

Relatório Técnico - Dashboard Interativo de Análise de Sazonalidade e Previsão de Vendas com Streamlit

1. Objetivo do Projeto

Desenvolver uma aplicação interativa com Streamlit para visualizar padrões sazonais em dados de vendas e realizar previsões com base no modelo previamente treinado. O sistema permite explorar a série temporal, interpretar componentes de sazonalidade e gerar previsões personalizadas com base nas características do cliente e produto.

2. Conjunto de Dados

O sistema é projetado para trabalhar com o mesmo dataset utilizado na modelagem preditiva, contendo:

- Data do Pedido
- Valor da Venda
- Segmento, País, Cidade, Estado
- Categoria e Subcategoria do Produto

Além disso, o aplicativo permite o upload de novos datasets em formato CSV contendo as colunas: Data_Pedido e Valor_Venda.

3. Metodologia

3.1 Análise Exploratória de Dados (EDA)

- Conversão e tratamento de colunas de data e valor de venda
- Agregação mensal de vendas para criar uma série temporal
- Cálculo da média mensal de vendas
- Decomposição sazonal (tendência, sazonalidade e resíduo), desde que a série tenha pelo menos 24 meses

3.2 Engenharia de Features

- Extração automática das variáveis Ano, Mês, Dia, DiaDaSemana no processo de modelagem
- One-hot encoding das variáveis categóricas para uso no modelo
- No dashboard, a entrada do usuário é convertida de forma automática para o mesmo formato esperado pelo modelo treinado

3.3 Algoritmos Escolhidos

O modelo carregado pelo aplicativo é o melhor modelo previamente treinado (entre DecisionTreeRegressor, RandomForestRegressor e XGBoostRegressor), utilizando joblib para recuperação.

4. Avaliação do Modelo

As métricas carregadas para avaliação do modelo são:

- RMSE (Root Mean Squared Error): R\$ [valor do modelo]
- MSE (Mean Squared Error): [valor do modelo]
- R^2 (Coeficiente de Determinação): [valor do modelo]

Essas métricas são exibidas diretamente no dashboard para interpretação do usuário.

5. Importância das Variáveis

Como a aplicação utiliza o modelo treinado previamente, a importância das variáveis se mantém a mesma. Entre as mais relevantes destacam-se:

- Subcategoria
- Estado e Cidade
- Mês do Pedido
- Segmento do Cliente

A interface permite testar variações nesses parâmetros e observar o impacto nas previsões de venda.

6. Conclusões e Recomendações

- O sistema é eficaz para visualizar padrões sazonais e realizar previsões interativas com base em dados históricos
- A interface com abas separadas permite explorar tanto o dataset original quanto novos datasets carregados
- A previsão com base nas características permite simulações para planejamento comercial
- O modelo é carregado de forma transparente, e suas métricas de performance estão disponíveis

para validação

7. Limitações e Trabalhos Futuros

- A previsão depende apenas das colunas categóricas atualmente - fatores como quantidade, desconto ou preços externos não são considerados
- Para melhorar a robustez da previsão, recomenda-se a inclusão de variáveis numéricas adicionais
- A aplicação pode ser expandida com recursos como exportação dos resultados, salvamento de simulações e treinamento contínuo com novos dados

8. Referências

- [Scikit-learn documentation](#)
- [XGBoost documentation](#)
- [Streamlit documentation](#)
- Artigos e estudos sobre decomposição sazonal e previsão de demanda no varejo

9. Passo-a-passo

Passo a passo para executar o projeto

A. Requisitos:

Python 3.10 ou superior instalado

Navegador web (Chrome, Firefox, etc.)

A. Instale o venv: `python -m venv venv`

B. Se tiver executando o Terminal CMD: `venv\Scripts\activate` ou Terminal Bash: `venv/Scripts/activate`

C. Execute `pip install -r requirements.txt`

D. Execute `streamlit run app_streamlit.py`

E. Passo Opcional: Pode ser utilizado novos dataset seguindo o mesmo padrão que o dataset original ou utilizar o app "gerar_dataset.py"

Autor: Jário de Lima Silva

Data: 06/06/2025