📘 Documento Explicativo – Etapa 2: Entendimento e Coleta de Dados

Projeto: Primeiro Desafio – Criando Modelos Preditivos (MLDS)
Responsável: Time de Data Science
Versão: 1.0
Status: Concluído ✅

🎯 Objetivo da Etapa

Esta fase tem como propósito **entender profundamente os dados disponíveis** e **prepará-los** para o uso em modelos preditivos.
O foco está em garantir **qualidade, consistência e legibilidade**, transformando dados brutos em **informações estruturadas e confiáveis**.

Em outras palavras:

> "Antes de criar qualquer modelo, precisamos conhecer e confiar nos dados."

🗂️ 1. Origem dos Dados

Os dados analisados foram obtidos de diferentes fontes corporativas e externas:

Fonte	Descrição	Tipo
CSV Interno	Histórico de vendas semanais por loja	Arquivo tabular

---

🧭 2. Entendimento do Conteúdo

Após a importação do arquivo `sales.csv`, realizamos uma **análise exploratória inicial**, que envolveu:

- Identificação de **colunas existentes** e seus tipos de dado (número, texto, data etc.);

Info

```
# Checa dados convertidos Tipologia de dados
print(df.dtypes)
```

```
Date            datetime64[ns]
Weekly_Sales           float64
Holiday_Flag             int64
Temperature            float64
Fuel_Price             float64
CPI                    float64
Unemployment           float64
dtype: object
```

- Contagem de **linhas e registros válidos**;
-
    - Verificação de **valores ausentes**, **duplicados** e **padrões inconsistentes**.

Info

```python
# Visualizar estrutura geral
df.info()
# Estatísticas descritivas
df.describe()
# Verificar valores ausentes
df.isnull().sum()
# Contar duplicados
df.duplicated().sum()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 143 entries, 0 to 142
Data columns (total 7 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   Date          143 non-null    object
 1   Weekly_Sales  143 non-null    float64
 2   Holiday_Flag  143 non-null    float64
 3   Temperature   143 non-null    float64
 4   Fuel_Price    143 non-null    float64
 5   CPI           143 non-null    float64
 6   Unemployment  143 non-null    float64
```

Essas análises são o equivalente a uma **revisão de qualidade** — o que garante que os dados refletem corretamente a realidade operacional da empresa.

📓 3. Dicionário de Variáveis

Criamos um **Dicionário de Dados**, que serve como legenda para interpretação das colunas. Cada campo foi traduzido para uma linguagem simples e direta:

Variável	Significado	Importância para o Negócio
Store	Identificador da loja	Permite comparar resultados por unidade
Date	Data da semana analisada	Permite acompanhar tendência temporal
Weekly_Sales	Total de vendas semanais	Indicador principal de desempenho
Holiday_Flag	Indica se a semana teve feriado	Feriados podem aumentar ou reduzir vendas

Variável	Significado	Importância para o Negócio
Temperature	Temperatura média semanal	Influencia produtos sazonais
Fuel_Price	Preço médio do combustível	Impacta no custo logístico
CPI	Índice de preços ao consumidor	Mede aquecimento econômico regional
Unemployment	Taxa de desemprego local	Relaciona-se ao poder de compra do consumidor

---

🧹 4. Tratamento dos Dados

Nesta fase, garantimos que o conjunto de dados esteja **limpo e consistente**.
Os principais ajustes foram:

- 🔸 **Remoção de duplicidades:** eliminamos registros repetidos de uma mesma loja e data. It will show up like this:
- 🔸 **Correção de ausentes:** valores vazios foram substituídos por médias ou medianas adequadas.
- 🔸 **Padronização de formatos:** datas, números e categorias foram uniformizados.
- 🔸 **Normalização numérica:** ajustamos escalas (ex: temperatura e vendas) para facilitar cálculos pelos modelos.

> ⓘ **Info**

```
# Remover duplicados
df = df.drop_duplicates()

# Preencher valores ausentes com mediana
df = df.fillna(df.median(numeric_only=True))

# Normalizar variáveis numéricas (exemplo)
scaler = MinMaxScaler()
cols_num = df.select_dtypes(include=np.number).columns
df[cols_num] = scaler.fit_transform(df[cols_num])

# Exportar para formato parquet
table = pa.Table.from_pandas(df)
pq.write_table(table, 'ds_bruto_vendas.parquet')

print("✅ Dataset tratado e salvo como ds_bruto_vendas.parquet")
```

Foi salvo um arquivo bruto em Parquet para prosseguimento do estudo em machine learn

Essas ações asseguram que o modelo aprenda com **informações reais e comparáveis**, sem distorções.

⚙️ 5. Transformações e Preparação

Além da limpeza, aplicamos transformações que facilitam o uso posterior dos dados:

- Criação de **novas colunas derivadas** (ex: mês, ano, se é feriado);
- Conversão de tipos (número, texto, data);
- Escalonamento das variáveis numéricas para um mesmo intervalo (0 a 1).

Essas etapas são comparáveis a “organizar e etiquetar uma prateleira”, para que cada dado esteja no lugar certo antes da análise avançada.

💾 6. Armazenamento e Reuso

O resultado final desta etapa foi salvo em formato **.parquet** (nome do arquivo: `ds_bruto_vendas.parquet`).

Esse formato foi escolhido por oferecer:

- 🚀 **Leitura mais rápida** em grandes volumes de dados;
- 📦 **Compressão automática**, reduzindo tamanho do arquivo;
- 🔄 **Compatibilidade** com ferramentas de análise e visualização (como **Streamlit**, **Power BI**, **Tableau**, etc.).

Com isso, os dados já estão prontos para:

- Análise descritiva (Etapa 3),
- Modelagem estatística (Etapa 4),
- Criação de dashboards executivos (Etapa final).

🧠 7. Resultado Final

Etapa	Status	Resultado
Leitura e entendimento dos dados	✅ Concluído	Estrutura geral compreendida
Verificação de qualidade	✅ Concluído	Dados sem duplicidades
Tratamento e normalização	✅ Concluído	Dataset padronizado
Exportação	✅ Concluído	Arquivo `.parquet` pronto para modelagem

💬 Conclusão

A etapa de *Entendimento e Coleta de Dados* é o **alicerce de qualquer projeto de ciência de dados**.
Assim como um engenheiro precisa de uma fundação sólida para construir um edifício, o cientista de dados precisa de um dataset **limpo, coerente e contextualizado** para gerar previsões confiáveis.

> “Sem entender o dado, qualquer modelo é apenas um chute sofisticado.”