Relatório Técnico: Análise do Pipeline de Predição de Churn em Telecomunicações

Análise de Execução de Código

29 de outubro de 2025

Conteúdo

1 Introdução

Este relatório apresenta uma análise dos resultados obtidos através da execução do pipeline de **machine learning** para predição de **churn** (cancelamento de serviço) de clientes na empresa de telecomunicações. O pipeline executado compreendeu as etapas de preparação dos dados, pré-processamento, modelagem comparativa entre Regressão Logística e XGBoost, e análise de explicabilidade (XAI) utilizando os valores SHAP (SHapley Additive exPlanations).

2 Configuração do Ambiente e Preparação dos Dados

A execução do pipeline iniciou-se com a instalação e importação das bibliotecas. Notavelmente, o código tentou fixar as versões 'xgboost==1.7.6' e 'shap==0.44.1'. No entanto, o ambiente de execução reportou as seguintes versões, que são mais recentes:

- **scikit-learn:** 1.6.1
- **xgboost:** 3.1.1
- **shap:** 0.49.1

Essa discrepância de versão, especificamente no XGBoost, teve um impacto direto na etapa de cálculo do SHAP, como será detalhado na Seção 5.

2.1 Carregamento e Limpeza Inicial

O arquivo `Churn.csv` foi carregado com sucesso. A etapa de pré-processamento incluiu:

1. **Remoção de Coluna:** A coluna `customerID` foi removida por não possuir valor preditivo.
2. **Tratamento de `TotalCharges`:** Esta coluna foi identificada como tipo `object` (texto). Na conversão para numérica, 11 valores nulos (`NaN`) foram gerados (correspondentes a entradas de texto vazias). Esses valores nulos foram subsequentemente preenchidos com `0`. Esta é uma decisão de negócio que assume que clientes sem `TotalCharges` registrado (provavelmente clientes novos com 0 meses de `tenure`) devem ter um total de 0.

2.2 Divisão Estratificada dos Dados

O conjunto de dados foi dividido em 80% para treino (5634 amostras) e 20% para teste (1409 amostras). Foi utilizada uma divisão estratificada pela variável alvo (`Churn`), garantindo que a proporção de clientes que deram *churn* fosse idêntica em ambos os conjuntos:

- **Proporção de Churn no Treino:** 26.54%
- **Proporção de Churn no Teste:** 26.54%

Isso assegura que a avaliação do modelo no conjunto de teste seja representativa da distribuição original dos dados.

3 Análise Exploratória Inicial (Boxplots)

A Figura 1 exibe a distribuição das três variáveis numéricas utilizadas pelo modelo.

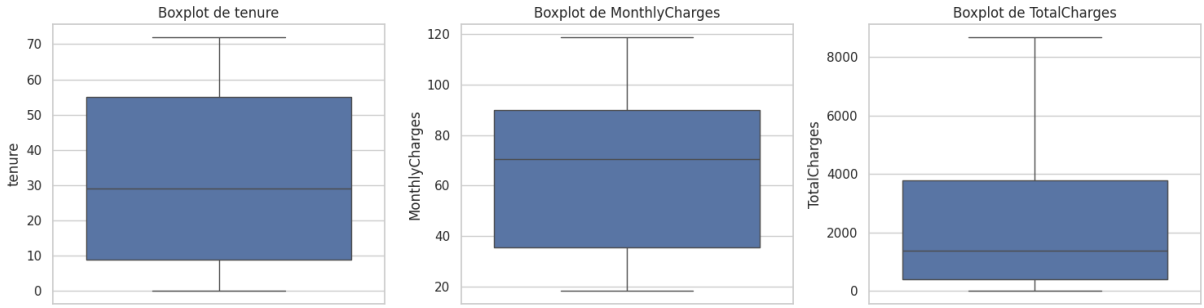

Figura 1: *Boxplots das variáveis numéricas* `tenure`*,* `MonthlyCharges` *e* `TotalCharges`*.*

- `tenure`**:** A mediana de tempo de contrato (antiguidade do cliente) situa-se em aproximadamente 29 meses. A distribuição é relativamente simétrica, sem *outliers* significativos.
- `MonthlyCharges`**:** A cobrança mensal mediana é de aproximadamente 70. A distribuição é ligeiramente assimétrica, com uma cauda mais longa para valores mais baixos, mas sem *outliers* extremos.
- `TotalCharges`**:** O total cobrado apresenta uma forte assimetria positiva (à direita). A mediana (cerca de 1400) é significativamente menor que o terceiro quartil e o limite superior. Os pontos acima do limite superior (whisker) indicam clientes com cobranças totais muito elevadas, o que é esperado para clientes com alto `tenure` e/ou altas cobranças mensais. Estes não são necessariamente erros, mas sim clientes de alto valor.

4 Avaliação Comparativa dos Modelos

Dois modelos foram treinados e avaliados no conjunto de teste (1409 amostras). A Tabela 1 compara as principais métricas de desempenho.

Tabela 1: *Métricas de Desempenho no Conjunto de Teste.*

Métrica	Regressão Logística	XGBoost
AUC-ROC	**0.8421**	0.8152
Acurácia Geral	**81%**	77%
Métricas para a Classe "Churn = 1"		
Precision (Precisão)	**0.66**	0.58
Recall (Revocação)	**0.56**	0.51
F1-Score	**0.61**	0.54

4.1 Interpretação dos Resultados

Contrariando a expectativa comum de que modelos baseados em árvores (como o XGBoost) superam modelos lineares, a **Regressão Logística apresentou um desempenho superior** em todas as métricas avaliadas neste conjunto de dados de teste.

- **AUC-ROC:** A Regressão Logística (AUC 0.8421) demonstrou uma capacidade superior de discriminar entre clientes que irão cancelar e os que não irão, em comparação com o XGBoost (AUC 0.8152).
- **Precision e Recall (Classe 1 - Churn):** A Regressão Logística (F1-Score 0.61) foi mais equilibrada na identificação de clientes que de fato cancelaram.
  - **Precision (0.66):** Das vezes que a Regressão Logística previu um *churn*, ela estava correta em 66% dos casos (vs 58% do XGBoost).
  - **Recall (0.56):** A Regressão Logística conseguiu identificar 56% de todos os clientes que realmente cancelaram (vs 51% do XGBoost).

5 Análise de Explicabilidade (SHAP) - Modelo XGBoost

A etapa de explicabilidade foi focada no modelo XGBoost.

5.1 Nota Técnica: Fallback para PermutationExplainer

Durante a execução, o pipeline encontrou um erro ao tentar usar o `shap.TreeExplainer`, que é o método otimizado para modelos de árvore:

```
Aviso: Falha ao ajustar base_score do Booster. [...] Detalhe: could
       not convert string to float: '[2.653532E-1]'
```

Este erro é resultado direto da incompatibilidade entre a versão do `xgboost==3.1.1` (que formata seu `base_score` como uma string de lista) e a versão do `shap==0.49.1` (que esperava um float simples).

O pipeline ativou corretamente seu mecanismo de *fallback*, utilizando o `shap.explainers.Permutation`. Este explicador é agnóstico ao modelo (trata-o como uma "caixa-preta") e estima os valores SHAP através de permutações dos dados. Embora funcional, é computacionalmente muito mais intensivo (levou 3 minutos e 20 segundos) e os valores são uma aproximação, não os valores exatos que o `TreeExplainer` calcularia.

5.2 Análise dos Gráficos SHAP

5.2.1 Importância Geral das Features

A Figura 2 mostra a média do impacto absoluto de cada *feature* nas previsões. Ela responde à pergunta: "Quais *features* mais influenciam o modelo, em média?".

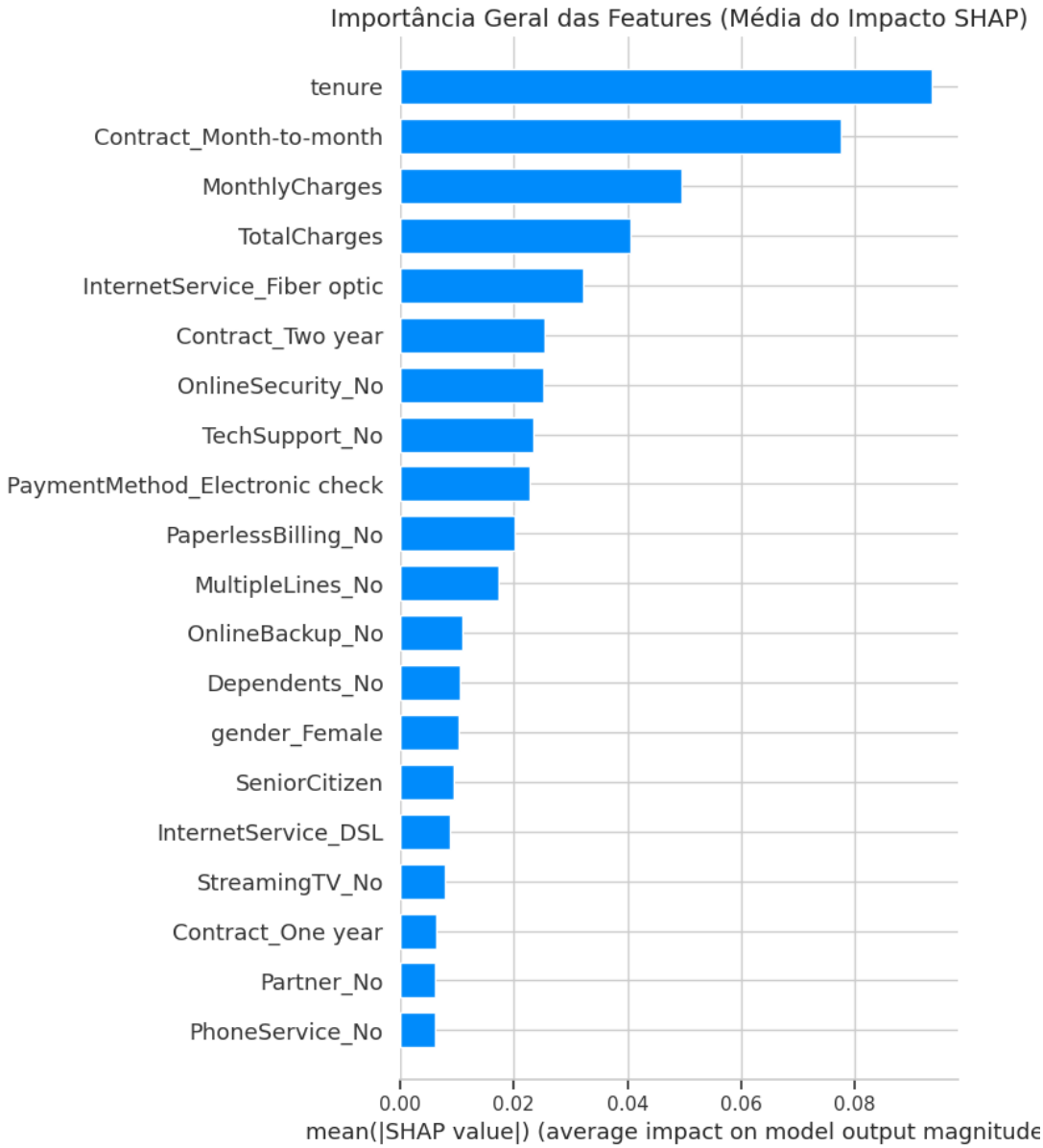

Figura 2: *Importância Geral das Features (Média do Impacto SHAP).*

As cinco *features* mais impactantes para o modelo XGBoost são:

1. `tenure`: O tempo que o cliente está na empresa.
2. `Contract_Month-to-month`: Se o cliente possui um contrato do tipo "mês a mês".
3. `MonthlyCharges`: O valor da cobrança mensal.
4. `TotalCharges`: O valor total cobrado historicamente.
5. `InternetService_Fiber optic`: Se o cliente utiliza o serviço de fibra óptica.

5.2.2 Impacto (Beeswarm Plot)

A Figura 3 fornece uma visão muito mais rica. Cada ponto é um cliente no conjunto de teste. O eixo X mostra o "valor SHAP"(o impacto na previsão), e a cor indica o valor da *feature* (Vermelho = Alto, Azul = Baixo).

- **Valores SHAP Positivos** (> 0)**:** Empurram a previsão para **Churn (Classe 1)**.
- **Valores SHAP Negativos** (< 0)**:** Empurram a previsão para **Não Churn (Classe 0)**.

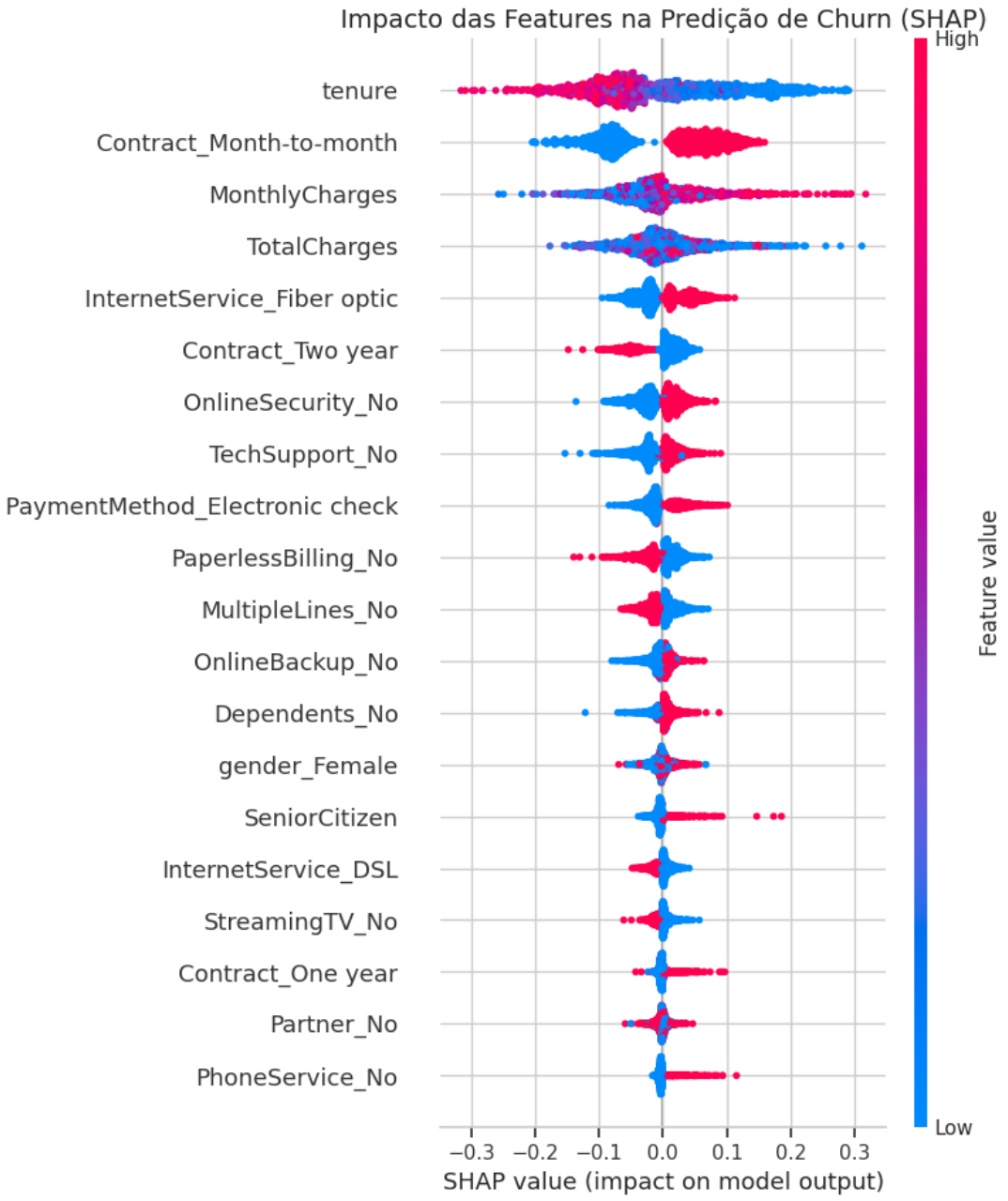

Figura 3: *Impacto das Features na Predição de Churn (SHAP).*

Interpretação Detalhada (Top 3):

1. `tenure`**:** É o fator mais claro.
    - **Pontos Azuis (Baixo** `tenure`**):** Estão concentrados à direita (valores SHAP positivos). Isso indica que **clientes novos têm altíssima probabilidade de *churn***.

- **Pontos Vermelhos (Alto `tenure`):** Estão concentrados à esquerda (valores SHAP negativos). **Clientes antigos (fiéis) têm baixíssima probabilidade de *churn*.**

2. `Contract_Month-to-month`**:** Esta é uma *feature* binária (0 ou 1) criada pelo OneHotEncoder.
   - **Pontos Vermelhos (Valor 1):** O cliente *possui* um contrato "mês a mês". Esses pontos têm valores SHAP fortemente positivos. Isso **aumenta muito o risco de *churn*.**
   - **Pontos Azuis (Valor 0):** O cliente *não possui* contrato "mês a mês"(ou seja, tem contrato de 1 ou 2 anos). Esses pontos têm valores SHAP negativos, **reduzindo o risco de *churn*.**
3. `MonthlyCharges`**:**
   - **Pontos Vermelhos (Cobranças Altas):** Estão predominantemente à direita (valores SHAP positivos). **Cobranças mensais altas aumentam o risco de *churn*.**
   - **Pontos Azuis (Cobranças Baixas):** Estão predominantemente à esquerda (valores SHAP negativos). **Cobranças baixas diminuem o risco de *churn*.**

6 Conclusão Geral

O pipeline foi executado com sucesso, desde a limpeza dos dados até a explicabilidade.

A principal conclusão da modelagem é que, para este conjunto de dados de teste, o modelo de **Regressão Logística (AUC 0.8421) superou o modelo XGBoost (AUC 0.8152)** em métricas de discriminação (AUC) e na capacidade de identificar corretamente os clientes que cancelaram (F1-Score).

A análise de explicabilidade do XGBoost, obtida via `PermutationExplainer`, identificou claramente os principais impulsionadores do *churn*:

- **Fatores de Risco (Aumentam Churn):** Baixo `tenure`, contrato `Month-to-month` e `MonthlyCharges` elevadas.
- **Fatores de Proteção (Diminuem Churn):** Alto `tenure` (fidelidade), contratos de 1 ou 2 anos e `MonthlyCharges` baixas.

A execução também destacou uma importante incompatibilidade técnica entre as versões recentes das bibliotecas `xgboost` e `shap`, que foi contornada com sucesso pelo mecanismo de *fallback* do código.