Predição de “churn” em empresa bancária fictícia: aplicação de modelos de aprendizado de máquina

José Geovani Correia[1]*; Sandro Ricardo Fuzatto [2]

[1]Analista de Dados. Rua Eugênia Sá Vitale, 943– Taboão; 09665-000 São Bernardo do Campo, São Paulo, Brasil
[2] Engº Agrônomo especialista em Genética e Melhoramento de Plantas. Rua Phenom, 35 - Portal das Araras, 79644-256 Três Lagoas, Mato Grosso do Sul, Brasil
*autor correspondente: Geovani.correia@icloud.com

Predição de “churn” em empresa bancária fictícia: aplicação de modelos de aprendizado de máquina

Resumo

Nos últimos anos, o uso de técnicas de Aprendizado de Máquina para prever a saída de clientes (“churn”) tem crescido significativamente, especialmente no setor bancário, onde a retenção de clientes é mais econômica do que a aquisição de novos. Com base nessa premissa, este trabalho teve como objetivo aplicar e comparar três modelos de aprendizado de máquina preditivos para o evento de “churn” — Regressão Logística, “XGBoost” e “CATBoost” — aplicados a uma base de dados fictícia de um banco. Utilizando a metodologia CRISP-DM, foram avaliadas as performances dos modelos por meio das métricas “ROC AUC” e Recall. Nos resultados, o modelo “CATBoost” se destacou, alcançando a maior métrica “ROC AUC” de 0,87 e “Recall” de 0,88, superando os demais modelos. Também foi aplicado o método de "Permutation Importance" para identificar as variáveis mais influentes, destacando-se o número de produtos adquiridos, a idade e o saldo bancário dos clientes. Esses fatores demonstraram grande impacto nas predições de “churn”, o que possibilita a empresa antecipar ações de retenção.
Palavras-chave: Ciência de Dados; Bancos; Aprendizado de Máquina; Evasão de Clientes; Retenção de Clientes.

Prediction of "churn" in fictitious banking company: application of machine learning models

Abstract

In recent years, the use of Machine Learning techniques to predict customer churn has grown significantly, especially in the banking sector, where retaining customers is more cost-effective than acquiring new ones. Based on this premise, this study aimed to apply and compare three predictive Machine Learning models for the churn event — Logistic Regression, XGBoost, and CATBoost — applied to a fictitious bank's dataset. Using the CRISP-DM methodology, the models' performances were evaluated through the ROC AUC and Recall metrics. In the results, the CATBoost model stood out, achieving the highest ROC AUC metric of 0.87 and Recall of 0.88, outperforming the other models. The Permutation Importance method was also applied to identify the most influential variables, highlighting the number of products purchased, customer age, and bank balance. These factors showed a significant impact on churn predictions, enabling the company to anticipate retention actions.
Keywords: Data Science; Banking; Machine Learning; Customer Churn; Customer Retention.

Introdução

O ambiente de negócios, em seus mais diversos segmentos, tem se tornado cada vez mais competitivo. A facilidade com que clientes podem alternar entre concorrentes torna a previsão de tal rotatividade parte importante na tarefa de gerenciar relações com seus respectivos clientes, e por consequência, gerar mais assertividade nas ações de retenção (Zhu, 2017). A evasão de clientes, ou como também pode ser chamado, “churn”, pode causar grandes perdas de receita para as empresas e o aumento do custo operacional, visto que

necessitam aumentar a aquisição de novos clientes para compensar as perdas (Hadden et al., 2007).

O custo de adquirir novos clientes tende a ser cerca de cinco a seis vezes maior do que o custo de reter os que já consomem seus produtos ou serviços. Quanto mais tempo de consumo menos suscetíveis ficam tais clientes a ações de marketing da concorrência. Acerca da necessidade da gestão de retenção de clientes, destacou-se que o crescimento dos resultados das organizações, tomadas como exemplo, passam por minimizar a ocorrência de "churn". As agências bancárias levadas em consideração, mostram que a redução de 5% do índice de evasão de clientes pode elevar os lucros a 85% (Reichheld e Sasser Jr ,1990; Franceschi, 2019).

Em virtude da transformação digital presente cada vez mais em todo mundo, o mercado financeiro também tem sido impactado na forma de relação com os clientes. Essa transformação e a rapidez com que novas tecnologias surgem, faz com que se tenha que repensar constantemente a relação com o cliente, que por sua vez muda de necessidades e até mesmo de desejos, gerando assim uma nova demanda a ser explorada pelos demais players do mercado (Rogers, 2016).

De a cordo com a pesquisa Fintechlab (2020), no Brasil essa concorrência tem crescido com a entrada de novos negócios, e que não tem necessariamente suas origens no mercado financeiro, como é o caso de empresas como "Google", "Amazon" e "Apple" entre outras que somam mais de 770 iniciativas e abrangem praticamente todas as áreas de produtos fornecidos pelos bancos tradicionais. Cerca de 35% dessas empresas não se faziam presentes no relatório do ano anterior, indicando que eram empresas emergentes no setor, sendo resultado de novas oportunidades de melhoria e criação de novos produtos financeiros, influenciados pelos avanços regulatórios como o "Open Finance" e o "Pagamento instantâneo brasileiro" [PIX]. O relatório divulgado "Pulse of Fintech-KPMG" indica que o investimento tem chegado a U$ 113 bilhões no mercado mundial, onde maior parte está alocada em aportes a "fintech's" relacionadas a soluções de pagamentos (KPMG, 2023).

Conforme Jamil e Neves (2000) a era da informação tem transformado as estruturas sociais, econômicas e políticas, onde cada vez mais a informação se torna um dos recursos mais importantes. Para as instituições bancárias, de acordo com a autoridade "antitruste" britânica, quanto mais se sabe sobre seus clientes, mais poder se tem na criação de produtos e serviços personalizados, e, portanto, maior probabilidade de atingir seus clientes com eficiência, além de ter insumo para analisar e descobrir tendências de mercado.

Em todo mundo tem-se visto ações de "Open Banking", cuja finalidade é dar ao cliente final poder de escolha sobre o fluxo de informações relacionados a seus dados financeiros e, com essas escolhas, podem ocorrer mudanças de fidelidade, decorrente de ofertas mais

aderentes por parte de outras empresas. Essa competitividade entre empresas financeiras gera uma busca em entender e solucionar, cada vez mais rápido e de forma mais precisa, as lacunas de compreensão dos clientes e suas características e comportamentos, para tomadas de decisão que evitem o máximo possível eventos de “churn” (Guimarães, 2021).

Nos últimos 5 anos, tem crescido cada vez mais o uso de técnicas de aprendizado de máquina com foco na precisão e agilidade de identificação de possíveis “churn’s”, através da melhoria de modelos ou de novos algoritmos, sejam individuais ou em soluções conjuntas. Tem-se aplicado também técnicas de avaliação, de divisão de dados para treino e teste, cada vez mais avançadas, nos mais diversos segmentos, proporcionando cada vez mais insumos para aprimoramento de tais modelos, aumentando suas capacidades preditivas ao passo que evoluem suas arquiteturas (Kim e Lee, 2022).

As aplicações de aprendizado de máquina na predição de “churn” têm ocorrido nos mais diversos segmentos e trabalhos. Esses modelos têm se destacado como poderosas ferramentas para predição de clientes que podem vir a evadir, permitindo que as instituições identifiquem esses potenciais clientes tempestivamente, bem como suas características e os principais motivos que tendem a levá-lo a tal decisão. É possível analisar exemplos de “cases” de sucesso em diversos segmentos, como o Banco do Brasil por Franceschi (2019), no caso aplicado a uma plataforma online de medicamentos Theodoridis e Tsadiras (2022), e ainda casos aplicados a empresas de telecomunicação como Ullah et al. (2019) e Jain et al. (2020). Todos esses casos evidenciam que cada vez mais, tanto a utilização quanto o impacto positivo em utilizar aprendizado de máquina, vem crescendo e tomando espaço dentro das instituições (Verbeke et al., 2012).

É relevante mencionar que o termo comumente utilizado “churn”, ou também chamado de evasão de clientes, é uma expressão inglesa que pode ser usada para nomear a ação de encerramento de contrato ou vínculo de um determinado cliente com uma instituição, seja tal encerramento afins de mudança para concorrência no presente ou futuramente (Glady, Baesens e Croux, 2009).

A presente pesquisa tem por objetivo aplicar três diferentes modelos de aprendizado de máquina para a predição de "churn" em uma empresa bancária fictícia e, ao final, eleger o modelo com maior capacidade preditiva.

Material e Métodos

Esse projeto utilizou como material uma base de dados pública e anônima que contêm o evento em estudo e as características de clientes expostas em variáveis categóricas e numéricas. Para a construção é utilizada a metodologia CRISP-DM (Wirth e Hipp, 2000).

Criada em 1996, se destacando por conter as etapas de compreensão de negócio, entendimento dos dados, preparação dos dados, modelagem e avaliação. Ainda existe uma última etapa desta metodologia, a de implementação, que foi deixada de fora por não fazer parte do escopo da pesquisa.

Ainda que o presente estudo não seja relacionada a um player real de mercado, os conceitos e práticas de aprendizado de máquina e experiência do usuário que se aplicam ao banco fictício em questão poderiam ser adaptados e implementados em casos reais, possibilitando a melhoria da experiência do usuário com seus produtos e a retenção de clientes com características evasivas. Isso por sua vez possibilita a compreensão e a detecção antecipada de possíveis evasões, instigando a ações de melhoria na qualidade do produto fornecido, oportunidade de crescimento financeiro e sustentabilidade a instituição.

1.1 Compreensão do Negócio

É do interesse de qualquer negócio que seus clientes permaneçam o máximo de tempo possível consumindo seus produtos e serviços, gerando lucros e crescimento para seus negócios, com sinergia entre ambas as partes. Ainda que o caso em questão seja de um banco fictício, é possível através de estudo de casos do mercado e de ferramentas de “Design Thinking”, conceito utilizado mundialmente para abstração de objetivos claros e compreensão de resultados a serem alcançados, como a ferramenta chamada de “Objetivo SMART”. (Drucker, 1954).

Essa ferramenta, conforme mencionado por Lawlor e Hornyak (2012), e chamada de metodologia “SMART” tem como premissa auxiliar na elaboração de objetivos claros a serem alcançados, de forma planejada e bem-sucedida, ou seja, gerar valor através do sucesso e de um caminho claro para alcançá-lo. O “Objetivo SMART” foi utilizado apenas na etapa presente de compreensão do negócio, auxiliando no planejamento claro e objetivo, podendo de forma alguma ser confundido com o objetivo principal do trabalho, sendo este mais amplo. “SMART” pode ser entendido como: “S”: Específico “M”: Mensurável, “A”: Alcançável, “R”: Relevante e “T”: Tempo para realização (Drucker, 1954).

1.2 Entendimento dos dados

Os dados utilizados, conforme mencionado anteriormente, são de um banco financeiro fictício e foram adquiridos no “Kaggle”, plataforma com diversas bases de dados e competições na área de machine learning e ciência de dados. A base de dados contém

informações de supostos clientes correntistas do “Anonymous Multinational Bank”, clientes que saíram e clientes que permanecem consumindo seus produtos.

A base contém 10.000 clientes, com um total de 18 variáveis, sendo essas 17 características e 1 variável a ser predita, a variável por nome de “Exited”; 14 são variáveis numéricas e 4 são categóricas. Nem todas as 18 variáveis foram utilizadas no presente trabalho, pois algumas não tem em si relação que influencie um cliente a “churnear”, como é o exemplo da variável “RowNumber” e “CustomerId” variáveis sem nenhum poder preditivo e que são meramente identificadoras, portanto, serão descartadas posteriormente.

Na Tabela 1 temos um breve dicionário de dados com seus respectivos nomes, descrição e tipo de cada variável original.

Tabela 1. Dicionário de dados: Anonymous Multinational Bank

Nome	Tipo	Descrição
RowNumber	int64	Número do registro (linhas), sem efeito na construção de modelos.
CustomerId	int64	ID do cliente, sem efeito sobre o estudo.
Surname	object	Sobrenome do cliente, sem impacto na análise.
CreditScore	int64	Pontuação de crédito, pode indicar tendência de permanência de clientes com pontuação alta.
Geography	object	Localização do cliente, pode influenciar a decisão de evasão.
Gender	object	Gênero do cliente, possível influência na evasão.
Age	int64	Idade do cliente, clientes mais velhos tendem a permanecer.
Tenure	int64	Anos que o cliente está no banco, clientes novos têm maior chance de evasão.
Balance	float64	Saldo na conta, pessoas com saldos altos são menos propensas a sair.
NumOfProducts	int64	Número de produtos adquiridos pelo cliente.
HasCrCard	int64	Indica se o cliente tem cartão de crédito, clientes com cartão são menos propensos à evasão.
IsActiveMember	int64	Clientes ativos têm menor chance de evasão.
EstimatedSalary	float64	Salário estimado, clientes com salários mais altos tendem a permanecer.
Exited	int64	Indica se o cliente saiu ou não do banco, variável de predição (“churn”).
Complain	int64	Indica se o cliente fez reclamação.
Satisfaction Score	int64	Pontuação de satisfação com a resolução de reclamação.
Card Type	object	Tipo de cartão que o cliente possui.
Points Earned	int64	Pontos ganhos pelo cliente ao usar o cartão de crédito.

Fonte: Resultados originas da pesquisa

De acordo com Favero e Belfiore (2023) a estatística descritiva tem por objetivo sintetizar e descrever características observadas em um conjunto de dados gerando maior

compreensão, sem que de fato se realize qualquer conclusão ou análise inferencial a princípio. A estatística descritiva, independentemente do tipo da variável, seja categórica ou numérica, é parte fundamental da compreensão dos dados.

Na tabela 2 temos um breve resumo estatístico das variáveis numéricas. Notamos que todas as variáveis têm a mesma contagem de valores, descartando a necessidade de tratamento para casos de valores faltantes. Foi notado também que há diferença considerável entre o valor mínimo e máximo das variáveis “EstimatedSalary” e “Balance”.

Tabela 2. Estatísticas Descritivas das variáveis quantitativas relacionadas ao perfil e comportamento financeiro dos clientes

Medidas	EstimatedSalary	Balance	CreditScore	Age	Tenure	Point Earned
Contagem	10.000,0	10.000,0	10.000,0	10.000,0	10.000,0	10.000,0
Média	100.090,2	76.485,9	650,5	38,9	5,0	606,5
Desvio padrão	57.510,5	62.397,4	96,7	10,5	2,9	225,9
Valor mínimo	11,6	0,0	350,0	18,0	0,0	119,0
25º percentil	51.002,1	0,0	584,0	32,0	3,0	410,0
Mediana	100.193,9	97.198,5	652,0	37,0	5,0	605,0
75º percentil	149.388,2	127.644,2	718,0	44,0	7,0	801,0
Valor máximo	199.992,5	250.898,1	850,0	92,0	10,0	1.000,0

Fonte: Resultados originas da pesquisa

Na Figura 1 vemos as frequências absolutas observadas das variáveis categóricas. Foi observado que a variável alvo encontra-se desbalanceada indicando que será necessário procedimento de balanceamento, pois o aprendizado dos modelos pode ser afetado na classe minoritária. Portanto, temos a contagem de clientes por localidade (“Geografhy”), Gênero (“Gender”), por clientes que fizeram ou não reclamações (“Complain”), por clientes que tem ou não cartão de crédito (“HasCrCard”), por clientes ativos ou não (“IsActivMember”), de clientes e seus respectivos tipos de cartão de crédito (“Card Type”) e clientes que deram ou não “churn” (“Exited”).

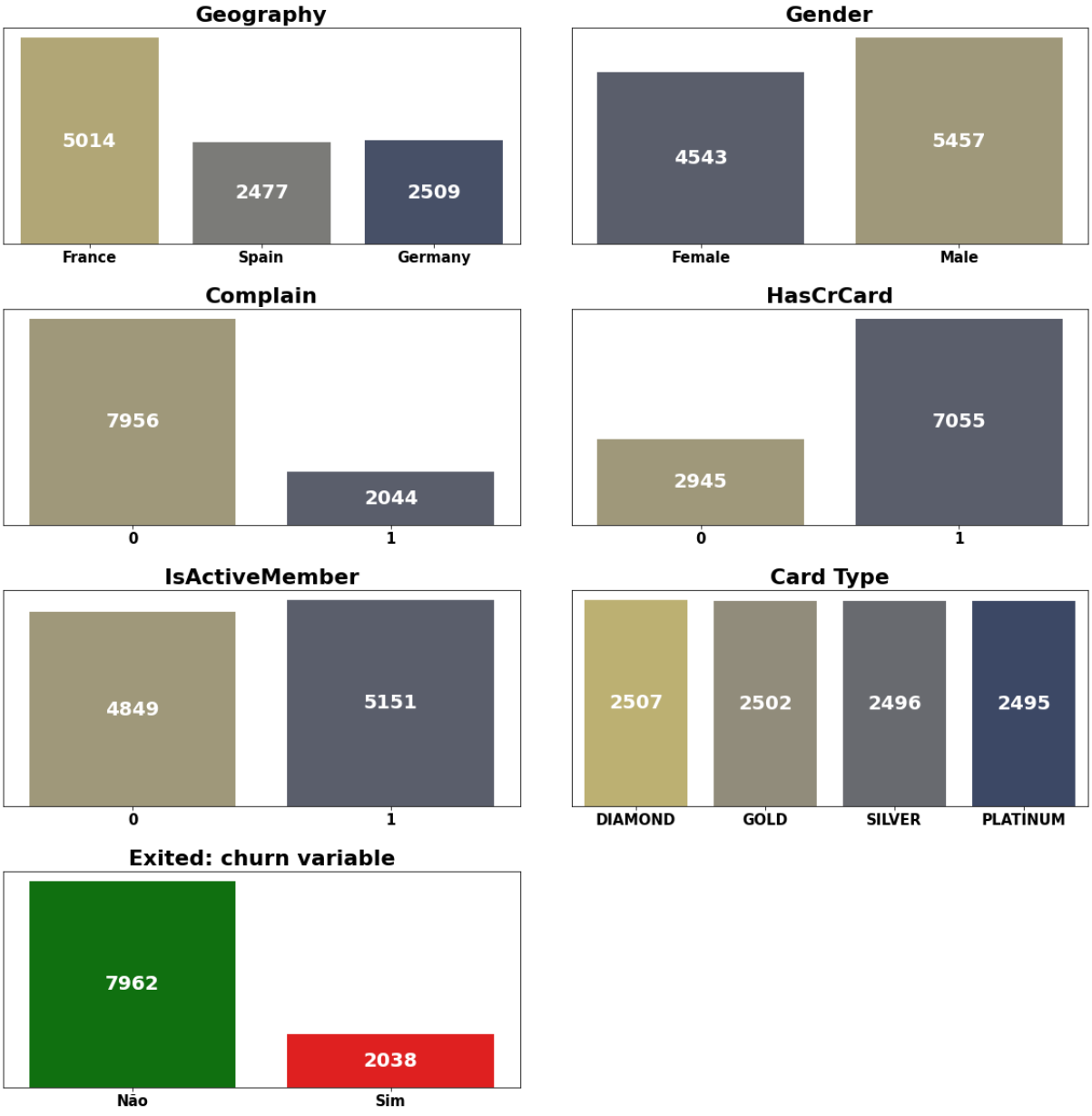

Figura 1. Frequência absoluta das variáveis qualitativas
Fonte: Dados originais da pesquisa

1.3 Preparação dos Dados

Neste projeto a linguagem de programação utilizada foi Python, desde a preparação, elaboração de modelos e geração de gráficos e tabelas, necessárias para análise dos resultados obtidos. Será desenvolvido através da IDE Spyder que por sua vez faz parte do ambiente disponibilizado pelo Anaconda (Anaconda, 2012).

Os pacotes utilizados para preparação dos dados antes de serem disponibilizados para os modelos de aprendizado de máquina foram Pandas por McKinney (2010), Numpy por

Oliphant (2006) e para a visualização dos resultados as bibliotecas Seaborn por Waskom (2012) e Plotly (Plotly Technologies Inc, 2023).

No tratamento dos dados foi constatado que não há valores faltantes para nenhuma das variáveis. Foram descartadas as variáveis “RowNumber”, “CustomerId” e “Surname” por serem meramente identificadoras e não serem úteis na modelagem. Além disso foi identificada forte correlação entre a variável “Complain” e a variável alvo “Exited”.

Após isso foram transformadas no tipo “object” as variáveis “HasCrCard”, “IsActiveMember” e “Exited”. Depois dessas transformações foram contabilizadas seis variáveis categóricas e oito variáveis numéricas. Das seis variáveis categóricas cinco foram submetidas a transformação “One Hot Encoder” Shaikh (2018), essa transformação é importante e necessária para que essas variáveis sejam fornecidas a modelos matemáticos de acordo com Klosterman (2020) e que de maneira alguma sejam fornecidas com valores atribuídos arbitrariamente, prática conhecida como ponderação arbitrária (Favero e Belfiore, 2023).

Depois de realizado procedimento “One Hot Encoder”, que resulta na criação das chamadas variáveis “dummies”, foi analisada a existência de multicolinearidade novamente e foram removidas as variáveis “HasCrCard_0”, “IsActiveMember_0” e “Gender_Female”.

1.4 Modelagem

Na etapa de modelagem aplicou-se três modelos de aprendizado de máquina com configurações focadas em classificação binária. Dois modelos de “boosting” baseados em arvores de decisão: o “XGBoost” por Chen e Guestrin (2016) e o “CATBoost” (Prokhorenkova et al., 2018); e o modelo de Regressão Logística por Fisher (1922), considerado um modelo GLM (“Generalized Linear model”) por Nelder e Wedderburn (1972). Esses modelos foram eleitos em meio a outros pelo fato de serem bem utilizados nos artigos e trabalhos teóricos base dessa pesquisa, e por serem constatados bons preditores nas mesmas. Além disso, foi aplicado um estudo de quais variáveis apresentam maior impacto nos resultados dos respectivos modelos por meio do teste de “Permutation Importance” (Permutation, 2023; Kumar e Pansari, 2016).

O modelo “XGBoost”, que significa "Extreme Gradient Boosting", é baseado em sugestões de otimização de resultados através de um processo sequencial de construção de árvores de decisão. Cada árvore é ajustada para corrigir os erros anteriores, funcionando de maneira aditiva para minimizar a função de perda. Ele incorpora técnicas de regularização e otimização, melhorando tanto a eficiência quanto a precisão preditiva. A regularização

controla a complexidade do modelo, ajudando a prevenir o “overfitting”. Sua forma funcional pode ser vista na eq. (1).

$$L(\theta) = \sum_{i=1}^{n} l(\hat{y}_i, y_i) + \sum_{k=1}^{K} \Omega(f_k) \quad (1)$$

em que, $L(\theta)$ é a função de perda total. Os primeiros termos $\sum_{i=1}^{n} l(\hat{y}_i, y_i)$ são a somatória das perdas (ou erros) para todas as “n” observações do conjunto de dados. Os termos $\sum_{k=1}^{K} \Omega(f_k)$ representam a penalização da complexidade do modelo, em que K é o número total de funções (ou árvores), e $\Omega(f_k)$ é a função de regularização que controla o número de parâmetros do modelo e a profundidade das árvores, influenciando diretamente na capacidade preditiva e no controle de “overfitting” (Chen e Guestrin, 2016; XGBoost Documentation, 2024).

O “CATBoost” que significa “Categorical Boosting” é um algoritmo de aprendizado de máquina eficiente, projetado para lidar com dados categóricos e otimizado para tarefas de classificação e regressão. No presente trabalho, ele é implementado através da classe “CatBoostClassifier”, que adota a função de perda logarítmica (“log loss”), ideal para problemas de classificação binária. A “log loss” mede a diferença entre as probabilidades previstas e os rótulos verdadeiros, avaliando a precisão do modelo. A função objetivo, usada pelo “CATBoost” durante o treinamento, minimiza a função de perda, definida na eq. (2).

$$\mathrm{L}(y, \hat{y}) = -\frac{1}{N} \sum_{i=1}^{N} [y_i \cdot \log(\hat{y}_i) + (1 - y_i) \cdot \log(1 - \hat{y}_i)] \quad (2)$$

em que, $\mathrm{L}(y, \hat{y})$ é a função de perda logarítmica, que mede a precisão entre o valor previsto e o valor real (y_i representa os valores reais e $\hat{y}_i$ os valores previstos). O termo $\log(\hat{y}_i)$ representa a penalização dos valores incorretos para classe positiva. O termo $(1 - y_i)$ representa a classe negativa. E por fim o termo $\log(1 - \hat{y}_i)$ penaliza as previsões incorretas para a classe negativa. (Prokhorenkova et al., 2018; Yandex LLC, 2024).

O modelo “Logístico” ou de Regressão Logística, aplicado neste presente trabalho, através da biblioteca “scikit-learn”, é um modelo usado para classificar observações em dois grupos. Ele modela a relação entre uma variável dependente binária e um conjunto de variáveis independentes, estimando a probabilidade de que uma observação pertença a classe positiva $P(yi = 1 \mid Xi)$. No modelo de regressão logística binária, a equação para a probabilidade é dada por: eq. (3).

$$\hat{p}(X_i) = \frac{1}{1 + \exp\left(-(X_i w + w_0)\right)} \tag{3}$$

em que, $\hat{p}(X_i)$ é a função de probabilidade. O termo w representa os coeficientes (pesos) do modelo, que determinam a importância de cada variável. O termo w_0 é o intercepto ou termo de bias do modelo, que ajusta a função para que ela se adapte melhor aos dados. O termo exp é a função exponencial. Com a determinação da probabilidade, então tem-se a função objetivo do modelo, que busca classificar os dados em duas categorias, minimizando a função de custo. eq. (4).

$$\min_w \quad \frac{1}{S}\sum_{i=1}^{n} s_i\left(-y_i \log\left(\hat{p}(X_i)\right) - (1 - y_i)\log\left(1 - \hat{p}(X_i)\right)\right) + \frac{r(w)}{SC} \tag{4}$$

em que, $\min_w$ é a função que visa minimizar a função de custo. Os termos $\frac{1}{S}\sum_{i=1}^{n} s_i\left(-y_i \log\left(\hat{p}(X_i)\right) - (1 - y_i)\log\left(1 - \hat{p}(X_i)\right)\right)$ são a função de perda logarítmica. Os termos $\frac{r(w)}{SC}$ correspondem a regularização, que é fundamental para evitar o "overfitting", permitindo um melhor desempenho em dados não vistos (Pedregosa et al., 2011).

O momento de treinamento e teste dos modelos foi realizado com base na adaptação de Rodrigues (2023) a técnica aplicada por (Deotte, 2020). Foi separada a base original em duas partes, a base de treino com 80% dos dados e a base de teste com os outros 20%.

Foi realizado um teste de multicolinearidade nos dados de treino a fim de que após a criação das variáveis "dummies" as variáveis que apresentassem correlação alta fossem tiradas do modelo, tanto na base de treino quanto na base de teste. Foi também realizada uma análise dos "outliers" das variáveis e foi necessário aplicar um procedimento chamado de truncamento ou "winsorization**".** (Dixon e Tukey, 1968). que consiste em substituir os outliers pelo valor do limite superior quando for acima e pelo limite inferior quando for abaixo dos valores da distribuição da variável. Isso foi necessário somente para as variáveis "Age", "CreditScore" e "NumOfProducts" e foi aplicado no ambiente de treino e teste.

Nessa etapa de treinamento foram realizados procedimento de "cross-validation" por Stone (1974) e uma aplicação por "k-folds" e cenários Rodrigues (2023) e Deotte (2020). A base de dados foi dividia em seis grupos, onde pode ser formado uma espécie de comissão de modelos. Foram gerados seis modelos onde cada modelo recebeu uma das partes das bases divididas (uma para treino e validação e outra para teste), de forma que realizou o treinamento em 5 e o validação em 1. Logo em seguida o foi realizado o teste na base de

dados para teste recebida. Para o treinamento houve a necessidade de balanceamento das classes da variável alvo pela técnica de "undersampling", após isso foi realizada uma normalização através da técnica de “Z-score” para garantir uma escala consistente nas variáveis quantitativas e um melhor desempenho dos modelos. O “Z-Score” mede a distância em desvios padrão entre uma observação e a média, expresso pela equação eq. (5)

$$Z = \frac{(X - \mu)}{\sigma} \tag{5}$$

em que, Z é a função de normalização, que através da distância entre desvios padrão das observações e a média, normaliza dos dados de dada amostra. O termo X é o valor da observação da amostra. O termo μ é a média da amostra a partir da qual X foi extraído. E por fim o termo σ é o desvio padrão da amostra (Field, 2024).

Depois do treinamento nas bases separadas para treino, os modelos então foram aplicados nas bases de dados de teste, isso para cada um dos seis respectivos grupos. A média das probabilidades geradas pelos seis grupos foi considerada como a predição final da classe de modelo aplicada. Outrossim aplicar os modelos a base de teste foi extremamente útil para avaliar como desempenham em dados inexplorados. Os dados do conjunto de teste também foram submetidos à normalização antes de sua aplicação nos modelos.

A Figura 2 mostra como foram aplicadas as separações de “folds” isso para todos os três modelos, “XGBoost”, “CATBoost” e Regressão Logística; de maneira a totalizar 18 modelos. As aplicações foram feitas com “seed” padrão em 42 para garantir uma reprodutividade.

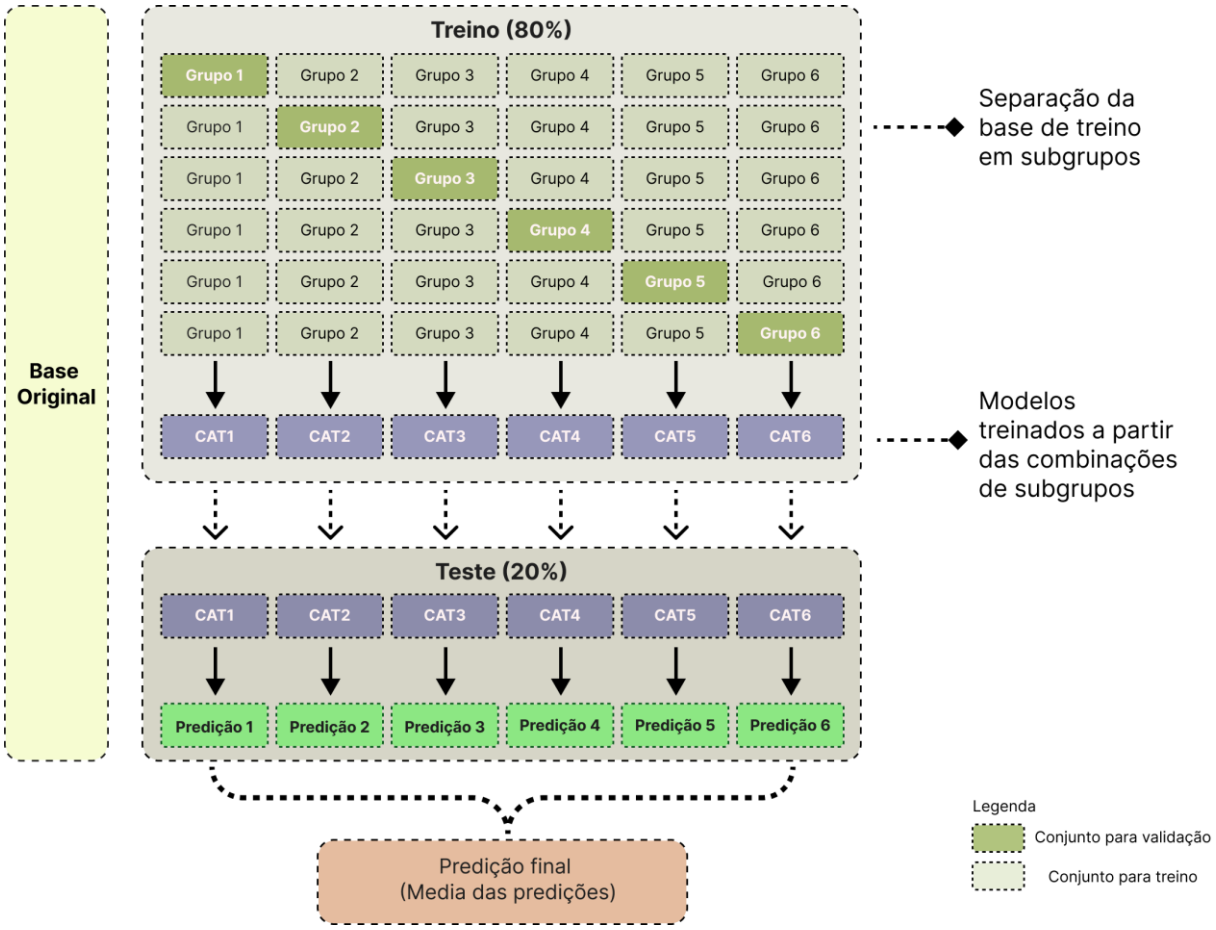

Figura 2. Fluxograma de Execução tomando como exemplo o modelo “CATboost”
Fonte: Dados originais da pesquisa

O presente estudo foi realizado em três cenários, a fim de encontrar as melhores configurações de hiperparâmetros dos modelos e maximizar a capacidade preditiva deles. No Cenário 1 foi realizado o experimento de acordo com o fluxo já citado e com as configurações padrão dos algoritmos dos modelos.

Para o Cenário 2 foi utilizado o Optuna criado por Akiba et al. (2019) a fim de encontrar os melhores hiperparâmetros para os modelos citados. Esse procedimento foi aplicado visando maximizar a métrica de desempenho “F1-Score”.

Ao fim, no Cenário 3 foi realizado o procedimento “Permutation Importance” a fim de identificar e selecionar as melhores variáveis para os modelos, isso com as configurações e hiperparâmetros maximizados pelo Cenário 2. Essa técnica possibilita a remoção das variáveis com baixa contribuição aos modelos, variáveis com valor menor que zero foram removidas, pois não contribuem para os resultados dos modelos.

Embora tenha sido usado como exemplo no gráfico acima apenas um modelo, vale ressaltar que os três modelos em estudo nessa pesquisa passaram pelos mesmos

procedimentos já citados. Para todos os cenários o “cutoff” (ou “threshold”) foi mantido em 0,5 de maneira que, as probabilidades geradas pelos modelos para cada observação que ultrapassassem esse valor foram consideradas como eventos de “churn”.

1.5 Avaliação

Foram aplicados cinco métricas para avaliação dos modelos. Essas métricas foram importantes nas construções tanto no momento de treinamento, na base de treino ou mais precisamente nos resultados dos “folds”, quanto nos valores médios das probabilidades da base de teste para cada modelo.

“ROC AUC”: a métrica “AUC”, é derivada da curva “ROC” (“Receiver Operating Characteristic”). Essa métrica nos permite, com apenas um valor, mensurarmos o desempenho do modelo classificador, pois ela nos fornece o valor (entre 0 e 1) da área sob a curva “ROC”, quanto mais próximo de 1, mais eficaz é o classificador (Bruce e Bruce, 2019).

Precisão (“Precision”): mensura a proporção de previsões positivas corretas feitas pelo modelo, ou seja, o quanto tal modelo está acertando ao prever clientes que realmente são eventos “churn” (Bruce e Bruce, 2019).

Revocação (“Recall”): mensura a capacidade do modelo de identificar corretamente todos os eventos positivos. Em outras palavras, é a proporção de “churn’s” verdadeiros que o modelo conseguiu identificar em relação ao total de “churn’s” reais. (Bruce e Bruce, 2019).

Acurácia (“Accuracy”): Porcentagem de classificações corretas, usada para saber quantas observações foram preditas corretamente pelo modelo. De acordo com Harrison (2019), essa métrica não é um bom termômetro de modelos para predições com classes raras, porém não é o caso do presente estudo.

“F1-Score”: é média harmônica entre Precisão e a Revocação (Harrison,2019).

Resultados e Discussão

1.1 Compreensão do negócio

Com a aplicação da abordagem SMART proposta por Drucker (1954) e por Lawlor e Hornyak (2012), foi possível determinar os seguintes critérios:

S - “Specific" (Específico): Investigar e comparar três modelos de predição de evasão (“churn”) e ao final escolher o melhor modelo (de acordo com as métricas de desempenho abordadas) utilizando os dados dos clientes correntistas, para prever possíveis cancelamentos com base em seus comportamentos observados.

M - “Measurable" (Mensurável): O êxito do projeto pode ser avaliado pelo nível de precisão do modelo conforme a literatura, considerando um "ROC AUC" superior a 0,50 e um “Recall” igual ou superior a 0,70.

"Attainable" (Alcançável):Realizar a construção e a avaliação de três modelos de aprendizado de máquina, a fim de prever potenciais “churn’s” e identificar as variáveis que mais impactam o desempenho do modelo escolhido com melhor “AUC ROC” e “Recall” (nessa mesma ordem).

R- "Relevant" (Relevante): Detectar antecipadamente possíveis clientes aderentes a evasão (“churn”) com o objetivo de guiar, ou servir de ponto inicial para ações estratégicas de retenção de clientes.

T- "Time-based" (Tempo para realização): Tempo de conceituação, construção e avaliação de 7 meses.

A abordagem SMART de maneira alguma deve ser confundida com os objetivos dessa pesquisa, pois ela foi importante apenas para ajudar no desenvolvimento, que por sua vez pode ser mais claro, regrado e com fases e problema bem esclarecidos

1.2 Entendimento dos dados

Na fase de entende entendimento dos dados foram realizadas investigações de valores faltantes, entendimento das proporções das classes das variáveis explicativas e da variável alvo desse estudo. Foi constatado que a variável alvo estava desbalanceada, ou seja, o evento em estudo (“Exited”) contém naturalmente mais valores de não “churn” (representado por 0) do que observações de “churn” (representados pelo valor 1). A proporção propriamente dita é que de 10.000 observações, tamanho total da base de dados em estudo, 7.962 (79,62%) são clientes não “churn” e 2.038 (20,38%) são clientes “churn”. Também foi constatado desbalanceamento nas variáveis “Complain”, com 80% para clientes que não realizaram reclamação e 20% que realizaram reclamação; e “HasCrCard” com 70% de clientes com cartão de crédito contra 30% de clientes sem cartão de crédito.

1.3 Preparação dos dados

Após os tratamentos dos dados descritos no item 1.3 Preparação dos dados em Material e Métodos na base “Anonymous Multinational Bank”, foi detectada forte correlação entre a variável "complain" e a variável alvo "Exited", portanto ela também foi removida a fim de evitar multicolinearidade nos modelos. Com esses procedimentos ficaram ao todo 19 variáveis, sendo uma delas a variável a ser predita. Não houve alteração no tamanho da

amostra no que diz respeito a quantidade de observações, visto que não se encontrou necessidade de remoção de valores faltantes em nenhuma das variáveis.

1.4 Modelagem

No momento de modelagem dos dados foi utilizada uma separação em treino e teste na base contendo 10.000 registros únicos já preparados. Foi separado com de proporção 80% para treinamento e 20% para teste. A separação foi realizada de forma aleatória e estratificada, visando manter as proporções de eventos de evasão de clientes (“churn”), conforme mencionado tanto na sessão de entendimento dos dados quanto no resultado e discussão.

As proporções mencionadas anteriormente podem ser vistas na Figura 3. Por ela podemos ver a separação de 80% da base original para treino e 20% para teste (soma dos valores das barras) e podemos notar o desbalanceamento natural que temos entre eventos de “churn” (1) e “não churn” (0). Para que isso não prejudicasse o aprendizado dos algoritmos foi aplicado balanceamento nos dados de treino.

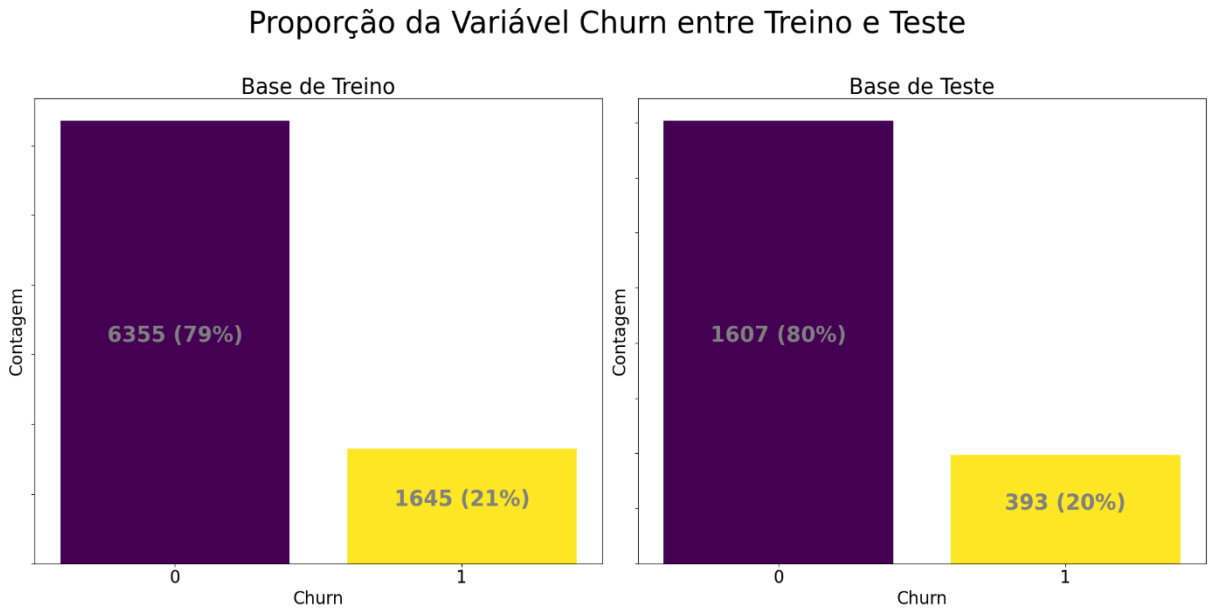

Figura 3. Gráfico representativo das proporções das bases de treino e teste e da evidência de desbalanceamento da variável alvo.
Fonte: Dados originais da pesquisa

Também foi realizada uma análise de Correlação de Person na base de dados de treino, para verificar a existência de multicolinearidade, que pode prejudicar o desempenho

do algoritmo, conforme já identificado anteriormente, a variável “Complain” apresentou forte correlação com a variável alvo, portanto ela foi removida. Permanecendo estão as variáveis a seguir na Figura 4. Nota-se que nenhuma delas apresentou correlação alta o suficiente para ser considerada perigosa ao desempenho do modelo.

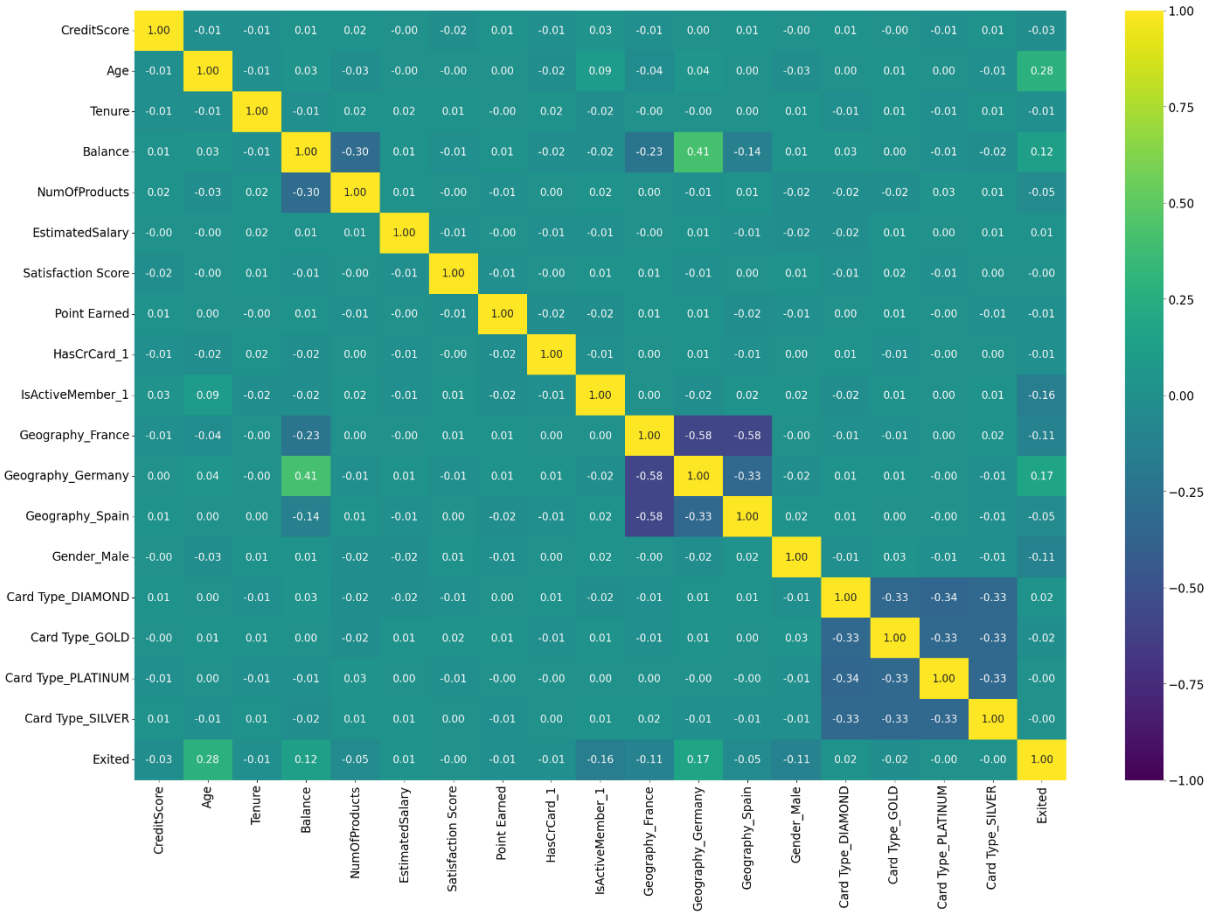

Figura 4. Matriz de Correlação de Person das 19 variáveis na base de treino através do gráfico de mapa de calor
Fonte: Dados originais da pesquisa

No desenvolvimento desta etapa constatou-se também a presença de alguns “outliers” em algumas variáveis: “CreditScore”, “Age” e “NumOfProducts”. Para tratar desses valores foi utilizada a técnica de “winsorization” onde valores abaixo do limite inferior (Q1 - 1.5 * IQR) são substituídos pelo limite inferior e os valores acima do limite superior (Q3 + 1.5 * IQR) são substituídos pelo limite superior; onde Q1 corresponde ao 25 percentil, Q3 ao 75 percentil e IQR é a amplitude interquartílica (Q3 -Q1). Através desse procedimento aplicado a base de treinamento, foi possível, conforme podemos ver comparando a Figura 5 e a Figura 6, que os outliers foram transformados em valores dentro da faixa aceitável, sem que fosse necessário removê-los. Assim a quantidade de observações pode ser preservada no estudo.

Na Figura 5 podemos notar a presença desses “outliers” bem como as demais variáveis numéricas que não apresentaram “outliers”.

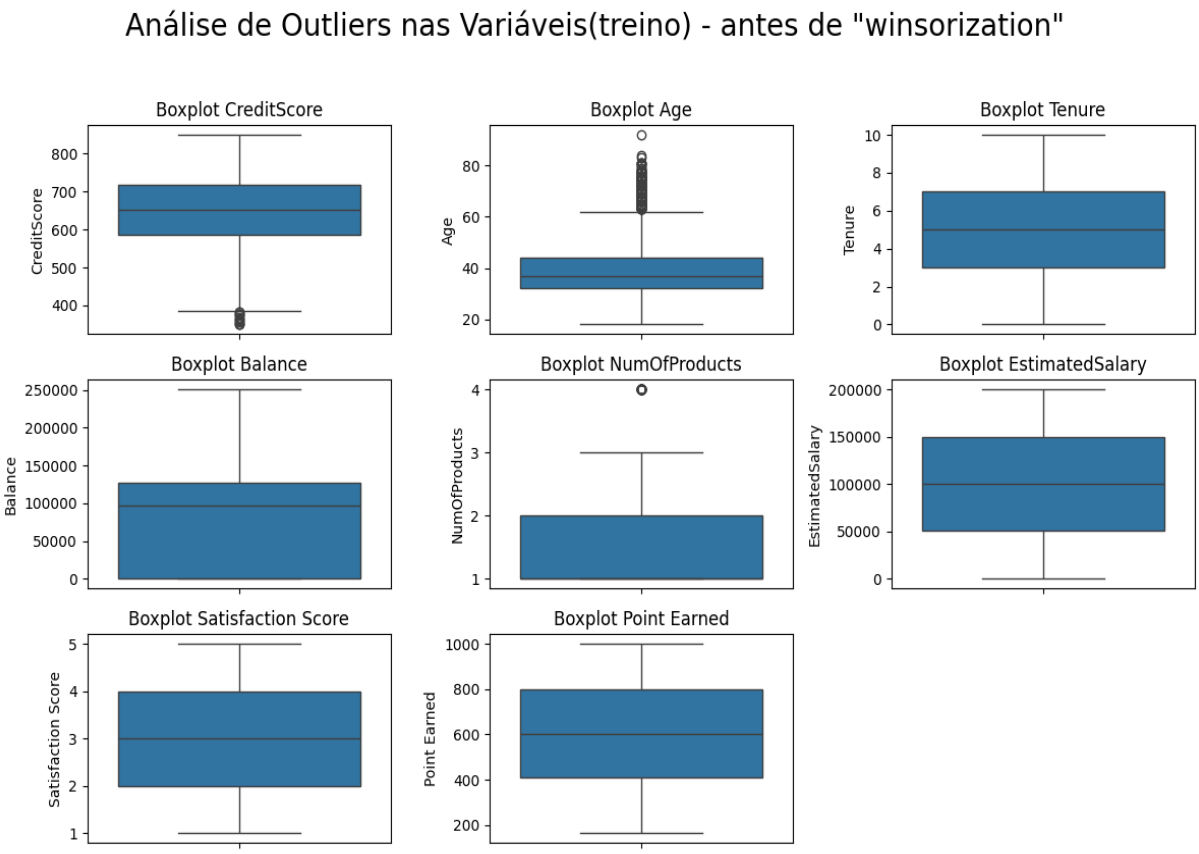

Figura 5. Gráfico de “Boxplot” para detecção de presença de “outliers” na base de treino
Fonte: Dados originais da pesquisa

Na Figura 6 podemos ver que os “outliers” foram perfeitamente tratados conforme a técnica mencionada e que agora se assemelham as demais variáveis numéricas que não necessitaram do procedimento.

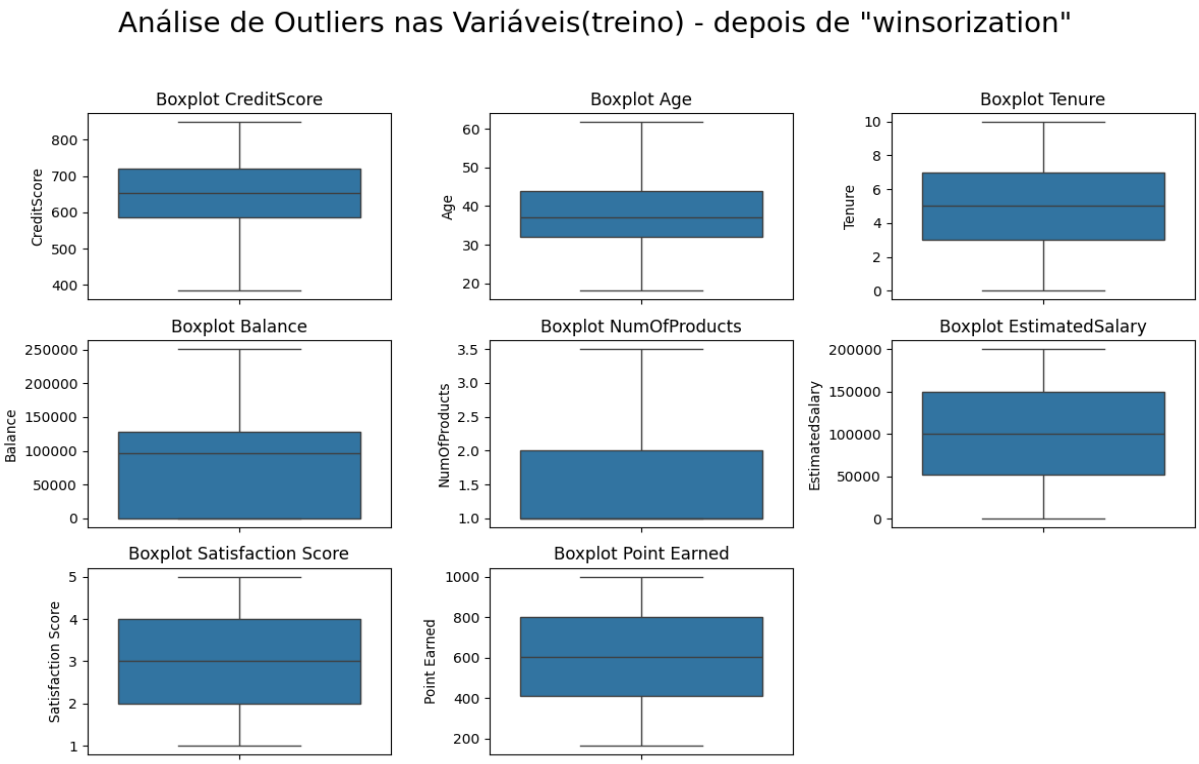

Figura 6. Gráfico de “Boxplot” com os “outliers” já tratados com “winsorization”
Fonte: Dados originais da pesquisa

1.4.1. Resultados do Cenário 1

No Cenário 1, conforme mencionado anteriormente, foi realizado procedimento com todos três modelos em comparação, em suas configurações originais sem qualquer alteração de hiper parâmetros.Foi alterado somente o parâmetro de avaliação chamado “scoring” no processo de validação cruzada para “f1”, que significa que ele utilizou a métrica “F1-Score” nas avaliações dos “folds”. Isso foi feito por que no momento de aplicação constatou-se um “Recall” (Revocação) excelente, porém um “Precision” (precisão) abaixo de 0.45 para todos os três modelos, “Logístico”, “XGboost” e “CATboost”. Por essa razão, a fim de ter um equilíbrio um pouco mais harmônico no momento de treinamento e teste dos modelos, foi feita essa configuração que se estendeu para todos outros modelos e cenários.

Todos três modelos apresentaram resultados razoáveis em termos globais de desempenho, mais precisamente descrito pela métrica “ROC AUC”, mantendo-se todos acima de 0.77, resultado considerado bom de acordo com as pesquisas e estudos base deste

trabalho. Conforme podemos ver na Figura 7, que representa os resultados na base de teste dos modelos, houve um resultado muito próximo entre os modelos “XGboost” e “CATboost”, já dando indícios de serem bons modelos para esse estudo.

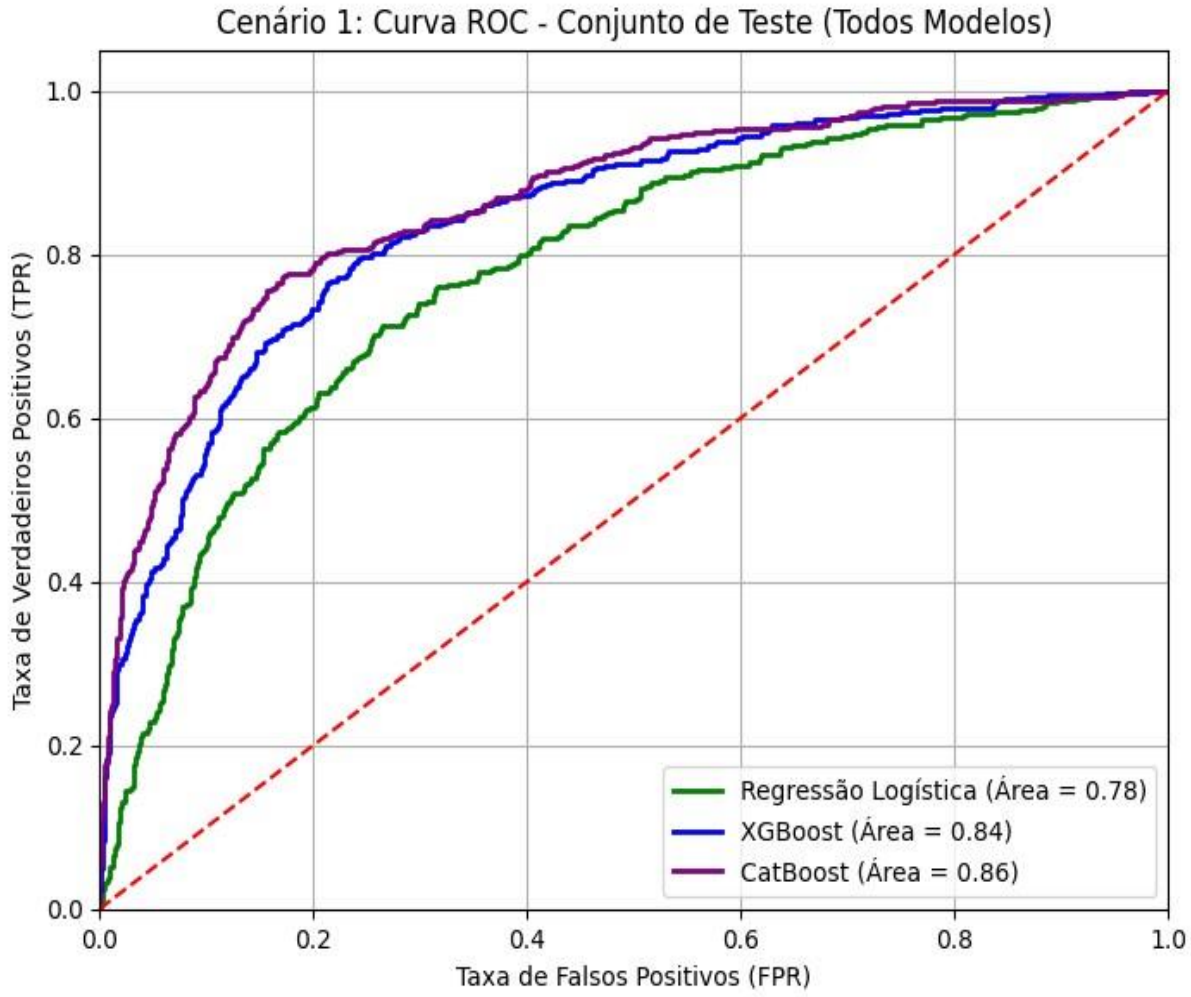

Figura 7. Gráfico de “Curva ROC” que mostra a curva e a área sob a curva “ROC” (AUC) dos modelos “Logístico”, “XGboost” e “CATboost” na base de teste no Cenário1
Fonte: Dados originais da pesquisa

Ainda sobre o desempenho dos modelos, podemos ver outras métricas de desempenho dos modelos, tais como “Recall” e “F1-Score” já mencionadas anteriormente. Como podemos observar com esses resultados, os modelos nas configurações iniciais têm bons resultados, com valores bons em termos globais conforme “ROC AUC”; bons valores de “Recall” significando que os modelos detectam bem os verdadeiros positivos ou “TP” e evitam os “FN” ou falsos negativos (Recall = TP/(TP+FN)). Porém vemos que os valores de Precisão do modelo são consideravelmente baixos, indicando uma dificuldade em identificar os valores

positivos realmente pertencentes a classe positiva, ou seja, quantos são “TP” (verdadeiros positivos ao invés de “FP” (falsos positivos). A expressão se dá por: Precisão = TP / (TP+FP).

Essa diferença entre essas duas métricas leva consequentemente a um “F1-Score” médio, pois ele é a média harmônica entres Precisão e “Recall” (F1-Score= 2*((Precision*Recall)/(Precision+Recall)), sendo um indicador muito importante em casos de desbalanceamento de classes. Conforme mencionado, isso levou a considerarmos o “F1-Score” como a medida de desempenho dentro dos “folds” de treinamento da validação cruzada, a fim de termos mais equilíbrio entre esses indicadores, o que gerou uma melhoria de 0.20 nesse indicador em relação ao que era anteriormente.

Ainda que os indicadores que serão utilizados nessa pesquisa, a fim de escolher o melhor modelo sejam “Recall” e “ROC AUC”, também, além dos já mencionados, consideraremos a Acurácia desses modelos. No presente cenário ela se mostrou com bons valores para os três modelos, com resultados acima de 0.70 indicando que os modelos têm boa capacidade em detectar tantos valores verdadeiramente positivos quanto verdadeiramente negativos, em outras palavras, quão frequentemente o modelo acerta suas predições. Todos os indicadores citados podem ser observados na Tabela 3.

Tabela 3. Métricas de desempenho dos modelos na base de teste no Cenário1

Métricas	Logístico	XGBoost	CATBoost
ROC AUC:	0.7821	0.8436	0.8643
Precisão:	0.3820	0.4550	0.4904
Revocação(Recall):	0.7125	0.7710	0.7812
F1-Score:	0.4973	0.5722	0.6026
Acurácia:	0.7170	0.7735	0.7975

Fonte: Resultados originas da pesquisa

Ainda que os resultados na base de teste tenham sido bons, é sempre muito importante também avaliar como se comporta os modelos no momento de treinamento. Para isso consideraremos somente o indicador de “ROC AUC” conforme a Figura 8. Como podemos observar, o desempenho da base de treinamento é muito próximo do desempenho na base de teste.

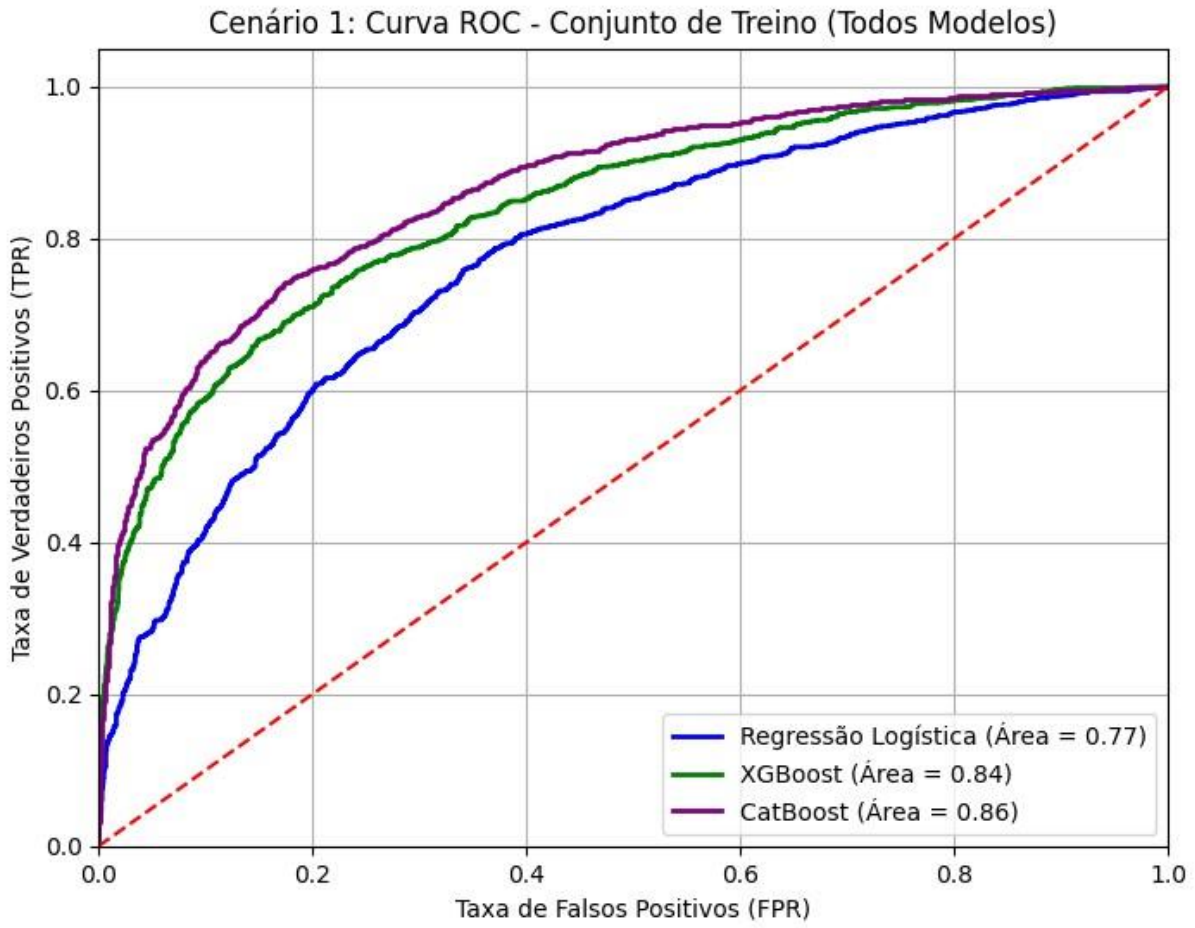

Figura 8. Gráfico de “Curva ROC” que mostra a curva e a área sob a curva “ROC” (AUC) dos modelos “Logístico”, “XGboost” e “CATboost” na base de treinamento no Cenário1
Fonte: Dados originais da pesquisa

Também foi realizado um teste de “Permutation importance” na base de teste, a fim de monitorarmos as influências das variáveis neste cenário e modelos, ainda que ele não seja levado em consideração para nenhuma alteração de estrutura dos modelos ainda, é importante monitorarmos, já no primeiro cenário como as variáveis respectivamente influenciam nos resultados. Conforme podemos ver na Figura 9, a exemplo do modelo que melhor desempenhou neste cenário, o “CATBoost”, foi observado que as variáveis “Age”, “NumOfProducts”, “IsActiveMember_1”,” Balance” e “Geography_Germany” são variáveis que tem significante impacto nos resultados do modelo quando seus valores são alterados durante o teste. Por outro lado, as variáveis “Geography_France” e “Tenure” tem leve impacto negativo nos resultados do modelo.

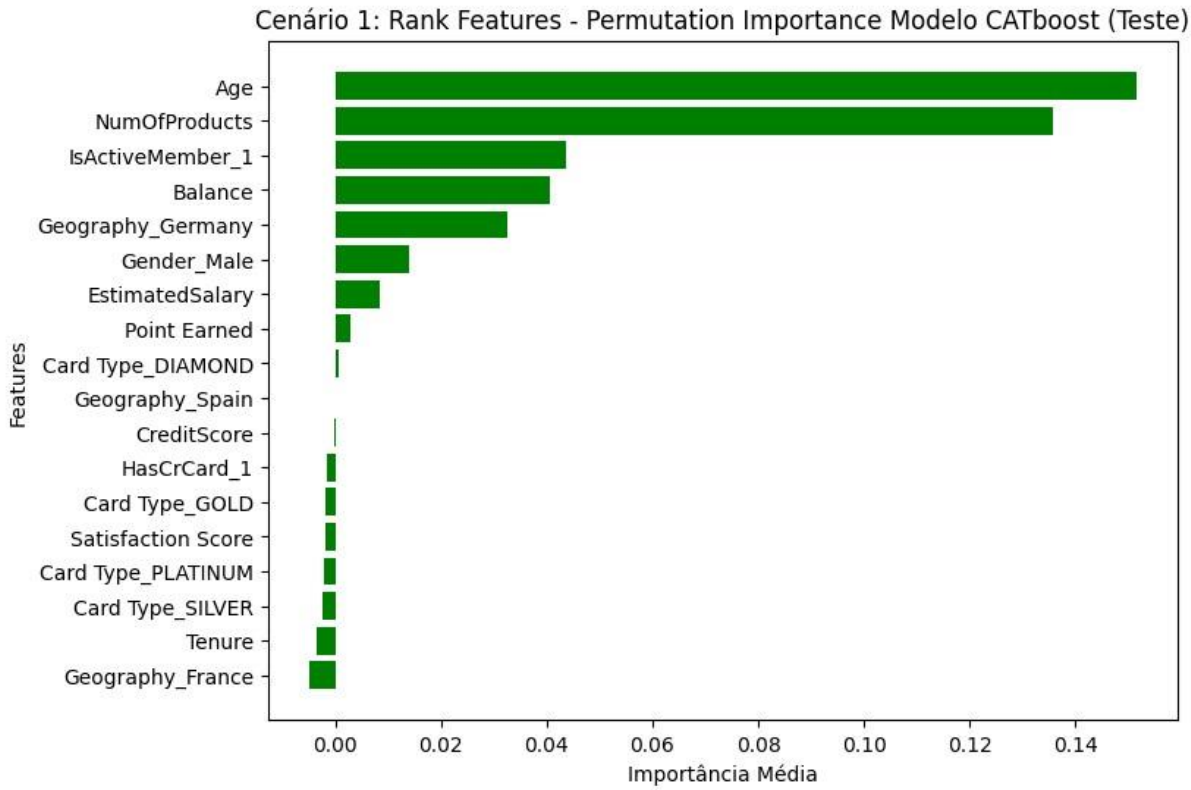

Figura 9. Gráfico de “Permutation Importance” do modelo “CATboost” na base de teste, usado como exemplo para analisar a importância das variáveis nos modelos no Cenário1
Fonte: Dados originais da pesquisa

1.4.2. Resultados do Cenário 2

No cenário 2, foi realizada a busca pelos hiperparâmetros ideais para cada um dos três modelos, de acordo com suas respectivas arquiteturas e parâmetros configuráveis, a fim de otimizarmos suas capacidades preditivas. Isso foi possível através do uso do Optuna, de maneira que para cada modelo, objetivando um ajuste melhor de “F1Score”, foram parametrizadas 500 tentativas. Na Figura 10 podemos ver como isso impactou cada um dos 3 modelos em termos de “ROC AUC”. Vemos que o modelo “Logistico” e “CATboost” tiveram um leve aumento de desempenho, enquanto o modelo “XGboost” melhorou um pouco mais, de maneira a empatar com o modelo “CATboost” em termos de “ROC AUC”.

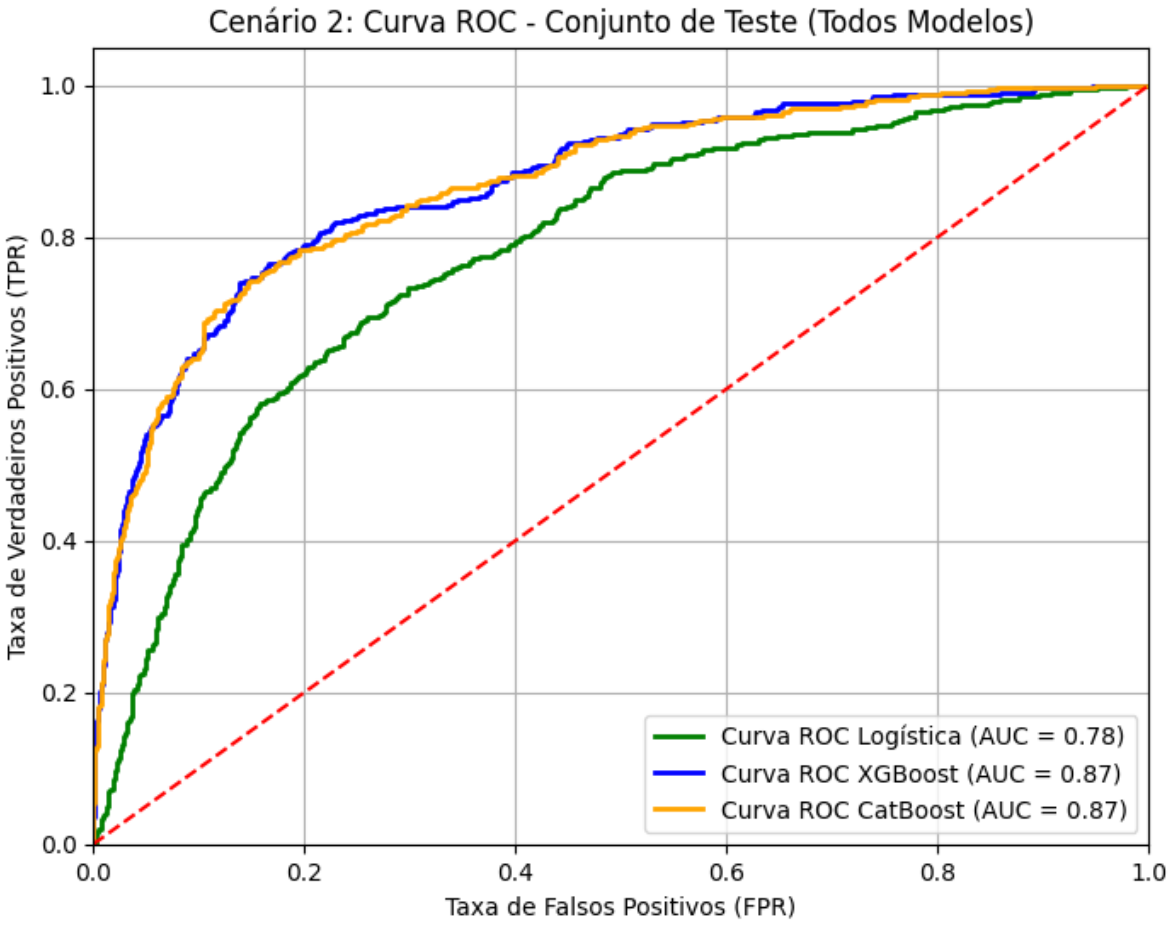

Figura 10. Gráfico de “Curva ROC” que mostra a curva e a área sob a curva “ROC” (AUC) dos modelos “Logístico”, “XGboost” e “CATboost” na base de teste no Cenário2
Fonte: Dados originais da pesquisa

Já pelos demais indicadores, é possível notar que o modelo “XGboost” se sobressaiu em relação aos demais em todas as métricas de desempenho, especialmente nas que serão usadas para decidir o melhor modelo, “ROC AUC” e “Recall” se mostrando um melhor preditor em termos de identificar eventos de “churn”, porém ainda com um índice muito alto de falsos positivos, ou seja, pela métrica “Precisão” vemos um valor baixo indicando que os modelos predizem muitos não “churn” como eventos de “churn”, e isso mesmo com os modelos parametrizados para buscar um equilíbrio de métrica “F1-Score”.

Vale ressaltar que todas as métricas são importantes dependendo do objetivo e problema a ser solucionado, em casos que a prioridade é acertar o máximo de eventos positivos e o custo de falsos positivos é baixo para o negócio, então busca-se mais um “Recall” elevado, já quando esse custo é alto então observa-se mais o equilíbrio entre “Precisão” e “Recall” através do “F1-Score”. Na maioria dos cenários ainda se observa a “ROC AUC”

independente dos objetivos, por ela dar sempre uma ideia geral dos resultados dos modelos e ser de fácil interpretação gráfica. As métricas citadas podem ser analisadas na Tabela 4.

Tabela 4. Métricas de desempenho dos modelos na base de teste no Cenário2

Métricas	Logístico	XGBoost	CATBoost
ROC AUC:	0.7814	0.8691	0.8676
Precisão:	0.3797	0.4613	0.4374
Revocação(Recall):	0.7226	0.8193	0.8092
F1-Score:	0.4978	0.5903	0.5679
Acurácia:	0.7135	0.7765	0.7580

Fonte: Resultados originas da pesquisa

1.4.3. Resultados do Cenário 3

Para o terceiro e último cenário, foi aplicado o teste de “Permutation Importance” em todos três modelos, a fim de removermos as variáveis que não contribuem para os resultados dos modelos positivamente. Os resultados em termos gerais de “ROC AUC” podem ser vistos na Figura 11. Há um empate considerando o critério de arredondamento gráfico entre os modelos “XGboost” e “CATboost”, indicando que após o procedimento de “Permutation Importance” os dois mantiveram seus desempenhos globais preditivos.

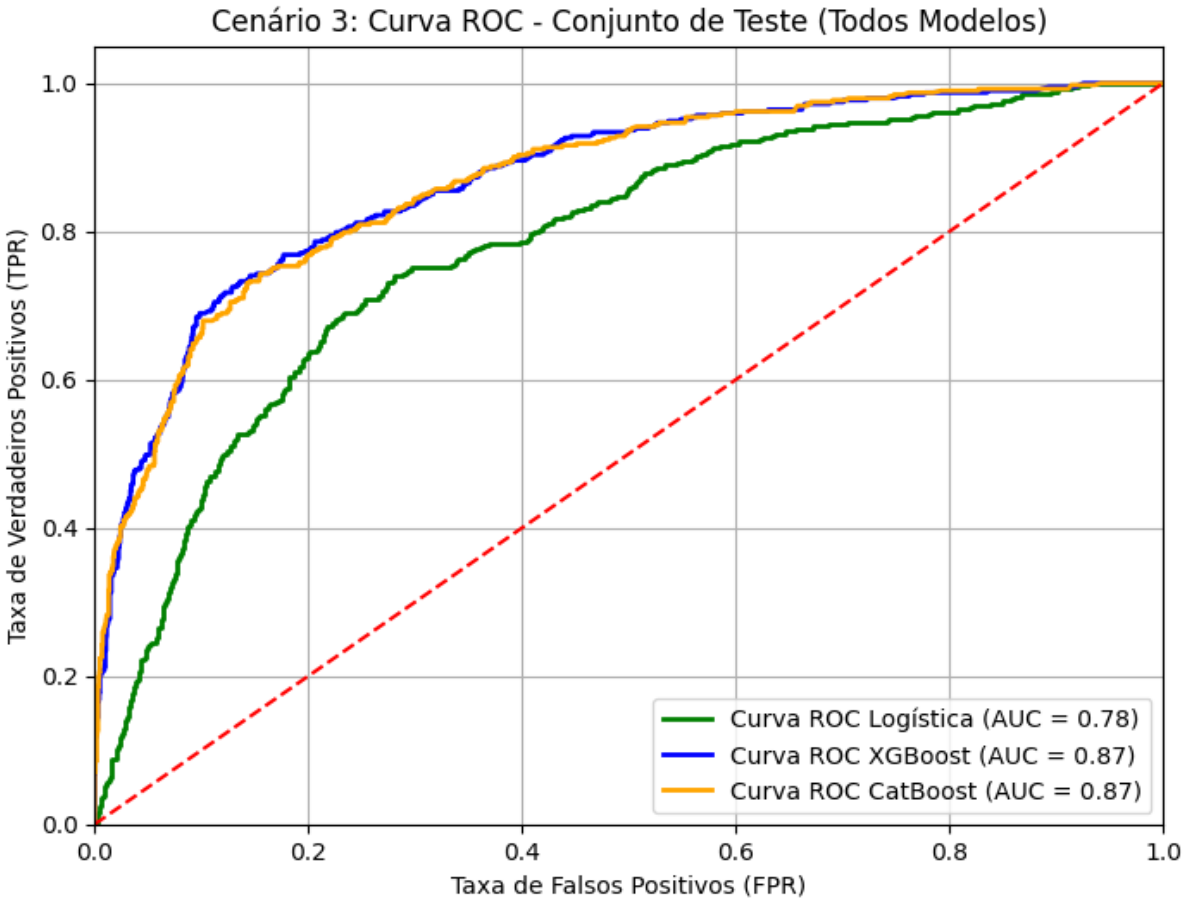

Figura 11. Gráfico de “Curva ROC” que mostra a curva e a área sob a curva “ROC” (AUC) dos modelos “Logístico”, “XGboost” e “CATboost” na base de teste no Cenário3
Fonte: Dados originais da pesquisa

Podemos observar na Tabela 5 que os modelos “CATboost” e “XGboost” tem valores numéricos diferentes na métricas “ROC AUC”, mas como mencionado anteriormente, são considerados empatados por critérios de arredondamento. Porém os valores na métrica “Recall” são consideravelmente diferentes, tornando o modelo “CATboost” mais eficiente em detectar casos reais de “churn”. Além disso, nas demais métricas, também importantes porém não decisivas nesta pesquisa, vemos que os resultados são relativamente parecidos entre os três modelos, sendo o “XGboost” um pouco melhor que os demais.

Tabela 5. Métricas de desempenho dos modelos na base de teste no Cenário3

Métricas	Logístico	XGBoost	CATBoost
ROC AUC:	0.7813	0.8700	0.8680
Precisão:	0.3901	0.4328	0.3724
Revocação(Recall):	0.7405	0.8193	0.8804
F1-Score:	0.5110	0.5664	0.5234
Acurácia:	0.7215	0.7535	0.6850

Fonte: Resultados originas da pesquisa

Com o intuito de entender quais tipos de variáveis costumam interferir no evento em estudo, foram analisadas outras pesquisas, em outros cenários também relacionados a predição de “churn” com modelos de aprendizado de máquina. Foram levantados alguns, ainda que não diretamente relacionados a empresas bancárias, como nos casos aplicados a empresas de telecomunicação. Em pesquisa realizada por Rodrigues (2023), que expõe que o levantamento de principais variáveis pode não ser igual, mesmo em cenários extremamente parecidos, pois dependem de fatores como a base estudada, técnicas e práticas de tratamento de dados, configurações dos modelos de aprendizado de máquina e diversos outros fatores que podem interferir em como e quanto essas variáveis vão ser importantes em termos preditivos (Hanif, 2019; Pamina et al., 2019).

Já em cenários bancários parecidos, como visto na predição de “churn” em clientes de cartão de crédito, foi possível observar que variáveis como “Total_Trans_Ct” que é a contagem de transações nos últimos 12 meses, e que se assemelha com “IsActiveMember” desta pesquisa, tem forte importância; a variável “Total_Revolving_Bal” que descreve o saldo total do cartão de credito, à semelhança de “Balance” saldo em conta corrente; “Total_Relationship_Count” que descreve o numero de produtos que o cliente tem e se assemelha a “NumOfProducts”. Todas apresentam forte importância, isso mostra que em alguns cenários, mesmo com diferenças, algumas variáveis podem apresentar semelhança e se mostrarem importantes mesmo em produtos diferentes, e mesmo segmento de negócio. Isso por sua vez ainda não garante que sejam as mesmas importâncias, visto que dependem de diversos fatores (Wu & Wang, 2022).

Na Figura 12 podemos ver as variáveis citadas e as demais variáveis mais importantes após aplicação após aplicação da técnica de “Permutation Importance” no modelo “CATboost”.

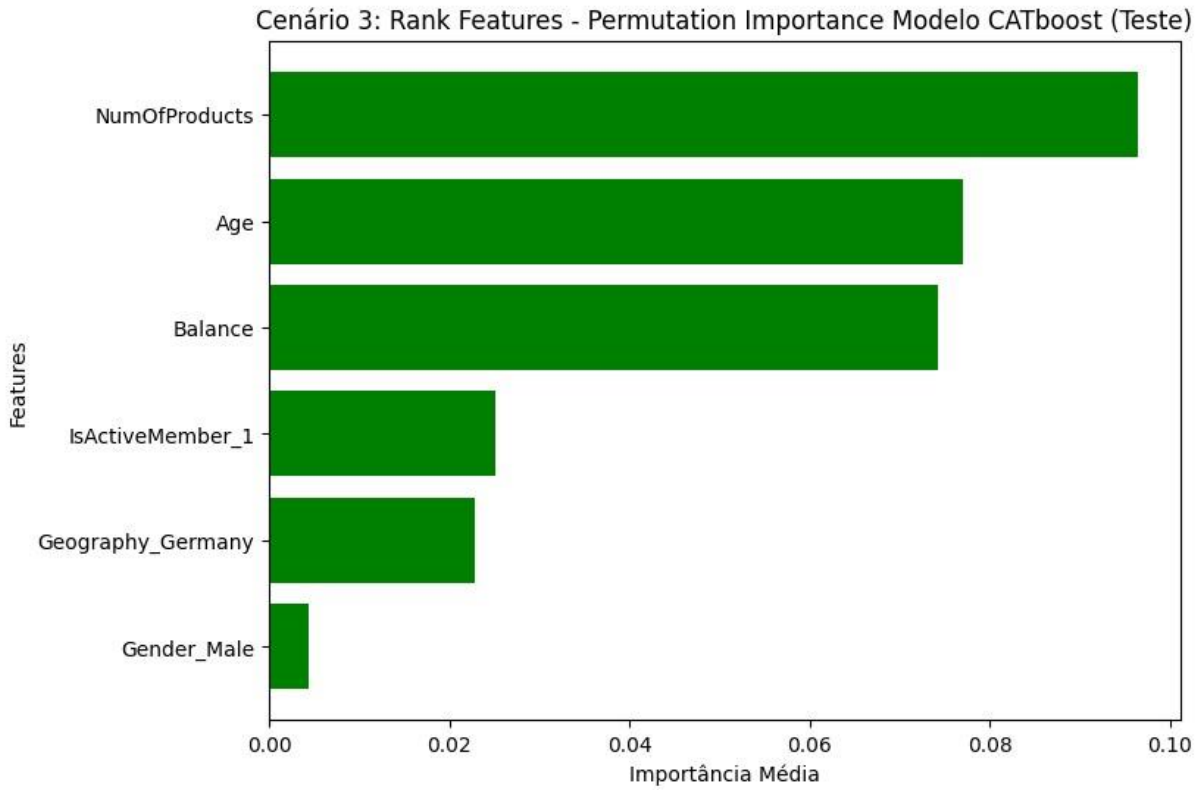

Figura 12. Gráfico de importância das variáveis após procedimento de “Permutation Importance” e corte de variáveis com baixa importância para o modelo “CATboost”
Fonte: Dados originais da pesquisa

1.4.4. Comparativo dos Modelos

Como dito anteriormente, foram consideradas todas as métricas de desempenho presentes nessa pesquisa, em todos os cenários, para monitorar e avaliar o desempenho dos modelos. Porém para a comparação dos modelos, a fim de decidir qual o melhor modelo, ou seja, o modelo que melhor prediz os eventos de “churn” dentro do contexto e objetivo proposto, foram consideradas as métricas “ROC AUC” e “Recall”, nessa respectiva ordem. Dessa forma na Tabela 6 temos os resultados das métricas de “ROC AUC” nos três cenários propostos para os três modelos aplicados.

Tabela 6. ROC AUC dos modelos por Cenário na base de teste

Modelos	ROC AUC		
	Cenário1	Cenário2	Cenário3
Logistico	0.7821	0.7814	0.7813
XGboost	0.8436	0.8691	0.8700
CATboost	0.8643	0.8676	0.8680

Fonte: Resultados originas da pesquisa

Analisando a evolução dos modelos ao longo dos três cenários conforme descrito na Tabela 6, em termos de “ROC AUC” o “XGBoost” e o “CATBoost” apresentaram desempenhos praticamente idênticos no cenário final, ambos atingindo um “ROC AUC” arredondado de 0.87. Embora o “XGBoost” tenha mostrado um crescimento mais expressivo ao longo dos cenários, com um aumento contínuo de 0.8436 para 0.87, o “CATBoost” manteve um desempenho mais estável, iniciando com 0.8643 e alcançando 0.87, com arredondamento ao final. Esses resultados indicam que ambos os modelos empataram em termos de eficiência geral após os ajustes. O modelo “Logístico”, no entanto, teve o pior desempenho, com uma leve queda de 0.7821 para 0.7813, e não mostrou melhorias significativas nos diferentes cenários.

Com esse empate de desempenho geral, a métrica “Recall” tornou-se ainda mais importante, assumindo papel de critério de desempate entre os modelos. Na Tabela 7 temos os resultados da métrica “Recall” nos três cenários, para os três modelos, aplicados na base de teste.

Tabela 7. Recall dos modelos por Cenário na base de teste

Modelos	Recall		
	Cenário1	Cenário2	Cenário3
Logistico	0.7125	0.7226	0.7405
XGboost	0.7710	0.8193	0.8193
CATboost	0.7812	0.8092	0.8804

Fonte: Resultados originas da pesquisa

Analisando os resultados de Recall, descritos na Tabela 7 para os três modelos, nos três cenários, o “CATBoost” se destacou, começando com 0.7812 e alcançando 0.8804 no cenário final, mostrando uma evolução significativa e consistente. O “XGBoost” apresentou uma melhora importante entre o cenário 1 e o cenário 2 (de 0.7710 para 0.8193), mas não evoluiu no cenário 3, mantendo-se com o mesmo Recall de 0.8193. O modelo logístico, apesar de ter apresentado a menor performance inicial, mostrou uma evolução constante, porém pouco relevante, subindo de 0.7125 para 0.7405, assim seguindo na posição de pior dos três modelos no cenário final.

Em termos gerais, se considerarmos os três modelos, não somente nas métricas usadas como decisores para escolher o melhor modelo, mas todas as apresentadas ao logo desta pesquisa, não houve grandes ganhos em capacidade preditiva ao logo dos três cenários. Notou-se, porém, que há diferença de complexidade de aplicação entre eles, sendo o modelo “Logístico” o mais simples, talvez mais indicado em cenários reais com baixa complexidade e qualidade de mão de obra, e o mais complexo, porém que mostrou ser mais

eficiente em termos de predição, o “CATboost” que seria o modelo ideal havendo mais estrutura e capacidade de mão de obra, pois possui maior número de hiperparâmetros e maior necessidade de processamento no momento de aplicação.

Considerações Finais

A aplicação dos três modelos na base de dados “Anonymous Multinational Bank” foi bem-sucedida, com todos os modelos apresentando bons resultados em prever “churn”, alcançando valores de “ROC AUC” acima de 0,78. O modelo “CATBoost” destacou-se como o melhor, com “ROC AUC” de 0,87 e “Recall” de 0,88. As variáveis mais importantes para as predições foram “NumOfProducts”, “Age” e “Balance”, conforme o teste de “Permutation Importance”. Embora os resultados não sejam considerados excelentes, o modelo se mostrou eficaz em identificar clientes propensos a “churn”. Sugere-se que futuras pesquisas explorem modelos que capturam contextos talvez não capturados nesta pesquisa, como modelos multiníveis e/ou redes neurais, que podem detectar padrões complexos nos dados. Além disso, técnicas de clusterização podem ajudar a entender comportamentos distintos entre variáveis preditoras. A adição de novas variáveis que melhorem a descrição dos eventos de “churn” e “não churn” também pode contribuir para aprimorar os modelos preditivos.

Agradecimentos

Agradeço a Deus pela força e sabedoria ao longo desta jornada. Sou grato à minha família pelo apoio incondicional e ao meu orientador pela orientação motivadora. Agradeço também aos professores da USP/Esalq, que tornaram o MBA um desafio prazeroso e enriquecedor.

Referências

Akiba, T.; Sano, S.; Yanase, T.; Ohta, T.; & Koyama, M. 2019. Optuna: A Next-generation Hyperparameter Optimization Framework. In: 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2019, Anchorage, AK, EUA. Anais... p. 2623-2631.

ANACONDA, INC. 2012. Anaconda Distribution. Disponível em: <https://www.anaconda.com/products/distribution>. Acesso em: 18 out. 2024.

Bruce, Peter; Bruce, Andrew. 2019. Estatística básica para cientistas de dados: 50 conceitos essenciais. O'Reilly, São Paulo, SP, Brasil.

Chen, T.; Guestrin, C. 2016. XGBoost: A Scalable Tree Boosting System. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, v. 3: 785-794.

Deotte, C. 2020. How To CV and How To Ensemble OOF Files. Disponível em: <https://www.kaggle.com/competitions/siim-isic-melanoma-classification/discussion/175614>. Acesso em: 29 set. 2024.

Dixon, W. J.; Tukey, J. W. 1968. Approximate behavior of the distribution of winsorized t (Trimming/Winsorization 2). Technometrics. 10(1): 83-93.

Drucker, P. F. 1954. The Practice of Management. Harper & Row Publishers, New York, EUA.

Favero, L. P. de; Belfiore, L. M. 2023. Manual de análise de dados. 2. ed. Atlas, São Paulo, SP, Brasil.

Field, A. 2024. Discovering Statistics Using IBM SPSS Statistics. Sage Publications, London, UK.

Fintechlab. 2019. Radar Fintechlab. Disponível em: <https://fintechlab.com.br/index.php/2019/06/12/8a-edicao-do-radar-fintechlab-registra-mais-de-600-iniciativas/>. Acesso em: 20 jun. 2024.

Fisher, R. 1922. Philosophical transactions of the Royal Society of London. Series A, containing papers of a mathematical or physical character, 222: 309-368.

Franceschi, P.R. de. 2019. Modelagens preditivas de churn: O caso do Banco do Brasil. Dissertação de Mestrado. Universidade do Vale do Rio dos Sinos, São Leopoldo, RS, Brasil.

Glady, N.; Baesens, B.; Croux. 2009. Modeling churn using customer lifetime value. European Journal of Operational Research 197(1): 402-411.

Guimarães, Olavo. 2021. Concorrência bancária e o Open Banking no Brasil. Revista de Defesa da Concorrência, v. 9, n. 1, p. 125-147, junho 2021. DOI: 10.62896/rdc.v9i1.709.

Hadden, J.; Tiwari, A.; Roy, R.; Ruta, D. 2007. Computer assisted customer churn management: State-of-the-art and future trends. Computers & Operations Research 34(10): 2902-2917.

Hanif, I. 2019. Implementing Extreme Gradient Boosting (XGBoost) Classifier to Improve Customer Churn Prediction. ICSA 2019 Proceedings of the 1st International Conference on Statistics and Analytics 2-3 August 2019: 434-453.

Harrison, Matt. 2019. Machine Learning: Guia de Referência Rápida. 1. ed. Novatec, São Paulo, SP, Brasil.

Jain, H.; Khunteta, A.; Srivastava, S. 2020. Churn prediction in telecommunication using logistic regression and logit boost. Procedia Computer Science 167: 101-112
James, G., Witten, D., Hastie, T., & Tibshirani, R. 2023. An Introduction to Statistical Learning with Applications in Python. 1ª edição. Springer, Nova York, NY, EUA.

Jamil, George Leal; Neves, Jorge Tadeu de Ramos. 2000. A era da informação: considerações sobre o desenvolvimento das tecnologias da informação. Perspectivas em Ciência da Informação. v. 5, n. 1, p. 41 - 53, jan./jun. 2000.

Kim, S.; Lee, H. 2022. Customer churn prediction in influencer commerce: An application of decision trees. Procedia Computer Science. v. 199: 1332-1339.

Klosterman, Stephen. 2020. Projetos de ciência de dados com Python. 1. ed. Novatec, São Paulo, SP, Brasil.

KPMG. 2023. Pulse of Fintech H2 2023 – Visão global. Disponível em: <https://kpmg.com/xx/en/home/industries/financial-services/pulse-of-fintech.html>. Acesso em: 20 jun. 2024.

Kumar, V.; Pansari, A. 2016. National culture, economy, and customer lifetime value: Assessing the relative impact of the drivers of customer lifetime value for a global retailer. Journal of International Marketing. v. 24, n. 1: 1-21.

Lawlor, K. B.; Hornyak, M. J. 2021. Smart goals: How the application of smart goals can contribute to achievement of student learning outcomes. Journal of Educational Leadership and Policy Studies. 5(1): 1-10.

McKinney, W. 2010. Data Structures for Statistical Computing in Python. Proceedings of the 9th Python in Science Conference, v. 9: 56-61.

Nelder, J.A., & Wedderburn, R.W.M. 1972. Generalized Linear Models. Journal of the Royal Statistical Society: Series A (General), 135(3): 370-384.

Oliphant, T.E. 2006. Guide to NumPy. NumPy.

Pamina, J.; Raja, J.; Bama, S.; Soundarya, S.; Sruthi, M.; Kiruthika, S.; Aiswaryadevi, V.; Priyanka, G. 2019. An Effective Classifier for Predicting Churn in Telecommunication. Jour of Adv Research in Dynamical & Control Systems 11: 221-229.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research, 12*, 2825-2830.

Permutation feature importance. Disponível em: <https://scikit-learn.org/1.5/modules/permutation_importance.html>. Acesso em: 20 Set. 2024.

Plotly Technologies Inc. [PLOTLY]. 2023. Plotly.py: Python Graphing Library. Disponível em: <https://plotly.com/python/>. Acesso em: 25 set. 2024.

Prokhorenkova, L.; Gusev, G.; Vorobev, A.; Dorogush, A.; Gulin, A. 2018. CatBoost: Unbiased boosting with categorical features. In: 32nd Conference on Neural Information Processing Systems, 2018, Montreal, Canadá. Anais... p. 6639-6649.

Reichheld, FF; Sasser Jr., WE 1990. Zero defections: Quality comes to services. Harvard Business Review. v. 68, n. 5: 105-111.

Rodrigues, L.M. 2023. Predição de churn usando modelos de aprendizado de máquina: estudo numa empresa fictícia de telecomunicação. Monografia. Universidade de São Paulo, Esalq, Piracicaba, SP, Brasil.

Rogers, David L. 2016. O manual de transformação digital. Columbia: Columbia Business School.

Stone, M. 1974. Cross-Validatory Choice and Assessment of Statistical Predictions. Journal of the Royal Statistical Society: Series B (Methodological). 36(2): 111–133.

Theodoridis, G.; Tsadiras, A. 2022. Applying machine learning techniques to predict and explain subscriber churn of an online drug information platform. Neural Computing and Applications, v. 34, n. 34: p. 19501–19514.

Ullah, I.; Raza, B.; Malik, A.; Imran, M.; Islam, S.; Kim, S. 2019. A churn prediction model using random forest: Analysis of machine learning techniques for churn prediction and factor identification in telecom sector. IEEE 7: 60134-60149.

Verbeke, W.; Martens, D.; Mues, C.; Baesens, B. 2012. Building comprehensible customer churn prediction models with advanced rule induction techniques. Expert Systems with Applications 39(17): 12202-12209.

XGBoost Documentation. (2022). XGBoost Python Package Documentation. Disponível em: https://xgboost.readthedocs.io.

Waskom, M. 2012. Seaborn: statistical data visualization. Disponível em: <https://seaborn.pydata.org/>. Acesso em: 25 set. 2024.

Wirth, R; Hipp, J. 2000. CRISP-DM: Towards a standard process model for data mining. Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining.

Wu, C.; Wang, L. 2022. A Comparative Analysis of Churn Prediction Models: A Case Study in Bank Credit Card. In: Journal of Supply Chain and Operations Management, Volume 20, Número 2, Dezembro 2022. Anais... páginas 120-138.

Yandex LLC. (2024). CatBoost Documentation. Disponível em https://catboost.ai/docs/

Zhu, B.; Baesens, B.; Broucke, SKLM 2017. Uma comparação empírica de técnicas para o problema de desequilíbrio de classes na previsão de rotatividade. Ciências da Informação. v. 408: 84-99.