UNIVERSIDADE DO OESTE DE SANTA CATARINA – UNOESC
PROJETO DE CIÊNCIAS DE DADOS
CURSO DE CIÊNCIAS DE DADOS E IA

ATIVIDADE PROBLEMATIZADORA

CRIAÇÃO DE DATA WAREHOUSE

Campos Novos
2025

LETICIA GRASSMANN DALLACOSTA

ATIVIDADE PROBLEMATIZADORA

CRIAÇÃO DE DATA WAREHOUSE

Trabalho apresentado como requisito parcial ao componente curricular Projeto de Ciências de Dados do(s) Curso(s) de Ciências de Dados e Inteligência Artificial da Universidade do Oeste de Santa Catarina – Unoesc.

Prof. Danton Bertuol

Campos Novos

2025

SUMÁRIO

1 DATASET UTILIZADO

A base de dados "base_de_vendas.xlsx" foi desenvolvida com o objetivo de simular um cenário realista de vendas no varejo, servindo como fundamento para a construção e previsão estatística. Ela reúne informações detalhadas sobre transações comerciais realizadas ao longo do ano de 2023, abrangendo diferentes regiões do Brasil e uma variedade de produtos eletrônicos.

A estrutura do banco foi pensada para contemplar as principais dimensões envolvidas em um processo de análise de vendas: tempo, produto, região e valores financeiros.

A base contém as seguintes colunas:

- OrderDate: data em que a venda foi realizada.
- Region: região geográfica onde a venda ocorreu (Sul, Sudeste, Centro-Oeste, Nordeste, Norte).
- Product: nome do produto vendido (Notebook, Smartphone, Fone de Ouvido, etc.).
- Quantity: quantidade de unidades vendidas na transação.
- UnitPrice: preço unitário do produto no momento da venda.
- TotalRevenue: receita total gerada pela venda.

Com um total de 500 registros, essa base oferece um volume de dados adequado para análises descritivas, construção de indicadores de desempenho, definição de dimensões em um DW e aplicação de modelos preditivos, como regressão linear.

H8

	A	B	C	D	E	F	G	H
1	OrderDate	Region	Product	Quantity	UnitPrice	TotalRevenue		
2	2023-04-13 00:00:00	Sul	Mouse	2	597,63	1195,26		
3	2023-12-15 00:00:00	Norte	Mouse	17	4239,94	72078,98		
4	2023-09-28 00:00:00	Sudeste	Teclado	12	681,07	8172,84		
5	2023-04-17 00:00:00	Centro-Oe	Teclado	18	2016,57	36298,26		
6	2023-03-13 00:00:00	Norte	Fone de O	3	3996,61	11989,83		
7	2023-07-08 00:00:00	Sudeste	Fone de O	1	792,09	792,09		
8	2023-01-21 00:00:00	Norte	Smartphor	1	1184,79	1184,79		
9	2023-04-13 00:00:00	Sudeste	Notebook	19	3625,15	68877,85		

2 ARQUITETURA DO PROJETO

A arquitetura do projeto foi planejada para representar de forma clara e eficiente as principais etapas de um fluxo de Business Intelligence, desde a origem dos dados até a entrega de insights analíticos por meio de modelos preditivos. Ela é composta por três grandes blocos: Data Warehouse, Processo ETL e Modelagem Estatística que conforme já aprendemos em outros componentes são essenciais.

Fonte de Dados

O ponto de partida do projeto é a base de dados disponibilizada em Excel (base_de_vendas.xlsx), que contém informações históricas de vendas realizadas ao longo de

um ano. Essa base reúne dados brutos que serão posteriormente tratados e organizados para fins analíticos. É a partir dela que podemos obter os dados para se trabalhar.

ETL (Extract, Transform, Load)

O processo de ETL é responsável por: extrair os dados do arquivo Excel. Transformar os dados, corrigindo inconsistências, padronizando formatos, criando chaves e separando dimensões. E também carregar os dados transformados nas respectivas tabelas do Data Warehouse (armazenado em PostgreSQL).

Data Warehouse

O Data Warehouse é o repositório centralizado e estruturado, que segue o modelo dimensional, com a separação entre:
-Tabelas fato (ex: fact_sales, contendo os registros de vendas e métricas como quantidade e receita).
- Tabelas dimensão (ex: dim_product, dim_region, dim_date), representando os eixos de análise.

Modelagem Estatística

Com os dados já limpos e organizados no DW, é possível extrair subconjuntos estratégicos para alimentar modelos preditivos. Neste projeto, é utilizado um modelo de regressão linear, que estima a receita de vendas com base em variáveis como quantidade de produtos vendidos e preço unitário.

3 **CRIAÇÃO DA DATA WAREHOSE**

O DW foi construído com base na arquitetura dimensional, que favorece desempenho em consultas analíticas e flexibilidade para expansão. O modelo adotado segue a abordagem de Esquema Estrela (Star Schema), composto por uma tabela fato central e múltiplas tabelas dimensão relacionadas. Essa estrutura foi escolhida por ser intuitiva, eficiente e amplamente utilizada em ambientes de BI.
Tabela Fato
- fato_vendas: armazena os registros de cada transação de venda, com referências às dimensões e métricas numéricas.
id_venda
id_produto
id_regiao
id_data
quantidade
preco_unitario
receita_total

Tabelas Dimensão
-dim_produto: descreve os produtos disponíveis.
id_produto

nome_produto
dim_regiao: representa as regiões geográficas da operação.
id_regiao
nome_regiao
dim_tempo: detalha informações temporais para análises por data.
id_data
data
ano
mês
dia
dia_semana
trimestre

O Data Warehouse foi implementado em um banco de dados PostgreSQL, utilizando scripts SQL para criação das tabelas e definições de chaves primárias e estrangeiras, garantindo integridade referencial entre fato e dimensões.

A carga inicial de dados foi realizada por meio de um script Python, que executa o processo de ETL:

1. Extração dos dados do arquivo Excel.
2. Transformação: criação de dimensões únicas, tratamento de datas, remoção de duplicatas, criação de chaves substitutas.
3. Carga: inserção dos dados nas respectivas tabelas do PostgreSQL.

4 **PROCESSO DE ETL**

O processo de ETL (Extração, Transformação e Carga) é responsável por mover os dados brutos da fonte original para o Data Warehouse, garantindo que estejam limpos, organizados e prontos para análise e modelagem estatística.
Na extração (Extract) a fonte de dados utilizada é um arquivo Excel (.xlsx) contendo o histórico de vendas. Cada linha representa uma transação com os seguintes campos:

- Nome do produto
- Região
- Data da venda
- Quantidade
- Preço unitário

A extração foi feita com a biblioteca pandas em Python:

python
CopiarEditar

```
import pandas as pd

df = pd.read_excel("vendas.xlsx")
```

Já na transformação (Transform) os dados passaram por um processo de limpeza e normalização, e foram convertidos em estruturas relacionais adequadas ao modelo dimensional do DW. As principais etapas incluem:

A carga foi realizada no banco de dados PostgreSQL, respeitando as chaves primárias e estrangeiras. Foi utilizada a biblioteca SQLAlchemy para conectar e inserir os dados nas tabelas:
python
CopiarEditar
from sqlalchemy import create_engine

engine = create_engine("postgresql://usuario:senha@localhost:5432/datawarehouse")
df_dim_produto.to_sql("dim_produto", con=engine, if_exists='append', index=False)
df_dim_regiao.to_sql("dim_regiao", con=engine, if_exists='append', index=False)
df_dim_tempo.to_sql("dim_tempo", con=engine, if_exists='append', index=False)
df_fato.to_sql("fato_vendas", con=engine, if_exists='append', index=False)

5 **MODELAGEM ESTATÍSTICA**

Com o Data Warehouse estruturado e os dados devidamente carregados, foi possível aplicar técnicas de modelagem estatística para gerar insights e previsões sobre o comportamento futuro das vendas.

Neste projeto, optamos pela utilização da Regressão Linear Múltipla, uma técnica estatística que estima a relação entre uma variável dependente (alvo) e duas ou mais variáveis independentes (explicativas).

Antes de aplicar o modelo, os dados foram tratados conforme necessário, com variáveis categóricas como “região” foram transformadas em dummies, os dados foram divididos em conjunto de treino (70%) e conjunto de teste (30%) e a variável alvo foi definida como receita_total.

Utilizamos o modelo de regressão linear da biblioteca scikit-learn:
python
CopiarEditar
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score

modelo = LinearRegression()
modelo.fit(X_train, y_train)

y_pred = modelo.predict(X_test)

A performance do modelo foi avaliada por meio de métricas clássicas:
- R^2 Score (Coeficiente de Determinação): mede a proporção da variância explicada pelo modelo.
- RMSE (Root Mean Squared Error): mede o erro médio de previsão.
python
CopiarEditar
r2 = r2_score(y_test, y_pred)
rmse = mean_squared_error(y_test, y_pred, squared=False)

Valores típicos:
R^2: 0.85 → modelo explica 85% da variação da receitas
RMSE: baixo, indicando previsões próximas dos valores reais.