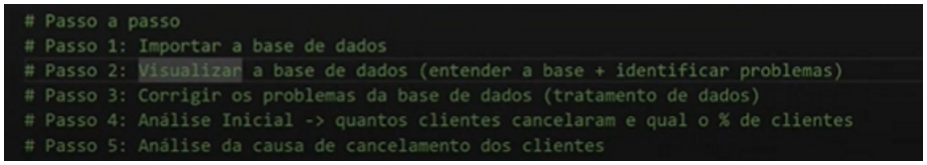
# Analise de dados com Python

sexta-feira, 17 de janeiro de 2025 02:14

Após ler o case, anotar o passo a passo do que fazer.

Correção de problemas da base de dados (tratamento de dados).
Pergunta chave para uma análise inicial.




Arquivo.pynb consegue dividir o código em blocos, permitindo textos e depois linhas de código.

Python é muito utilizado na área de dados pelas bibliotecas que existem. Para instalar basta digitar ! pip intall (nome da biblioteca) ou no terminal sem a !.

Nesse projeto serão utilizadas as bibliotecas: pandas, numpy, openpyx1, nbformat, ipykernel e plotly.

**Passo 1: Importar base de dados**

*Import pandas as pd* (pd é um apelido)

Uso do comando pd.read_csv ("nome do arquivo.csv") que recebe como parâmetro o nome do arquivo, serve para ler a base de dados.

Criar uma variável para receber essa base de dados.

*tabela = pd.read_csv ("cancelamentos_sample.csv")*

**Passo 2: Visualizar base de dados**

É possível utilizar o comando Display, que só existe nessa extensão .pynb, para visualizar os dados de forma mais bonita. Entretanto pode-se utilizar o print.

*display (tabela)*

Obs: A análise deve ser feita comparando uma coluna da tabela com outras e buscar uma relação.

Verificar quais serão possivelmente as colunas com dados úteis para resolver o problema.

**Passo 3: Tratamento de dados**

Utilizar o .drop para retirar colunas inúteis da variável tabela.

*tabela = tabela.drop(columns = "CustomerID")*
*display (tabela)*

Utilizar *display(tabela.info)* para verificar as informações da base de dados, quais as colunas, quantas estão preenchidas e etc.

Obs: Para desfazer algo é só retirar a linha do código e rodar todos os blocos de código.

Quando existe colunas que possuem poucos dados vazios, comparando sempre com o número total, pode-se excluir. Caso seja uma porcentagem alta (30% >=) é possível escolher se deseja preencher esses dados de forma manual, com a média de valores, com o maior valor possível ou o menor.

Descartar linhas que possuem dados vazios de informações relevantes para análise.

*tabela = tabela.dropna()*

**Passo 4: Análise inicial**

Contar os valores de uma coluna, nesse caso, a coluna 'cancelou'.

```
display(tabela["cancelou"].value_counts())
```

**Passo 5: Análise da causa do problema**

Olhar cada coluna e comparar com a coluna do cancelamento.
É melhor utilizar isso com gráficos e para isso há a biblioteca plotly.

*Import plotly.empress as px*

Para criar um gráfico é sempre necessário criar o gráfico e depois exibir o gráfico. Uso do gráfico do tipo histograma (quantidade de pessoas em cada situação). Os gráficos criados são quase dashboards.

Pesquisar no site do plotly para ir pegando como personalizar o gráfico.

Obs: É sempre bom fazer um gráfico simples, comparando sempre 2 informações.

**Passo 6: Insights**

Fazer uma simulação caso suas soluções sejam atendidas.
Filtrar sua base de dados para exibição.

tabela = tabela[tabela["duracao_contrato"] != "Monthly"]
A nova tabela será igual a antiga, sendo que a coluna duração_contrato deverá ter valor diferente de Monthly

É possível pegar todos os os gráficos gerados na análise e criar um dashboard no power BI.

Código da aula de hoje: pythoncomdados