**Criação da base de dados em PostgreSQL com um arquivo CSV**

Criei a database diretamente no PostgreSQL com o nome loan\_risk. Em seguida, importei o csv para o banco de dados sql utilizando o seguinte em Python, com as bibliotecas pandas, psycopg2 e sqlalchemy.

Testei se a importação do csv para o banco de dados foi bem sucedida também com um select dos primeiros 10 clientes no pgAdmin (Inseri os nomes usuário e senha nos seus respectivos lugares, utilizei minhas credenciais para o caso):

import pandas as pd

import psycopg2

from sqlalchemy import create\_engine

import os

current\_dir = os.getcwd()

file\_path = os.path.join(current\_dir, "Default\_Fin.csv")

# Carregar o CSV

df = pd.read\_csv(file\_path)

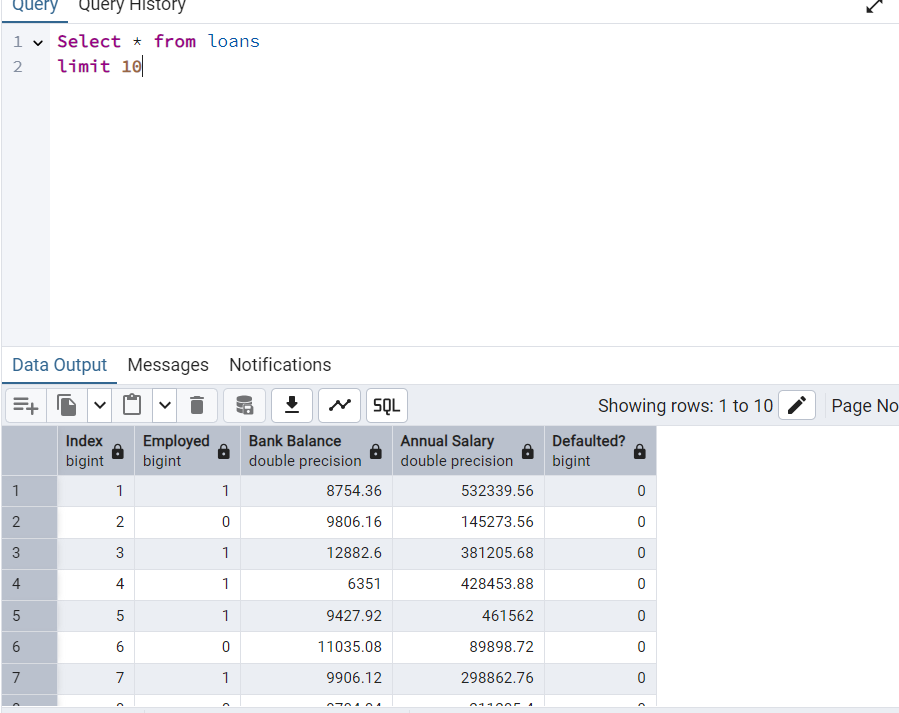
# Conexão com o PostgreSQL

engine = create\_engine("postgresql+psycopg2://usuário:senha@localhost:5432/loan\_risk")

# Exportar para SQL

df.to\_sql("loans", engine, *if\_exists*="replace", *index*=False)

print("Dados importados para o PostgreSQL!")



Após a importação, conectei com o banco de dados loan\_risk ao Power para gerar os relatórios,visualizações e análises.

Fonte do CSV utilizado para análise: <https://www.kaggle.com/datasets/kmldas/loan-default-prediction?resource=download>

**Tratamento de dados ETL**

O tratamento inicial foi feito no próprio Power Querry (Power BI). Fazendo o tratamento de dados inicial do csv no próprio power querry do power bi, coloquei a coluna index que se refere ao índice do indivíduo como um número inteiro, as colunas employed e defaulted que significam respectivamente empregado e regularizado vieram em formato número como 1 e 0, troquei para serem valores booleanos no power bi também. Para Bank Balance e Annual Salary, coloquei como números decimais fixos a fim de normalizar o comportamento dos valores monetários sem oscilação muito grande de decimais, apenas considerando os valores possíveis da moeda.

Após esse tratamento comecei a identificação dos outliers. Buscarei tratar os outliers criando medidas de quartil no power bi e criando limitações para definir se é um outlier ou não. A partir disso extrairei uma coluna com valor booleano para identificações dos outliers.

Criei as medidas para o primeiro quartil(q1), o terceiro quartil(q3) e para o intervalo de quartil(IQR).

Dax

Q1 Salary = PERCENTILEX.INC(ALL('public loans'),'public loans'[Salário anual], 0.25)

Q3 Salary = PERCENTILEX.INC(ALL('public loans'), 'public loans'[Salário anual], 0.75)

IQR Salary = [Q3 Salary] - [Q1 Salary]

Agora criei as medidas de limite para detectar os outiliers:

Dax

Limite Inferior = [Q1 Salary] - 1.5 \* [IQR Salary]

Limite Superior = [Q3 Salary] + 1.5 \* [IQR Salary]

Após isso crio uma coluna Outlier Salary de valor booleano para identificar os salários que aparentam ser outliers ou não.

Dax

Outlier Salário =

IF(

    'public loans'[Salário anual] < [Limite Inferior] ||

    'public loans'[Salário anual] > [Limite Superior],

    "Sim", "Não"

)

Não"

)

Segundo a análise, nenhum outlier aparentemente foi identificado.

Após este tratamento de identificação de outliers criarei colunas derivadas para poder realizar uma análise mais abrangente.

Colunas criadas Faixa de Renda (income range) Classe baixa, média ou alta.

Dax

Faixa de renda = SWITCH(true(),'public loans'[Salário anual]<30000,"Classe baixa", 'public loans'[Salário anual]<80000,"Classe média","Classe alta")

Balance level (Nível de saldo na conta) Negativo, baixo ou alto.

Dax

Balance level = if ('public loans'[Bank Balance] < 0, "Negative", IF('public loans'[Bank Balance] < 5000, "Low", "High"))

Agora criei uma medida para indicar a taxa de inadimplência =

Dax

Taxa Inadimplência =

DIVIDE(

    CALCULATE(

        COUNTROWS(

            FILTER(

                'public loans',

                'public loans'[Regularizado?] = FALSE()

            )

        )

    ),

    COUNTROWS('public loans'),

    0

)

Medida para perda financeira estimada se os clientes de alto risco inadimplirem

Dax

Perda Financeira Estimada =

SUMX(

    FILTER('public loans', [Score de Risco Simples] >= 3),

    [Salário Anual] \* 0.1  // Assume-se 10% do salário como exposição

)

coluna para criar um cluster de cilentes baseado em perfis utilizado no caso para exemplo como “conservador”, “endividado” e “equilibrado”

Dax

Perfil financeiro (Coluna) =

VAR nivel\_saldo = 'public loans'[Nível de saldo bancário]

VAR score\_risco = 'public loans'[Score de Risco Simples]

VAR percentual = 'public loans'[% Salário por saldo]

RETURN

SWITCH(

    TRUE(),

    AND(nivel\_saldo = "Alto", score\_risco = 0), "Conservador",

    AND(percentual < 0.1, score\_risco > 2), "Endividado",

    "Equilibrado"

)

Coluna de score de upsell para avaliar clientes em potencial para produtos financeiros premium:

Dax

Score de Upsell (Coluna) =

VAR Potencial = IF('public loans'[Potencial de Investimento] = "Alto Potencial", 3, 0)

VAR SaldoAlto = IF('public loans'[Nível de saldo bancário] = "Alto", 2, 0)

VAR RendaAlta = IF('public loans'[Faixa de renda] = "Alta Renda", 1, 0)

RETURN

    Potencial + SaldoAlto + RendaAlta

Nesta etapa do projeto traduzi todos os termos da tabela para o português.

Durante a elaboração deste projeto neste momento pensando em escalabilidade, reconheci que tratar os outliers em python otimizaria significativa o projeto para uma fonte de dados grande uma vez que em power bi, para cada novo indivíduo inserido pesaria mais este tratamento de outliers, idealmente passariam para a análise apenas os que não fossem. Farei esta adaptação ao longo da execução. Pensando também na automatização do banco de dados e monitoramento do csv, pretendo aprender como utilizar o Apache Airflow para criar um pipeline automatizado para atualizações constante do banco de dados.

Criei uma coluna de score de risco simples baseando-se em múltiplos fatores para priorizar clientes críticos.

Dax

Score de Risco Simples = Var RiscoEmprego = IF(NOT([Empregado]), 2, 0)

Var RiscoSaldo = IF([Nível de saldo bancário] = "Negativo", 3, 0)

Var RiscoRenda = IF([Faixa de renda] = "Classe baixa", 1, 0)

RETURN

    RiscoEmprego + RiscoSaldo + RiscoRenda

Dax

% de clientes em risco crítico

% Clientes em Risco Crítico =

DIVIDE(

    COUNTROWS(

        FILTER(

            'public loans',

            'public loans'[Score de Risco Simples] >= 4

        )

    ),

    COUNTROWS('public loans'),

    0

)

Como no dataset não possui nenhum cliente nestas condições, deixei está medida criada como ferramenta a ser utilizada no mundo real e criei uma nova medida visual de clientes em maior risco contando que o maior risco no gráfico são os clientes com score de risco 1 a 3 chamado porcentagem de clientes em risco:

Dax

% Clientes em Risco =

DIVIDE(

    COUNTROWS(

        FILTER(

            'public loans',

            'public loans'[Score de Risco Simples] >= 2

        )

    ),

    COUNTROWS('public loans'),

    0

)

E para elaborar clientes que podem ser alvos de produtos financeiros premium achei interessante criar a coluna potencial de investimento também

Dax

Potencial de Investimento = IF(AND('public loans'[Nível de saldo bancário] = "Alto", 'public loans'[Faixa de renda] = "Classe alta"), "Alto Potencial", "Padrão")

Para identificar a prioridade de ação também criei uma coluna levando como base o score de risco simples que foi criado anteriormente:

Prioridade de Ação = SWITCH(TRUE(), 'public loans'[Score de Risco Simples] >= 4, "Contatar Urgente", 'public loans'[Score de Risco Simples] >= 2, "Monitorar","Baixa Prioridade")

Potencial dos top 10 clientes premium, sendo oferecido para eles produtos exclusivos:

Dax

Potencial Top 10 Premium =

VAR TopClientes =

    TOPN(

        10,

        SUMMARIZE(

            'public loans',

            'public loans'[Índice],

            'public loans'[Saldo Bancário],

            'public loans'[Score de Upsell (Coluna)],

            'public loans'[Faixa de Renda]

        ),

        'public loans'[Saldo Bancário] \* ('public loans'[Score de Upsell (Coluna)]/10),

        DESC

    )

RETURN

    SUMX(

        TopClientes,

        'public loans'[Saldo Bancário] \* 0.12  // Taxa de conversão estimada em 12%

    )

**Reflexões em relação ao projeto**

A ideia do projeto é ir se incrementando com o tempo, mas lapidar para que não os dados para o Power BI cheguem mais processados tratando diretamente via Python para tornar o projeto mais escalável e que não pese tanto na geração do relatório, além de extrair de diversos datasets para criar uma análise mais enriquecida sobre possíveis causas em potencial da inadimplência, pois analisando apenas um dataset a análise fica muito restrita.

Durante este projeto reconheci e percebi a necessidade de aprender novas ferramentas para tornar o projeto mais abrangente como o apache airflow para a automatização da ingestão de de dados como citado anteriormente e ferramentas de computação em nuvem como AWS ou Azure, para possibilitar a otimização de consultas sql e para que o banco de dados não seja um banco de dados local e sim um virtual e colaborativo.

Com um dataset mais robusto com histórico de crédito, idade ou variações ao longo do tempo também seria possível aplicar técnicas de machine learning para calcular probabilidades mais precisas de inadimplência. Criei um modelo em power point para construir a base de onde estarei inserindo o dashboard via Power Bi. Em seguida construi os dashboards para visualizações em potencial dos gráficos necessários planejei e estruturei como eu queria montar esboçando antes de partir para a elaboração.

Realizei dois dashboards um de análise de risco de crédito e outro identificando oportunidades e upsell.

Elaborei gráficos e tabelas em potencial para destacar as observações deste dataset e visualizarmos os dados de uma maneira mais clara, a fim de guiar a tomada de decisões. Neste processo continuo utilizando o Power BI e mantive os dashboards na ferramenta para manter a interatividade nos gráficos.

**Insights retirados**

Após a elaboração da visualização, baseada na visualização guiada pelas minhas impressões apenas para este dataset relatei os insights que foram retirados para avaliar o valor que estes dados possuem para o incremento dos negócios. Da construção e do processo de construção segundo os insights para este banco fictício obtidos foram:

**Insights para análise de risco de crédito**

* Alta Taxa de Inadimplência:

Mostrando a necessidade de revisar os critérios de concessão de crédito e implementar políticas de cobrança mais rigorosas e investir em análise preditiva para identificar clientes propensos à inadimplência antes da concessão.

* Quase 30% dos Clientes em Risco:

Desenvolver campanhas de renegociação e educação financeira para este grupo ou monitorar de perto e criar alertas automáticos para movimentações suspeitas.

* Perda Financeira Estimada Significativa:

Priorizar ações de recuperação de crédito e renegociação com clientes inadimplentes e avaliar provisionamento para perdas e ajustar estratégias de mitigação.

* O risco está Concentrado em Clientes de Baixa e Média Renda:

Oferecer produtos financeiros mais acessíveis e personalizados para esses segmentos e reforçar acompanhamento e suporte para evitar agravamento do risco.

* Clientes de Alto Saldo Bancário Também em Risco:

Realizar análise individualizada desses clientes para entender causas do risco e propor soluções de investimento ou diversificação para retenção.

* Ações Predominantemente de Baixa Prioridade ou Monitoramento:

Refinar critérios de segmentação para identificar oportunidades de ação mais assertivas e reavaliar clientes classificados como "Baixa Prioridade" para evitar negligência de riscos ocultos.

**Insights para oportunidades e upsell**

* Perfis Financeiros Conservador e Equilibrado Dominam:

Desenvolver produtos e comunicações alinhados ao perfil conservador, como investimentos de baixo risco e incentivar a diversificação de portfólio para clientes equilibrados.

* Potencial Premium Concentrado em Poucos Clientes:

O potencial dos 10 principais clientes premium soma $35,56 mil, indicando alta concentração de valor em poucos clientes. Criar programas de fidelização e atendimento personalizado para esses clientes premium e monitorar de perto para evitar churn e identificar oportunidades de cross-sell.

* Alto Potencial de upsell em Clientes de Classe Alta:

Focar campanhas de upsell e produtos exclusivos neste segmento e oferecer consultoria financeira personalizada para maximizar o engajamento.

* Clientes com Alto Potencial de Investimento Têm Saldos Elevados:

Priorizar abordagem comercial ativa para esses clientes e apresentar oportunidades de investimento diferenciadas e acompanhamento dedicado.

**Conclusões e aprendizados em relação a este projeto:**

Este dataset apresenta resultados discrepantes e distantes do que ocorrem na realidade a exemplo de 97% de taxa de inadimplência nesta base de dados, além de se limitar a esta única base de dados o que prejudica muito para uma análise mais robusta e rica. Um padrão muito atípico, só se houvesse uma significativa fraude ou uma falha muito considerável na avaliação este caso poderia ser realidade. Estes dados se fossem num caso real exigiriam uma investigação imediata. É possível replicar este projeto para outros bancos de dados apenas fazendo as modificações necessárias e adaptações. O mais recomendado é automatizar o processo de ETL, neste projeto este processo está elaborado de forma completamente manual. Pretendo após a execução deste projeto de entrada aprender ferramentas que me auxiliem neste processo de automatização, diversificar a base de dados para os futuros projetos e estudar mais afundo cases reais a fim de edificar projetos que realmente possam agregar valor para qualquer empresa.