5	
1	0	0	318	5	7	
2	0	0	2453	60	359	
3	0	0	4198	66	1	
4	0	0	2393	58	2	

```
y.head(100)
```

```
0     0
1     0
2     0
3     0
4     0
     ..
95    0
96    0
97    0
98    0
99    1
Name: Churn, Length: 100, dtype: int64
```

```
from sklearn.model_selection import train_test_split
from sklearn import tree
from sklearn.metrics import  accuracy_score

X_train, X_test, y_train, y_test  = train_test_split(x, y,random_state=12, test_size= 0.25, stratify = y)
modelo_arvore = tree.DecisionTreeClassifier(max_depth=10 , random_state=12)
modelo_arvore.fit(X_train, y_train)
previsoes_arvore = modelo_arvore.predict(X_test)
```

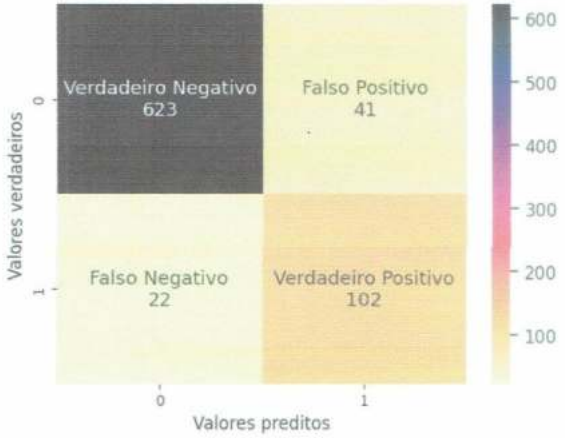
```
from sklearn.metrics import confusion_matrix
```

```
mc = confusion_matrix(y_test,previsoes_arvore)  # Matriz de confusão
mc
```

```
array([[623,  41],
       [ 22, 102]])
```

```
# Função para gerar a matriz de confusão como está no para saber mais.

import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np

def gerando_matriz(matriz_confusao, labels, categorias, cmap="viridis"):
    group_counts = [f"{value}" for value in matriz_confusao.flatten()]
    labels = [f"{v1}" for v1 in labels]
    lb = [f"{v1}\n{v2}" for v1, v2 in zip(labels, group_counts)]
    lb = np.asarray(lb).reshape(matriz_confusao.shape)

    ax = plt.subplot()
    sns.set(font_scale=1.1)
    sns.heatmap(matriz_confusao, annot=lb, ax=ax, cmap=cmap, fmt="", xticklabels=categorias,yticklabels=categorias)

    label_font = {'size':'12'}
    ax.set_xlabel('Valores preditos', fontdict=label_font);
    ax.set_ylabel('Valores verdadeiros', fontdict=label_font);

    ax.tick_params(axis='both', which='major', labelsize=10)
```

```
categorias = ["0", "1"]
labels = ['Verdadeiro Negativo','Falso Positivo',
          'Falso Negativo','Verdadeiro Positivo']

gerando_matriz(mc, labels, categorias, cmap="magma_r")
```




```
acuracia_arvore = accuracy_score(y_test, previsoes_arvore) * 100
print("A acurácia foi %.2f%%" % acuracia_arvore)
```

```
A acurácia foi 92.01%
```

```
print(" Quantidade de nós da arvore ",modelo_arvore.get_depth())
```

```
 Quantidade de nós da arvore  10
```

```
from sklearn.metrics import precision_score

ps_arvore_decisao = precision_score(y_test, previsoes_arvore)*100
print("A precisão do modelo foi de %.2f%%" % ps_arvore_decisao)
```

```
A precisão do modelo foi de 71.33%
```

```
from sklearn.metrics import recall_score

sensibilidade_arvore = recall_score(y_test, previsoes_arvore)*100
print("A sensibilidade do modelo foi de %.2f%%" % sensibilidade_arvore)
```

```
A sensibilidade do modelo foi de 82.26%
```

```
from sklearn.metrics import f1_score

media_harmonica_arvore = f1_score(y_test, previsoes_arvore)* 100
print("A média harmônica entre o recall e a precisão do modelo foi de %.2f%%" % media_harmonica_arvore)
```

```
A média harmônica entre o recall e a precisão do modelo foi de 76.40%
```

```
import matplotlib.pyplot as plt

def salvar_arvore(classificador, nome):
  plt.figure(figsize=(200,100))
  tree.plot_tree(classificador, filled=True, fontsize=14)
  plt.savefig(nome)
  plt.close()

#criacao da figura da arvore de decisao
salvar_arvore(modelo_arvore, "arvore_decisao_tcc.png")
```

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA
Campus João Pessoa - Código INEP: 25096850
Av. Primeiro de Maio, 720, Jaguaribe, CEP 58015-435, Joao Pessoa (PB)
CNPJ: 10.783.898/0002-56 - Telefone: (83) 3612.1200

Documento Digitalizado Ostensivo (Público)

ENTREGA DA VERSÃO FINAL DE TCC

Assunto:	ENTREGA DA VERSÃO FINAL DE TCC
Assinado por:	Damião Conceicão
Tipo do Documento:	Anexo
Situação:	Finalizado
Nível de Acesso:	Ostensivo (Público)
Tipo do Conferência:	Cópia Simples

Documento assinado eletronicamente por:

- **Damião Otávio da Conceição**, **ALUNO (20192430031) DE TECNOLOGIA EM SISTEMAS DE TELECOMUNICAÇÕES - JOÃO PESSOA**, em 18/03/2024 19:00:07.

Este documento foi armazenado no SUAP em 18/03/2024. Para comprovar sua integridade, faça a leitura do QRCode ao lado ou acesse https://suap.ifpb.edu.br/verificar-documento-externo/ e forneça os dados abaixo:

Código Verificador: 1119084
Código de Autenticação: 22e20adb22