**PROJETO FINAL**

**Análise de Dados - Telós**

**Relatório Técnico de Tratamento de Dados**

**Nome:** Lívia de Santana Pessoa

**Curso:** Formação Análise de Dados

**E-mail:** [livia.spessoa95@gmail.com](mailto:livia.spessoa95@gmail.com)

**Data de Entrega**: 31/07/2025

**Tema:** EDA e Pré-processamento de Dados: Um Estudo Aplicado ao Dataset "Pool Dados – Simulador"

**Fontes de Dados**:

A base de dados foi fornecida para fins educacionais, sem informações pessoais sensíveis, em conformidade com a LGPD.

## **Problema de Negócio e Propósito da Análise**

O projeto buscou responder à seguinte pergunta central:

**"Como podemos entender melhor o comportamento dos clientes, os fatores que influenciam sua fidelização e a efetividade das ações de marketing em um varejo regional?"**

Trata-se de um varejo localizado exclusivamente no estado de **Pernambuco**, com atuação em diversas cidades e focado na venda de produtos essenciais. Apesar do seu alcance, o negócio enfrenta desafios estratégicos e operacionais como:

* **Alto índice de clientes inativos (71%)**;
* **Sazonalidade acentuada nas vendas**, com quedas relevantes em **fevereiro, junho e setembro**;
* **Baixa conversão em campanhas de marketing**;
* **Alta taxa de reclamações encerradas sem resolução (aproximadamente 15%)**;
* **Dificuldade em identificar o perfil do cliente mais valioso e engajado**.

### **🎯 Propósito do Projeto**

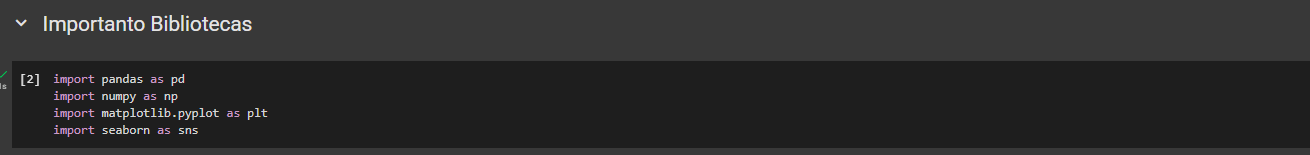
Utilizando os dados disponíveis, o projeto teve como objetivos:

* Identificar padrões de comportamento e engajamento dos clientes;
* Avaliar a efetividade e os gargalos das campanhas de marketing digital;
* Investigar a influência do **perfil demográfico** e do **score de crédito** na fidelização;
* Detectar **sazonalidades** nas compras ao longo do ano;
* Compreender os motivos e canais de **reclamações recorrentes e não resolvidas**;
* Identificar os **canais de marketing** mais eficazes para conversão.

### **📌 Hipóteses Investigadas**

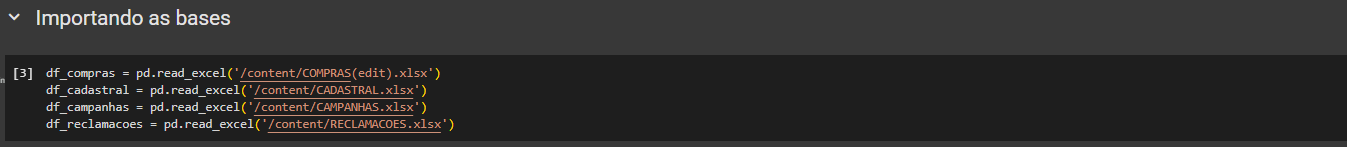
* Clientes com **score de crédito mais baixo** tendem a se tornar inativos;
* Alguns **canais de campanha** possuem desempenho de conversão muito superior a outros;
* O **perfil demográfico** (faixa etária, sexo) influencia diretamente o comportamento de compra;
* **Meses com menor volume de vendas** estão associados à sazonalidade do consumo;
* **Reclamações não resolvidas** impactam negativamente na fidelização do cliente.

Essas hipóteses guiaram toda a análise exploratória, os tratamentos aplicados, a criação de métricas e a geração do dataset final para visualizações no Power BI.

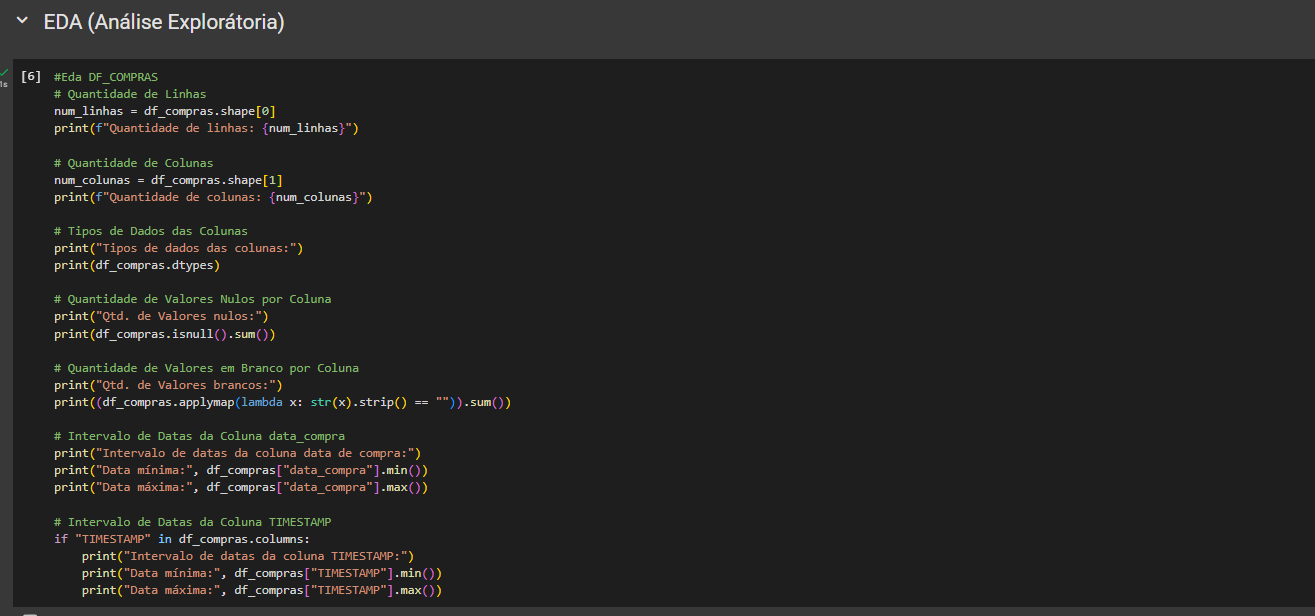


Nesta célula, foram importadas as bibliotecas essenciais para a análise exploratória e o tratamento dos dados.

Essas bibliotecas formam a base para leitura, limpeza, análise e visualização dos dados no decorrer do projeto.



Nesta célula, foram importadas quatro planilhas do Excel que compõem as bases de dados utilizadas no projeto. A função pd.read\_excel() da biblioteca **pandas** foi usada para ler cada arquivo e armazená-lo em um DataFrame separado



Nesta célula foi realizada uma análise exploratória inicial da base de dados df\_compras, com o objetivo de compreender a estrutura do dataset e identificar possíveis problemas de qualidade nos dados. As análises executadas incluem:

* **Contagem de linhas e colunas**, para entender o tamanho do dataset.
* **Verificação dos tipos de dados** de cada coluna, o que ajuda a identificar possíveis conversões necessárias.
* **Identificação de valores nulos (ausentes)**, permitindo avaliar a completude das informações.
* **Contagem de valores em branco**, que não são tecnicamente nulos, mas representam ausência de preenchimento.
* **Análise do intervalo de datas** da coluna data\_compra, informando a data mínima e máxima de registros.
* **Verificação condicional da coluna TIMESTAMP**, caso exista, também com análise do intervalo de datas correspondente.

Essas informações são fundamentais para orientar os próximos passos do tratamento de dados e garantir que as análises sejam feitas sobre uma base limpa e confiável.

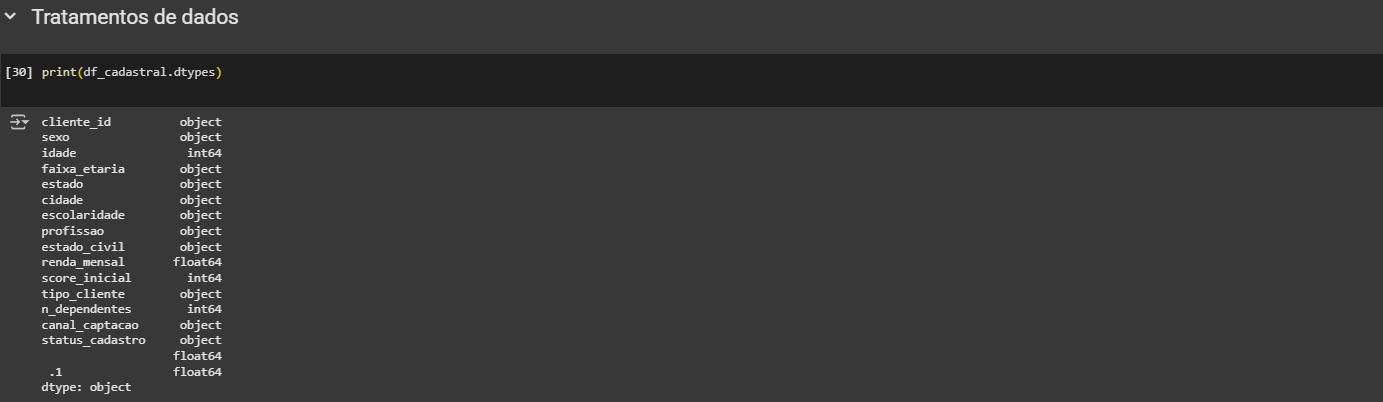
### 

### 

### **Análise Exploratória (EDA)**

A análise exploratória apresentada acima foi realizada inicialmente sobre o DataFrame df\_compras. Os mesmos procedimentos também foram aplicados aos demais DataFrames (df\_cadastral, df\_campanhas e df\_reclamacoes), com o objetivo de:

* Verificar o número de linhas e colunas;
* Identificar os tipos de dados;
* Quantificar valores nulos e valores em branco;
* Explorar colunas com datas, quando presentes.

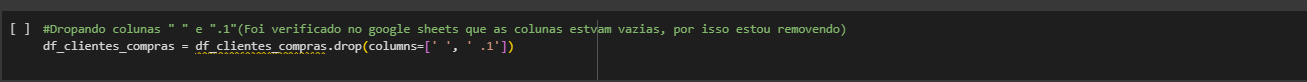


Nesta etapa foi realizada a inspeção dos tipos de dados do DataFrame df\_cadastral

### **Verificação dos Tipos de Dados**

Foi realizada a verificação dos tipos de dados em todos os DataFrames importados (df\_compras, df\_cadastral, df\_campanhas e df\_reclamacoes). Essa etapa teve como objetivo identificar o formato de cada coluna (numérico, texto, data, etc.) e detectar eventuais inconsistências, como colunas com nomes incorretos ou tipos inadequados para análise.

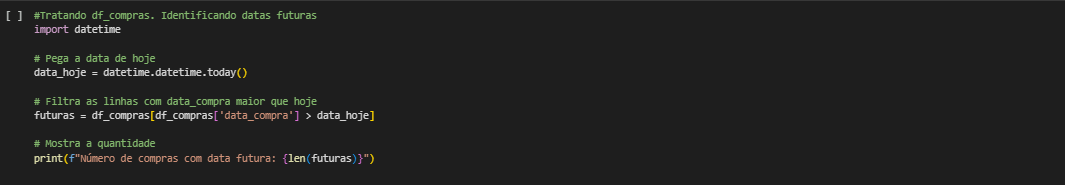
Durante essa inspeção, observou-se também a presença de colunas possivelmente criadas de forma indevida (ex: colunas com nome .1), que foram posteriormente tratadas ou removidas conforme necessário.



Nesta célula foram removidas as colunas " " (vazio) e ".1" do DataFrame df\_clientes\_compras. Essas colunas foram identificadas como **inúteis ou vazias** após verificação manual no Google Sheets.



Montou-se o Google Drive no Colab para acessar os arquivos necessários diretamente da nuvem.



Foi identificado se havia registros no df\_compras com data de compra no futuro (posterior ao dia atual). Isso ajuda a detectar possíveis erros nos dados.



Saída: 700 linhas com datas futuras.



As compras com data no futuro foram removidas do df\_compras para garantir que os dados reflitam apenas transações válidas até a data atual.



Checando se as datas foram removidas.

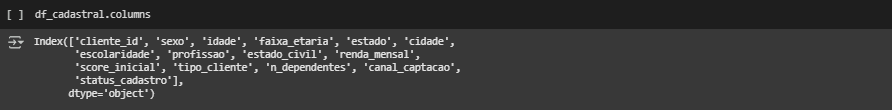


A imagem mostra os tipos de dados (dtypes) do DataFrame df\_cadastral. O objetivo era também ver as colunas e foram identificas duas colunas vazias: “ “ e “.1”.

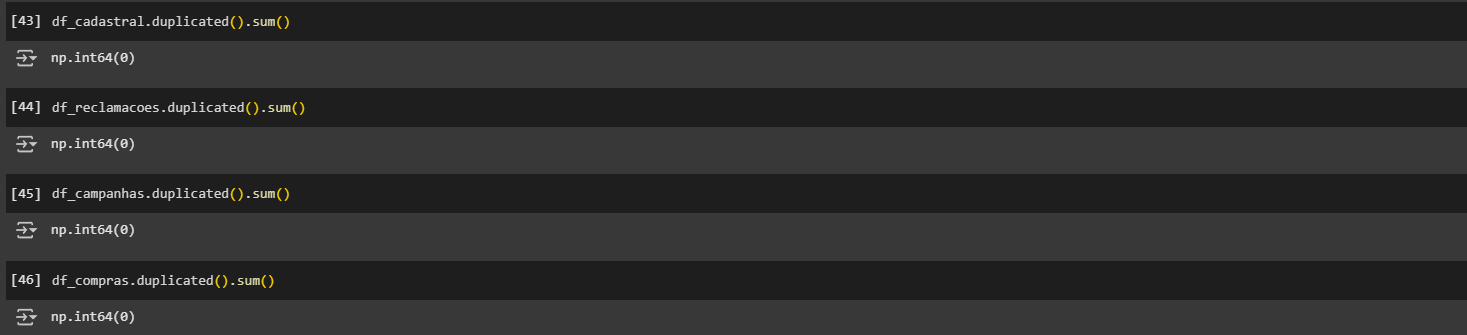


Remove as colunas " " e ".1" que estavam vazias (verificado no Google Sheets)

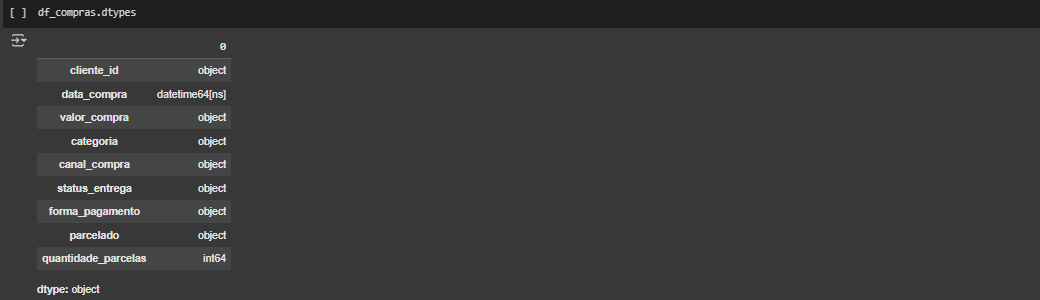
Essas colunas eram lixo da importação e não continham dados úteis.



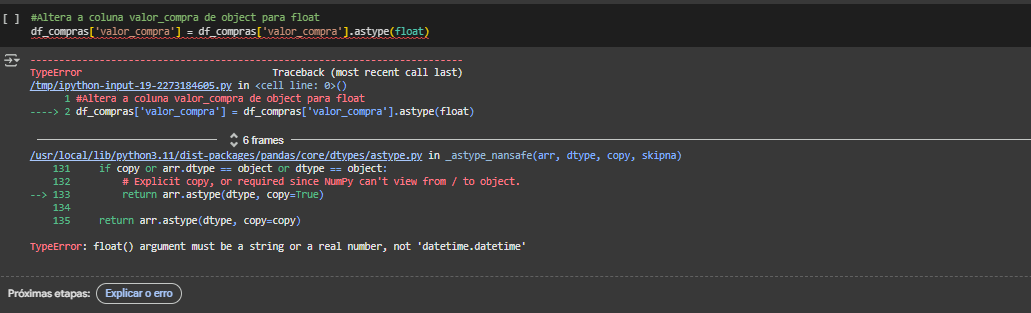
Verificando se a remoção funcionou.



Nesta etapa, foi realizada a verificação de registros duplicados em todos os DataFrames (df\_cadastral, df\_reclamacoes, df\_campanhas e df\_compras). Nenhuma duplicação foi identificada .



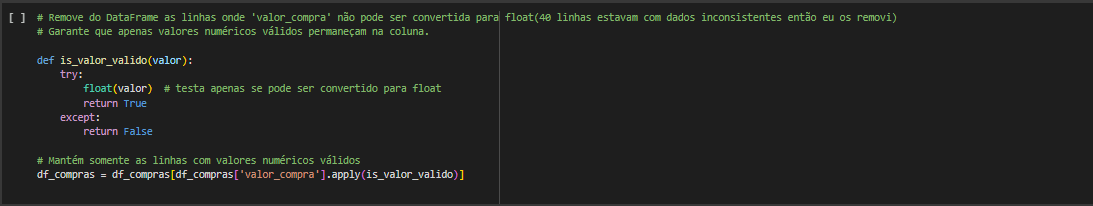
Nesta célula foi inspecionados os tipos de dados de cada coluna do df\_compras e foi identifcado que a coluna valor\_compra esta como object, o que posteriormente foi alterado para float.



Nesta célula tentei alterar o tipo da coluna valor\_compra de object para float, mas não deu certo porque a coluna continha valores do tipo datetime, e datas não podiam ser convertidas diretamente para número.

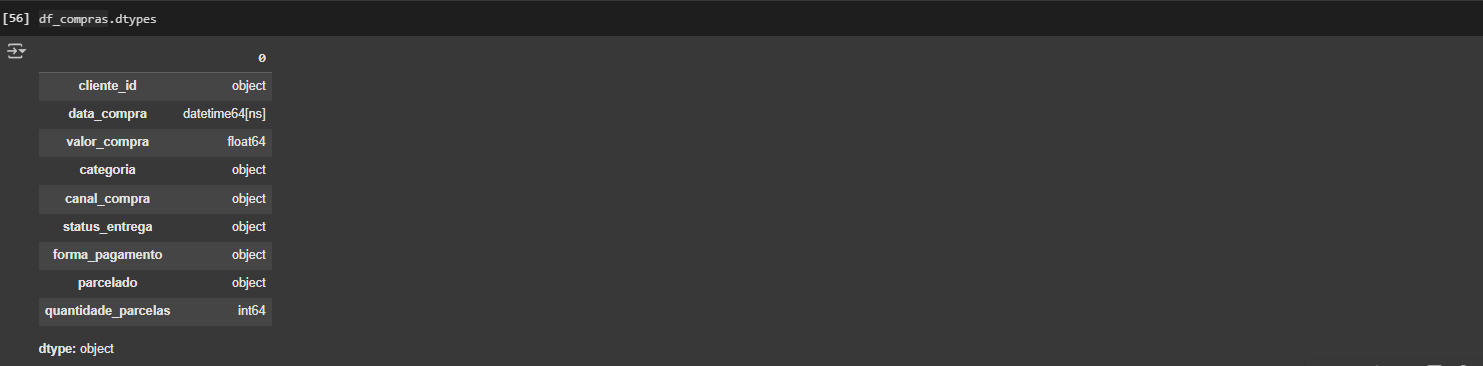


Nesta célula, investiguei a coluna valor\_compra para verificar os tipos de dados presentes, já que ela estava causando erro na conversão. Usei apply(type).value\_counts() para contar quantos valores de cada tipo havia.

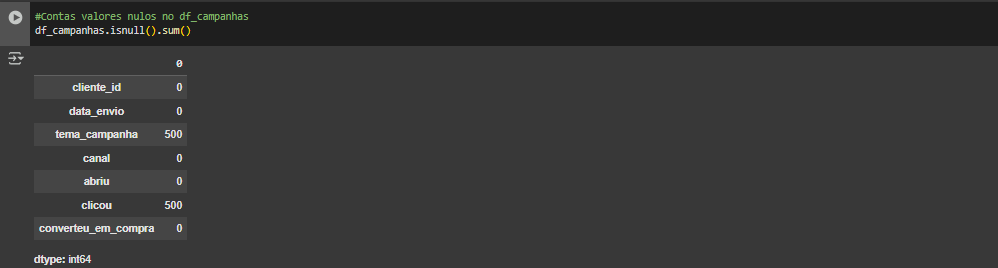
Nesta célula, criei uma função para verificar se os valores da coluna valor\_compra podiam ser convertidos para float. Removi 40 linhas com dados inconsistentes, mantendo apenas os valores numéricos válidos."



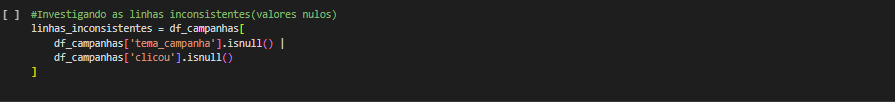
Nesta célula consegui converter a coluna valor\_compra de object para float, pois já havia removido os valores inválidos anteriormente.



Nesta célula foi realizada a verificação do tratamento da célula acima, checando se de fato os valores foi alterados para float.



Nesta célula realizei a inspeção dos valores nulos no df\_campanhas.



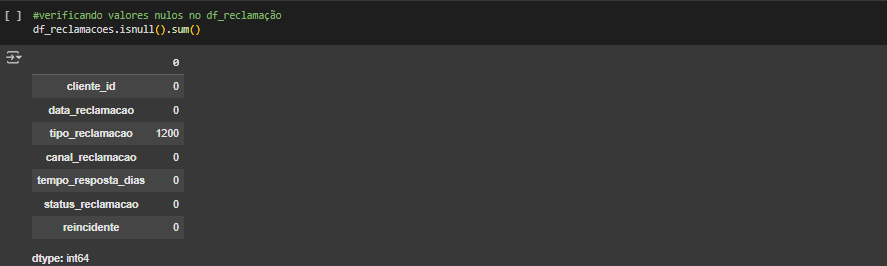
Nesta célula, foi feita uma verificação para identificar linhas da base df\_campanhas que possuem valores nulos (vazios) nas colunas 'tema\_campanha' ou 'clicou'. Essas linhas foram armazenadas na variável linhas\_inconsistentes.



Nesta célula foram dropadas as linhas inconsistentes (nulos)



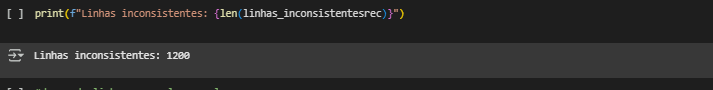
Nesta célula foi verificado se as linhas foram de fato excluídas.



Nesta célula foi verificado os valores nulos no df\_reclamação



Nesta célula foram inspecionados os valores nulos no df\_reclamaçao



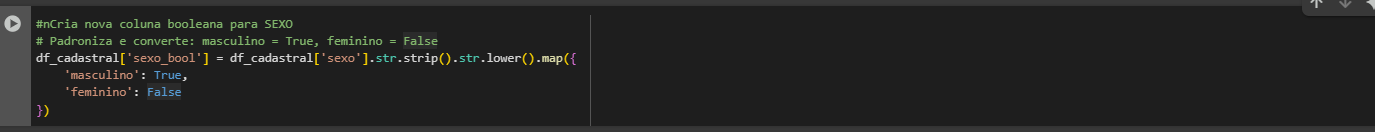
Nesta célula foram visualizadas as linhas con inconsistências (valores nulos) no df\_reclamação.



Nesta célula foram removidas as linhas com valores nulos no df\_reclamação.



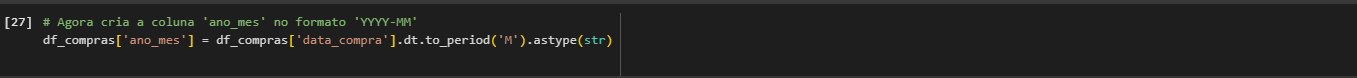
Checando se de fato foram removidas.



Nesta célula foi criada uma nova coluna booleana chamada sexo\_bool no DataFrame df\_cadastral. Os valores da coluna sexo foram padronizados (removendo espaços e convertendo para letras minúsculas) e, em seguida, mapeados: "masculino" foi convertido para True e "feminino" para False.

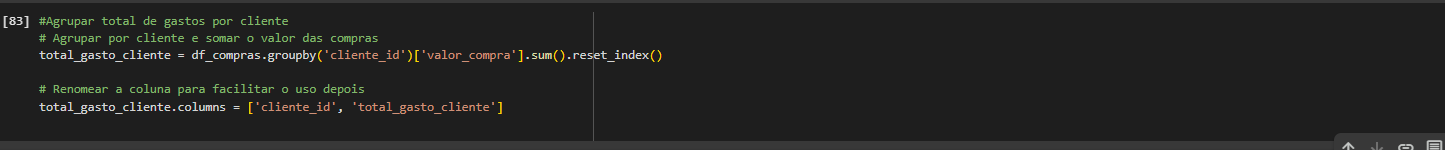


Nesta célula foram criadas três novas colunas booleanas a partir das colunas abriu, clicou e converteu\_em\_compra. Os valores de texto dessas colunas foram padronizados (removendo espaços e convertendo para minúsculas) e, em seguida, convertidos para True quando o valor era "sim" e para False quando era "não".

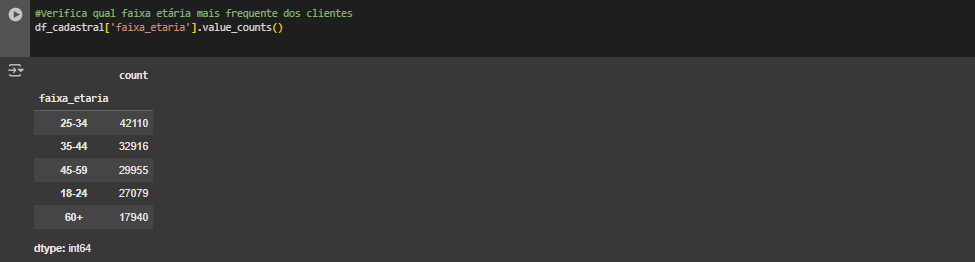


Nesta célula foi criada a coluna ano\_mes, com o objetivo de agrupar as datas no formato YYYY-MM. Para isso, a coluna data\_compra foi convertida para o tipo periodo mensal e, em seguida, transformada em string para facilitar análises e agrupamentos por mês.

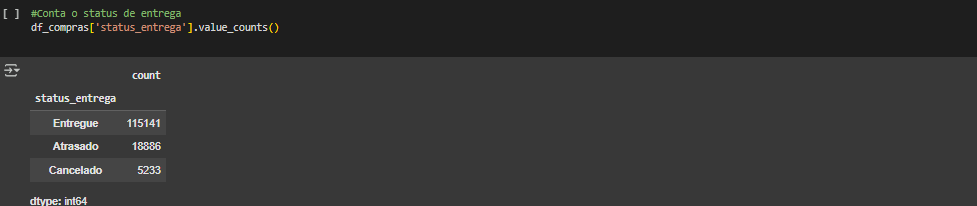
### **Agrupamentos**



Agrupou-se o DataFrame df\_compras por cliente\_id e somou-se o valor das compras de cada um.  
 Depois, a coluna foi renomeada para total\_gasto\_cliente para facilitar o uso no restante da análise



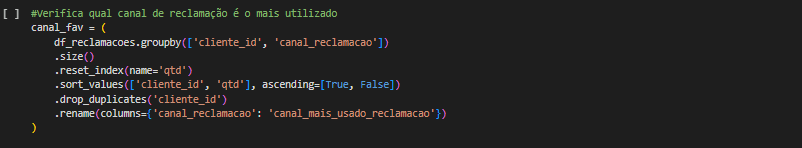
Nesta célula, foi verificada a faixa etária mais frequente entre os clientes.Para isso, foi usada a função value\_counts() na coluna 'faixa\_etaria' do DataFrame df\_cadastral, que contou quantos clientes pertencem a cada faixa. O resultado mostra que a faixa 25-34 anos é a mais comum entre os clientes.



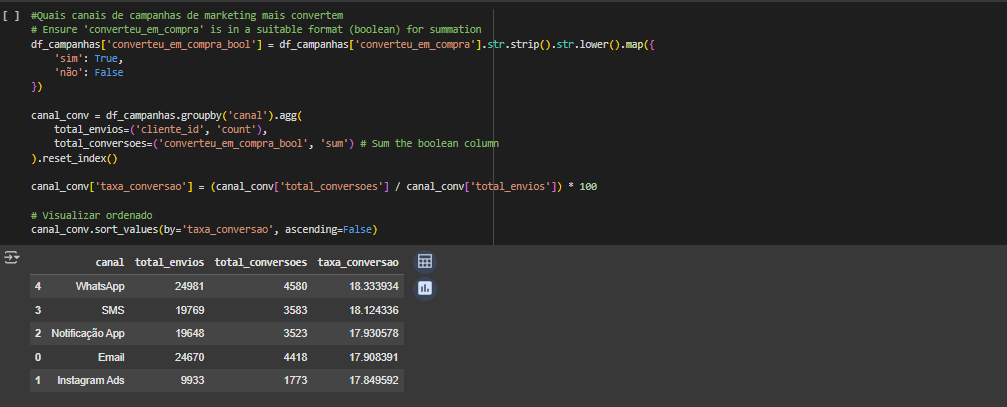
Nesta célula, foi feita a contagem da quantidade de compras por status de entrega.Foi utilizada a função value\_counts() na coluna 'status\_entrega' do DataFrame df\_compras para mostrar quantas compras foram entregues, atrasadas ou canceladas.



Nesta célula, foi calculada a quantidade de reclamações feitas por cada cliente.  
 Para isso, o DataFrame df\_reclamacoes foi agrupado pela coluna 'cliente\_id' e contou-se o número de ocorrências (reclamações) por cliente. O resultado foi armazenado no DataFrame qtd\_reclamacoes, com uma coluna chamada 'qtd\_reclamacoes'.

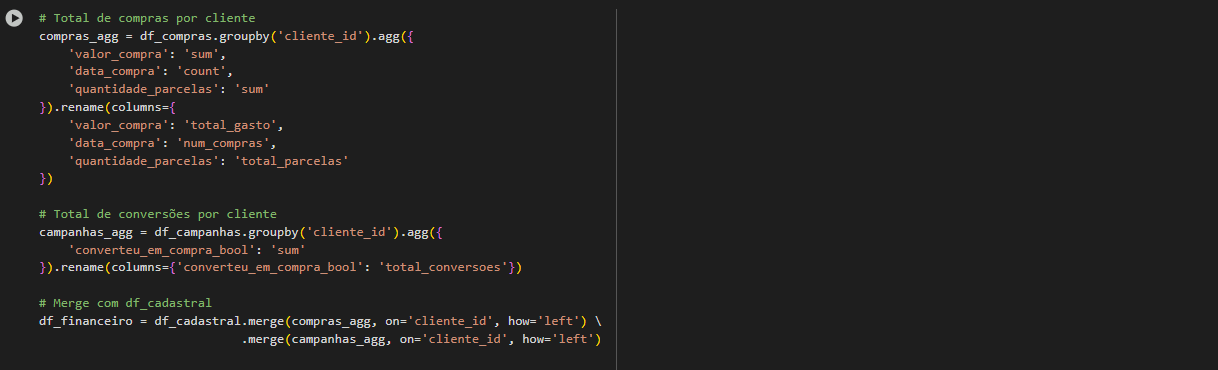


Nesta célula, foi identificado qual canal de reclamação é o mais usado por cada cliente.  
 O código contou quantas vezes cada cliente utilizou cada canal de reclamação, ordenou os resultados da maior para a menor quantidade, e selecionou apenas o canal mais frequente por cliente. Por fim, renomeou a coluna para canal\_mais\_usado\_reclamacao.

****

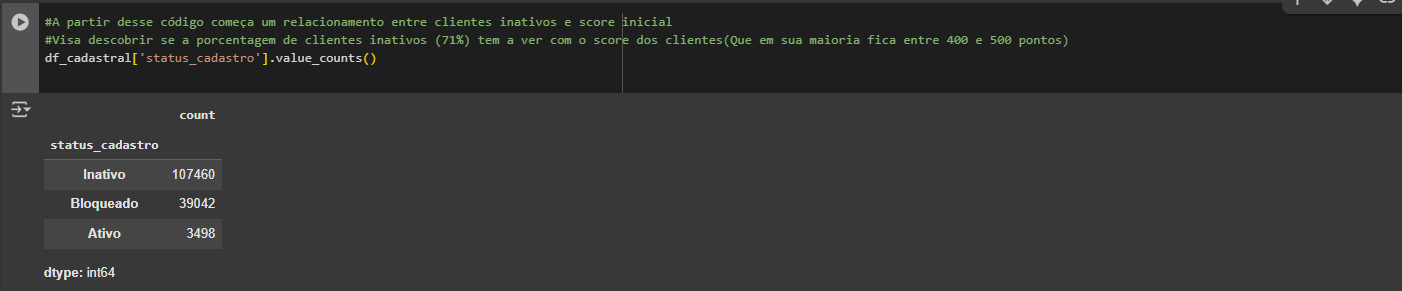
**Nesta célula, foi analisado quais canais de marketing geram mais conversões em compras.** As etapas foram:

1. **Transformou a coluna 'converteu\_em\_compra'** em valores booleanos (True para "sim" e False para "não").
2. **Agrupou os dados por canal de marketing** para calcular:  
   * Quantos envios foram feitos (total\_envios).
   * Quantas conversões ocorreram (total\_conversoes).
3. **Calculou a taxa de conversão** dividindo o total de conversões pelo total de envios e multiplicando por 100.
4. **Ordenou os canais pela taxa de conversão**, do maior para o menor.  
   O resultado mostra que o **WhatsApp** foi o canal com **maior taxa de conversão**, seguido por **SMS**.

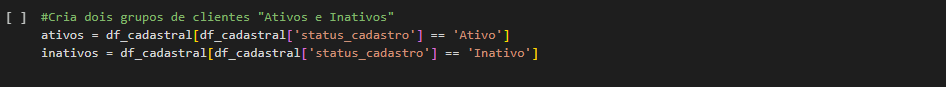


**Nesta célula, foram consolidadas informações financeiras e de conversão por cliente.** As etapas foram:

1. **Calculado o total de compras por cliente**, incluindo:  
   * Valor total gasto (total\_gasto);
   * Número de compras realizadas (num\_compras);
   * Total de parcelas acumuladas (total\_parcelas).
2. **Calculado o total de conversões por cliente**, somando os casos em que o cliente converteu em compra após uma campanha (total\_conversoes).
3. **Feito o merge dessas informações com o cadastro de clientes (df\_cadastral)**, resultando no DataFrame df\_financeiro, que reúne dados cadastrais, financeiros e de conversão de cada cliente.

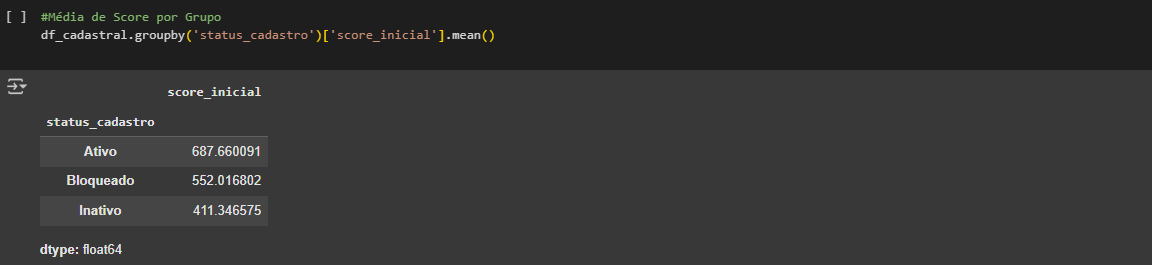


Nesta célula comecei a investigar a relação entre o status do cliente (ativo, inativo ou bloqueado) e o score inicial. Percebi que 71% estão inativos, o que pode ter ligação com scores mais baixos.

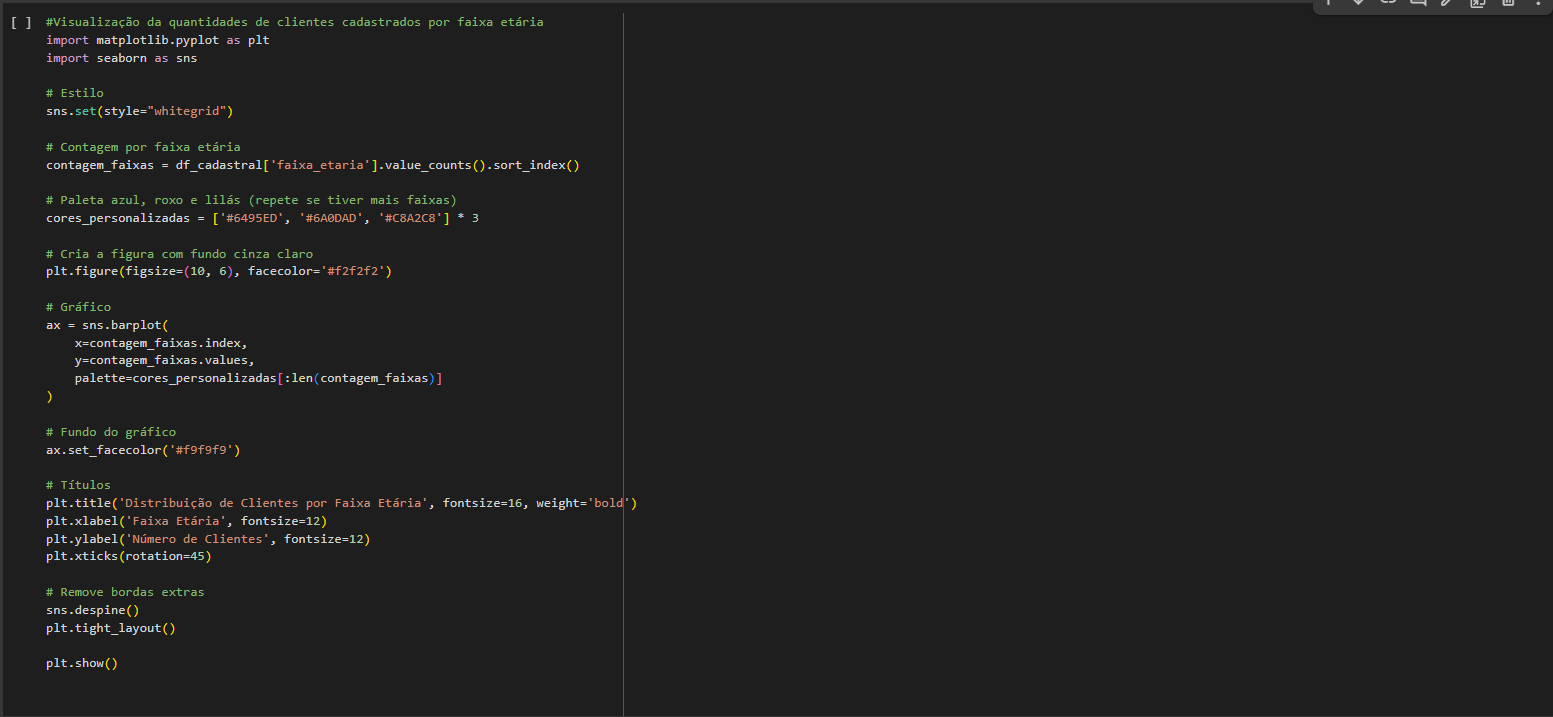


Nesta célula foi feito o seguinte:

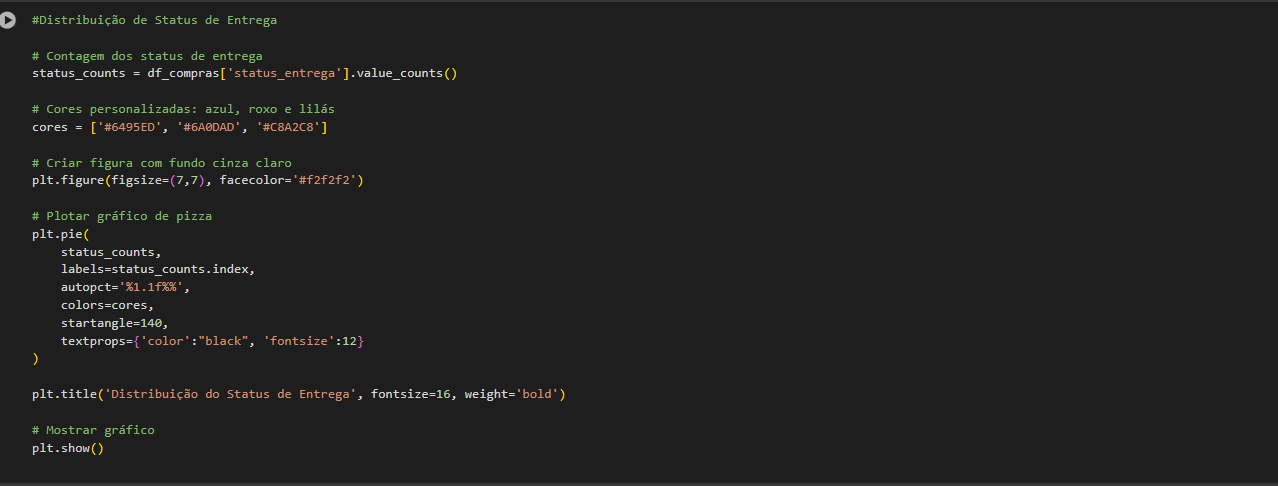
* Foi criado um grupo de clientes **ativos**, filtrando o DataFrame df\_cadastral onde a coluna 'status\_cadastro' é igual a 'Ativo'.
* Foi criado um grupo de clientes **inativos**, filtrando o mesmo DataFrame onde a coluna 'status\_cadastro' é igual a 'Inativo'.



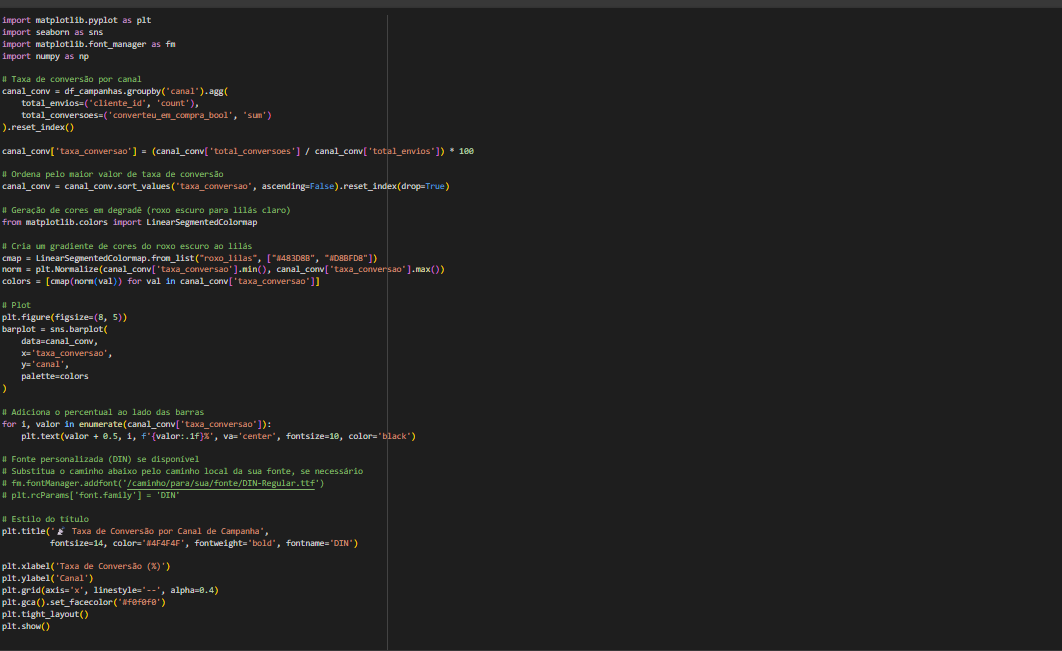
Nesta célula foi calculada a **média do score inicial** para cada grupo de status cadastral: **Ativo**, **Bloqueado** e **Inativo**



Essa célula cria um gráfico de barras que mostra a quantidade de clientes cadastrados por faixa etária.

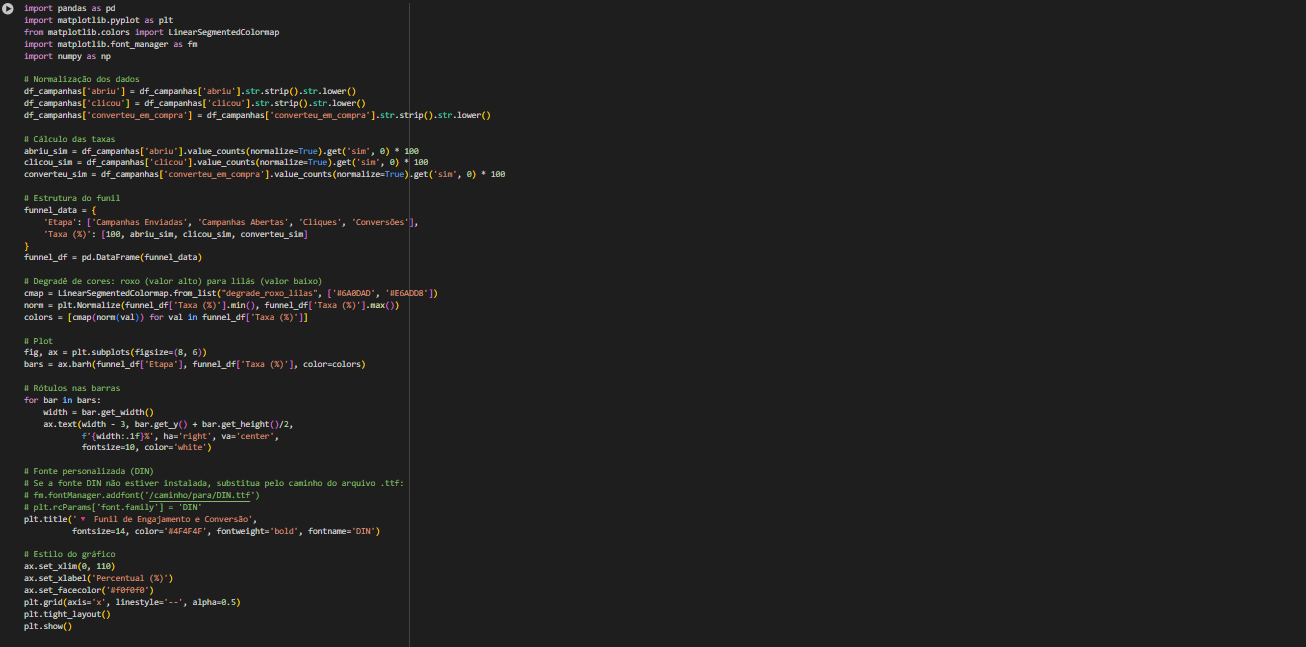


Essa célula calcula quantos pedidos existem para cada status de entrega usando value\_counts() e em seguida cria um **gráfico de pizza (pizza chart)** que mostra a **distribuição dos pedidos por status de entrega**.



Nesta célula foi feito o seguinte:

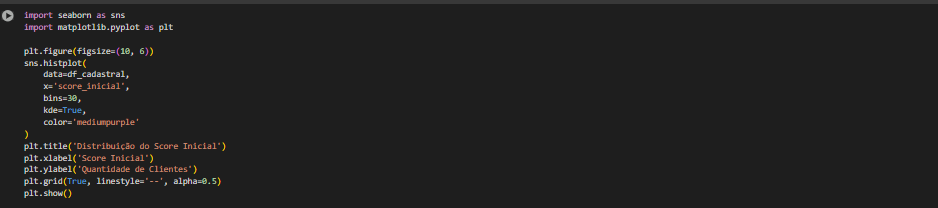
* Calculada a **taxa de conversão por canal**.
* Ordenados os canais pela maior taxa de conversão.
* Criado um **gráfico de barras** com cores em degradê do roxo escuro ao lilás.
* Adicionado o **percentual de conversão** em cada barra.
* Estilizado o gráfico com título, rótulos e fonte personalizada.



Nesta célula o código calcula a taxa de conversão por canal de origem dos clientes.  
 Processa datas, filtra dados e agrupa por canal.  
 Gera um gráfico de barras horizontal com cores em gradiente   
 Adiciona rótulos com porcentagem de conversão.  
 Personaliza o estilo e exibe o gráfico.

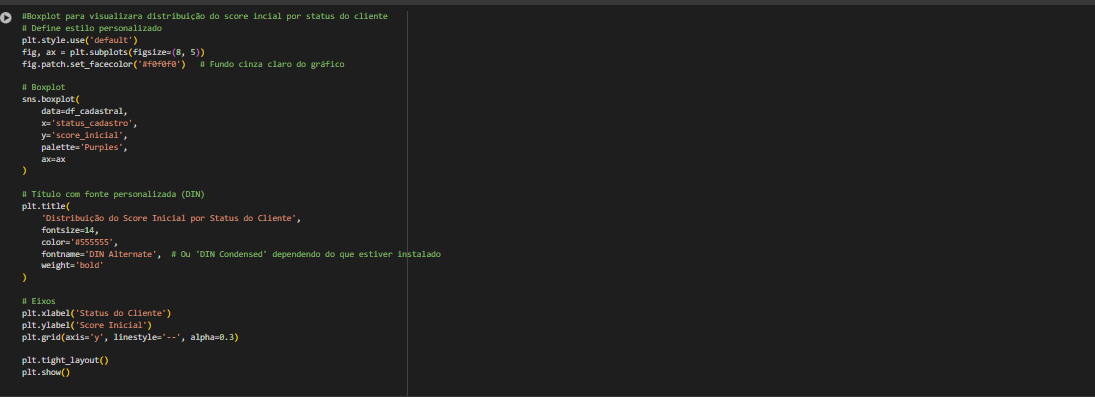


O código plota um **gráfico de barras horizontais** mostrando canais de campanha ordenados por **taxa de conversão**.  
 Cria um gradiente de cor com base na taxa (vermelho → verde).  
 Adiciona os valores nas barras e personaliza o visual.  
 Define título e rótulo do eixo X.  
 Exibe o gráfico final.



O código gera um **histograma** da coluna score\_inicial do DataFrame df\_cadastral.  
 Usa o Seaborn para visualizar a **distribuição dos scores**, com 30 bins e curva de densidade (kde).  
 Define cor roxa, título e rótulos dos eixos.  
 Adiciona grade ao gráfico.  
 Exibe o gráfico com plt.show().

Perguntar ao ChatGPT



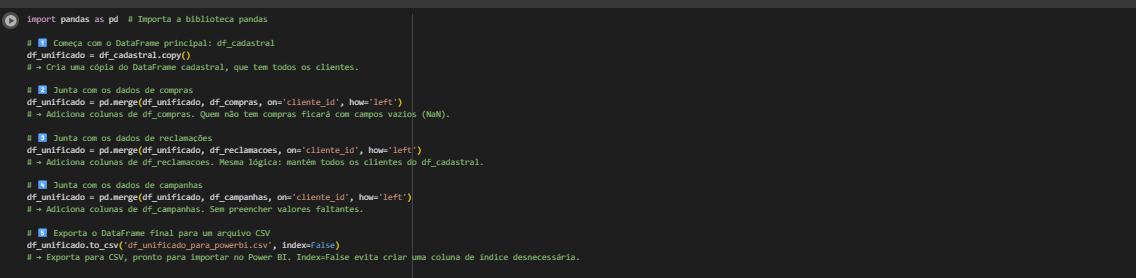
O código cria um **boxplot** para mostrar a **distribuição do score inicial por status do cliente**.

Personaliza o estilo do gráfico com fundo claro e layout ajustado.

Define título com fonte estilizada (se disponível).

Adiciona rótulos nos eixos e grade no eixo Y.

Exibe o gráfico com plt.show().

  
O código começa criando uma cópia do DataFrame principal df\_cadastral, que contém todos os clientes.  
 Em seguida, realiza três junções (merges) usando a chave cliente\_id:

1. **df\_compras**: une com o DataFrame de compras, usando how='left', ou seja, mantém todos os clientes e adiciona informações de compra (clientes sem compras terão valores nulos).
2. **df\_reclamacoes**: une da mesma forma, adicionando informações de reclamações.
3. **df\_campanhas**: une com dados de campanhas, mantendo também todos os clientes.

Por fim, o DataFrame final é exportado para um arquivo CSV, sem incluir o índice.



O código importa o módulo files do Google Colab.

Em seguida, faz o **download do arquivo CSV** gerado anteriormente (df\_unificado\_para\_powerbi.csv).

Isso permite ao usuário **baixar o arquivo localmente**.

Útil para usar os dados no Power BI ou outras ferramentas.Importando os arquivos

**RESUMO:**

Este projeto é o Trabalho de Conclusão de Curso da formação em Análise de Dados promovida pela **Telós & Neurotech**.  
 O foco da análise é um varejo de Pernambuco que atua em diversos segmentos: alimentos, utilidades domésticas, higiene e limpeza.  
 Foram analisados dados de compras, cadastros de clientes, campanhas de marketing e reclamações.

O trabalho envolveu uma **análise exploratória (EDA)** detalhada para entender os dados e suas características.  
 Houve tratamento de dados faltantes, remoção de inconsistências e padronização de tipos.  
 Foram criadas colunas booleanas e temporais para facilitar os agrupamentos e análises.

O projeto buscou responder perguntas como:

* Quais canais de campanha têm melhor taxa de conversão?
* Qual faixa etária é mais presente entre os clientes?
* Há relação entre inatividade e score de crédito?
* Como está o status de entregas e volume de reclamações?

As análises resultaram em visualizações com gráficos de barras, pizza, boxplot e funil de conversão.  
 Foi utilizada uma paleta visual personalizada (roxo, lilás e azul) para reforçar a identidade visual.  
 Todas as informações foram unificadas em um único dataset final (df\_unificado) para exportação ao Power BI.

O projeto utilizou **Python** com bibliotecas como Pandas, NumPy, Seaborn e Matplotlib, dentro do ambiente Google Colab.  
 O resultado oferece uma visão estratégica do comportamento do cliente, da eficácia de marketing e da operação do varejo.  
 Esse estudo apoia decisões baseadas em dados, permitindo ao negócio compreender melhor seus pontos fortes e fracos.  
 O arquivo final está pronto para ser explorado em ferramentas de BI.  
 Todo o processo foi documentado e estruturado para futuras análises ou reuso.

## **Resumo das Principais Problemáticas Identificadas**

### **1. Clientes com Alto Risco de Inadimplência**

A empresa enfrenta um índice elevado de inadimplência, indicando possíveis falhas na análise de crédito e nas políticas de concessão de vendas a prazo. Isso compromete o capital de giro, gera custos com cobrança e distorce o faturamento.  
 **Ações recomendadas incluem:**

* Reforçar critérios de concessão de crédito com base em score e dados históricos.
* Implementar modelos preditivos simples para análise de risco.
* Direcionar campanhas para clientes com bom histórico.
* Criar mecanismos de cobrança eficientes e incentivar modalidades pré-pagas.

### **2. Sazonalidade nas Vendas**

Meses como fevereiro, julho e setembro apresentam quedas naturais de vendas devido a fatores externos (férias, impostos, falta de datas promocionais), o que afeta o fluxo de caixa e dificulta o planejamento anual.  
 **Ações para mitigar os efeitos sazonais:**

* Desenvolver campanhas específicas para períodos de baixa.
* Replanejar investimentos de marketing ao longo do ano.
* Oferecer incentivos como cashback e promoções temáticas.
* Analisar clusters de clientes que mantêm consumo nesses meses.

### **3. Alto Volume de Reclamações e Baixa Efetividade na Resolução**

A maioria das reclamações está relacionada a atrasos na entrega e cobrança, com destaque para o SAC e WhatsApp como canais mais acionados. Quase metade dos casos não é resolvida adequadamente, indicando gargalos na gestão de atendimento.  
 **Ações estratégicas sugeridas:**

* Melhorar processos logísticos e comunicação proativa sobre atrasos.
* Automatizar a triagem nos canais digitais e ampliar a equipe do SAC.
* Estabelecer SLAs claros e indicadores de desempenho por canal.
* Investir na usabilidade dos canais digitais para incentivar o autoatendimento.

**Aprendizados e Reflexões:**

O projeto permitiu aprofundar conhecimentos em EDA, limpeza de dados e integração entre diferentes fontes. Um desafio enfrentado foi o tratamento de colunas inconsistentes e a conversão de tipos. Em projetos futuros, priorizaria a padronização prévia dos dados e adotaria uma estrutura modular de análise.

Para uma visão mais detalhada do projeto acesse meu **repositório** no **GitHub**: <https://github.com/code-liv/projeto_final>