Introdução

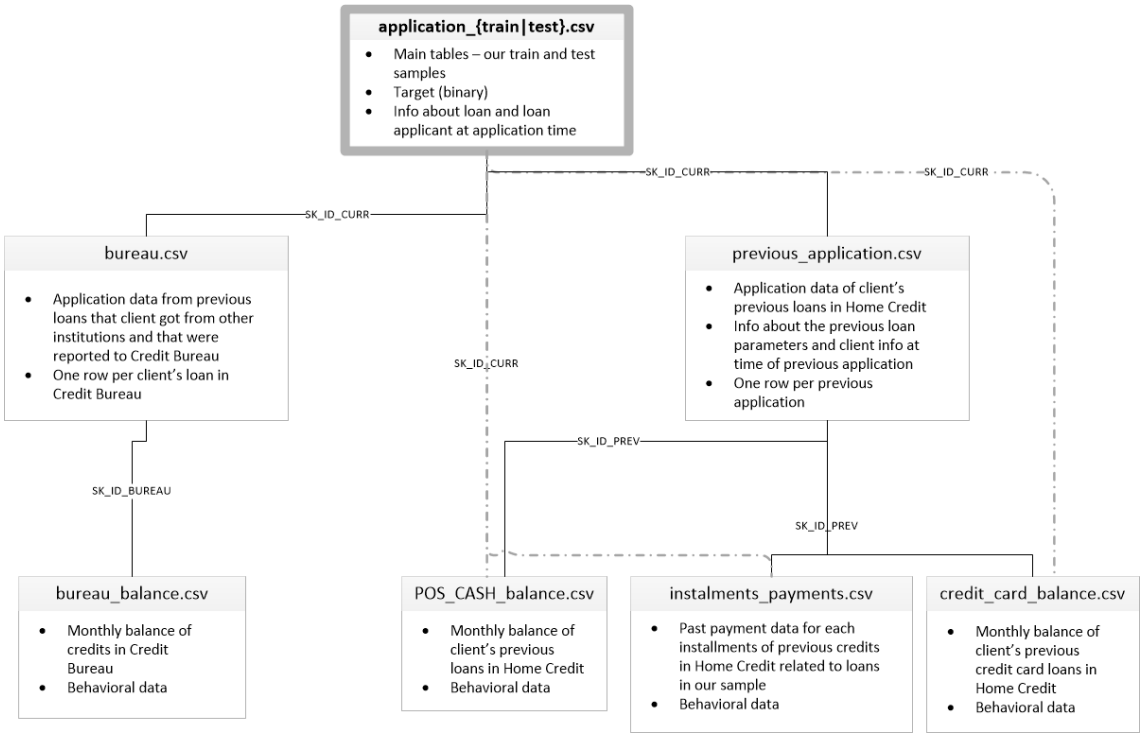
Sejam bem-vindo ao desafio técnico para o processo seletivo referente a posição de Engenheiro(a) de Machine Learning Sr. no Itaú!

Nesse documento você encontrará as orientações a respeito dos insumos para realização do desafio. Quaisquer dúvidas, envie e-mail para:

- armindo.guerra@itau-unibanco.com.br

Dados

Como fonte de dados utilizaremos o case ***Home Credit Default Risk***, disponível no Kaggle (https://www.kaggle.com/competitions/home-credit-default-risk/data). A figura seguir apresenta a organização dos dados e seus respectivos relacionamentos.



Cenário

Suponhamos que você tenha sido contratado pelo Itaú para desenvolver um sistema de detecção de risco de crédito para empréstimos bancários. A empresa coletou uma grande quantidade de dados de transações passadas. Sua tarefa é construir um modelo de Machine Learning robusto que seja capaz de identificar a probabilidade de o cliente pagar o empréstimo.

Desafios

Desafio nº 1: Explique de forma resumida todas as etapas que compõe a metodologia CRISP-DM. Se possível, dê exemplos.

Desafio nº 2: Considerando o case ***Home Credit Default Risk*** como sua base de dados principal, faça uma análise exploratória acerca dos clientes, representados na base pelo identificador único SK_ID_CURR. Destaque padrões e/ou anomalias nos dados e identifique como isso pode ser útil (ou não) para o processo de modelagem. Sempre que possível use gráficos para apresentar visualmente as análises realizadas.

Desafio nº 3: Construa um modelo de Machine Learning que seja capaz de identificar a probabilidade de o cliente (SK_ID_CURR) pagar o empréstimo, seguindo as seguintes etapas:

1. Crie o modelo, inicialmente, olhando apenas para os dados da tabela application_train.csv;
2. Teste no mínimo 3 algoritmos diferentes, sendo que um deles DEVE ser Regressão Logística;
3. Escolha a métrica mais adequada para avaliar robustez do modelo e justifique as escolhas e os resultados;
4. Agregue à tabela application_train.csv *features* vindas das demais tabelas disponíveis (como a bureau.csv, previous_application.csv, etc.) e refaça os itens 2 e 3 com a nova base criada.
5. Compare os resultados obtidos até o momento.

Obs.: Recomendável usar Pipelines com scikit-learn: https://scikit-learn.org/stable/modules/generated/sklearn.pipeline.Pipeline.html

Desafio nº 4: Transforme melhor modelo obtido em um arquivo serializado (com pickle ou joblib) e crie um mecanismo de software para carregá-lo em tempo de execução e executá-lo sob demanda, acionado por uma API. Crie a API com duas rotas, uma apenas para acionamento da execução *batch* do modelo (salvando os resultados em um arquivo *parquet* "localmente", sem a necessidade de devolver o resultado na *response*) e outra rota que receba os dados em um *payload* e retorne à probabilidade de o cliente pagar o empréstimo.

Obs.: É livre a escolha da tecnologia para criação da API (recomendamos Flask). Não se preocupe com safras, ou com a questão temporal. A execução batch pode utilizar sempre os mesmos dados.

Critérios de Avaliação:

- Qualidade do modelo;
- Capacidade de explicar as decisões tomadas;
- Organização e documentação do código;
- Eficiência na implantação do modelo simulando um *deploy* (mesmo que local).

Outras recomendações:

- Desenvolva os desafios com Python, na versão mínima 3.8;
- Adote as recomendações da PEP8;
- Organize o código com orientação a objetos, sempre que pertinente;
- Escreva testes unitários, sempre que pertinente;
- Commite os resultados no seu github/gitlab pessoal, com as orientações para execução dos códigos no README.md, e envie o link para avaliarmos.