Previsão de Churn de Clientes: Do SQL ao Machine Learning

🔗 **Link para o Repositório no GitHub:** Previsão de Churn de Clientes

Descrição do Projeto

Este projeto apresenta a construção de um pipeline de dados end-to-end focado na resolução de um problema clássico de negócios: a evasão de clientes (Churn). O objetivo principal é identificar padrões de cancelamento e desenvolver um modelo preditivo capaz de classificar clientes com alto risco de abandono, permitindo que a equipe de retenção atue de forma proativa.

O escopo técnico abrange desde a extração de dados brutos em um banco de dados relacional até a análise exploratória, engenharia de recursos e treinamento de algoritmos de Machine Learning, comparando diferentes abordagens para maximizar a precisão da predição.

Tecnologias Utilizadas

- **Banco de Dados:** SQLite (consultas e agregações iniciais)
- **Linguagem Principal:** Python 3
- **Manipulação de Dados:** Pandas, NumPy
- **Integração de Dados:** SQLAlchemy
- **Visualização:** Matplotlib, Seaborn
- **Machine Learning:** Scikit-Learn (Random Forest, Gradient Boosting)

Estrutura do Repositório

- `sql/`: Contém os scripts de consultas SQL para extração e análise preliminar dos dados (ex: cálculo de taxa de churn geral, média de mensalidade e distribuição de planos).
- `python/`: Diretório contendo o banco de dados SQLite local (`churnDB`) e o notebook principal (`churn_model.ipynb`) com todo o fluxo de análise e modelagem.
- `requirements.txt`: Lista de dependências para reprodução do ambiente virtual.

Etapas de Desenvolvimento

1. Extração e Limpeza de Dados

Os dados foram armazenados em um banco SQLite simulando um ambiente de produção. A conexão foi estabelecida via SQLAlchemy. A etapa de pré-processamento incluiu:

- Tratamento de valores nulos e conversão de tipos de dados (`TotalCharges`).
- Padronização de categorias redundantes (agregação de serviços inexistentes na categoria principal "No").
- Binarização de variáveis categóricas dicotômicas nativas e aplicação de One-Hot Encoding (`pd.get_dummies` com `drop_first=True`) para variáveis multicategóricas, prevenindo problemas de multicolinearidade.

2. Análise Exploratória de Dados (EDA)

A investigação estatística e visual revelou insights críticos de negócio:

- **Fator Financeiro:** O gráfico de caixa (Boxplot) demonstrou que o churn é significativamente mais comum entre clientes com cobranças mensais mais elevadas.
- **Fator Temporal:** A distribuição por tempo de contrato revelou um pico acentuado de cancelamentos nos primeiros 5 meses, indicando uma vulnerabilidade crítica no processo de onboarding e retenção inicial da empresa.

3. Modelagem de Machine Learning

O conjunto de dados foi dividido de forma segura (Train/Test Split) e dois modelos principais baseados em árvores foram avaliados:

- **Random Forest Classifier:** Utilizado como modelo base, apresentou uma acurácia global de 78%. No entanto, a matriz de confusão indicou oportunidades de melhoria na distinção da classe minoritária (Churn positivo).
- **Gradient Boosting Classifier:** Selecionado como modelo final devido à sua abordagem sequencial de otimização de erros.

4. Otimização Direcionada a Negócios (Threshold Tuning)

Em problemas de Churn, o custo de um falso positivo (oferecer descontos a quem não iria cancelar) pode ser alto. Para otimizar a lista de clientes enviada à equipe de marketing, o *Decision Threshold* do Gradient Boosting foi ajustado empiricamente para 0.70.

- **Resultado do Ajuste:** A precisão do modelo para clientes que realmente iriam cancelar saltou para mais de 75%, garantindo alta assertividade nas campanhas de retenção para os clientes de maior risco.

5. Comparação Final

O desempenho estatístico foi validado utilizando a Curva ROC. O Gradient Boosting apresentou uma área sob a curva (AUC) superior, consolidando-se como a melhor solução para este conjunto de dados quando aliado ao ajuste de limiar.

Como Executar o Projeto

1. Clone este repositório:

```
git clone [https://github.com/seu-usuario/seu-repositorio.git]
(https://github.com/seu-usuario/seu-repositorio.git)

 Crie e ative um ambiente virtual:
 Bash

 python -m venv .venv
 source .venv/bin/activate  # No Linux/Mac
 .venv\Scripts\activate     # No Windows
```

```
 Instale as dependências:
 Bash

 pip install -r python/requirements.txt

 Execute o Jupyter Notebook contido na pasta python/ para visualizar as
análises e o treinamento dos modelos.
```

Projeto desenvolvido com foco na aplicação prática de Engenharia e Ciência de Dados para tomada de decisões estratégicas.