

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS
NÚCLEO DE EDUCAÇÃO A DISTÂNCIA
Pós-graduação *Lato Sensu* em Ciência de Dados e Big Data

Murilo Ibrahim Bonomo

Provisionamento de Vendas

Belo Horizonte
2023

Murilo Ibrahim Bonomo

Provisionamento de Vendas

Trabalho de Conclusão de Curso apresentado
ao Curso de Especialização em Ciência de
Dados e Big Data como requisito parcial à
obtenção do título de especialista.

Belo Horizonte
2023

SUMÁRIO

1. Introdução	4
1.1. Contextualização	4
1.2. O problema proposto	5
1.3. Objetivos	7
2. Coleta de Dados	7
2.1 Estrutura dos Dados	8
2.1.1 Tabela de Vendas (sales)	9
2.1.2 Tabela de Clientes (costumers)	9
2.1.3 Tabela de Estoque (stock)	9
2.1.4 Tabela de Itens (items)	10
2.1.5 Tabela de Posições (positions)	10
2.2 Relacionamento entre os Datasets	10
2.3 Alterações nos Valores	10
2.4 Procedimentos Éticos	11
2.5 Data de Acesso aos Dados	11
3. Processamento/Tratamento de Dados	11
3.1 Ferramentas utilizadas	12
3.2 Consolidação de Dados	14
3.3 Ajustes na Descrição de Itens	14
3.4 Estrutura Final dos Dados	15
3.5 Verificação Final	15
4. Análise e Exploração dos Dados	16
5.1 Ferramentas Utilizadas	19
5.2 Pré-processamento dos Dados	20
5.3 Escolha dos Modelos	20
5.4 Implementação dos Modelos	21
5.4.1 Modelo ARIMA (AutoRegressive Integrated Moving Average)	21
5.4.2 Modelo de Machine Learning (Regressão Linear)	23
5.5 Avaliação dos Modelos	25
6. Interpretação dos Resultados	26
7. Apresentação dos Resultados	28
8. Links	29
REFERÊNCIAS	30
APÊNDICE	32

1. Introdução

1.1. Contextualização

Nos dias de hoje, a gestão eficaz de estoque é uma preocupação fundamental para empresas de todos os setores e tamanhos. Manter o equilíbrio entre a oferta e a demanda de produtos é essencial para garantir a satisfação do cliente, reduzir custos operacionais e maximizar os lucros. Para enfrentar os desafios de um mercado dinâmico e altamente competitivo, as organizações recorrem à Ciência de Dados como uma ferramenta poderosa para otimizar suas operações.

A previsão de vendas e a otimização de estoque são componentes vitais dessa equação. A capacidade de prever com precisão a demanda futura de produtos e, ao mesmo tempo, manter níveis de estoque adequados é uma busca contínua que visa melhorar a eficiência operacional e atender às expectativas dos clientes.

No cenário econômico atual, as empresas enfrentam desafios constantes de flutuações na demanda, sazonalidades, concorrência acirrada e pressões econômicas. Portanto, a aplicação da Ciência de Dados na previsão de vendas e otimização de estoque não é apenas uma questão de vantagem competitiva, mas uma necessidade premente para a eficiência operacional e o atendimento às demandas da sociedade.

Este estudo tem como objetivo explorar as técnicas e abordagens disponíveis para a previsão de vendas e otimização de estoque e, ao fazê-lo, oferecer insights valiosos tanto para as empresas quanto para a sociedade em geral. A otimização do estoque não apenas promete reduzir os custos operacionais e melhorar a eficiência, mas também contribuir para a sustentabilidade e a estabilidade da cadeia de suprimentos em um mundo em constante mudança.

1.2. O problema proposto

Why (Por que esse problema é importante)?

Este problema é de extrema importância, uma vez que afeta diretamente a eficiência operacional e a satisfação do cliente nas empresas. A gestão inadequada de estoque pode resultar em custos excessivos, rupturas de estoque e insatisfação do cliente. Além disso, em um contexto mais amplo, a otimização de estoque contribui para a eficiência na utilização de recursos naturais e para a redução do impacto ambiental.

Who (De quem são os dados analisados)?

Os dados analisados provêm de uma variedade de fontes, incluindo registros de vendas internas, histórico de estoque da empresa, informações de fornecedores e, em alguns casos, dados do mercado e econômicos. Esses dados podem ser coletados de empresas de diversos setores, como varejo, manufatura, distribuição, saúde e muitos outros.

What (Quais os objetivos com essa análise)?

Os principais objetivos desta análise são:

- Prever com precisão a demanda futura de produtos ou serviços.
- Otimizar a gestão de estoque, mantendo os níveis de estoque adequados para atender à demanda.
- Reduzir custos operacionais, minimizando estoques em excesso.
- Melhorar a satisfação do cliente, garantindo que os produtos estejam disponíveis quando necessários.
- Contribuir para a eficiência na utilização de recursos naturais e para a sustentabilidade ambiental.

Where (Trata dos aspectos geográficos e logísticos de sua análise)?

A análise pode ser aplicada a empresas em todo o mundo, independentemente de sua localização geográfica. Ela leva em consideração os aspectos logísticos associados ao reabastecimento de estoque e às operações de distribuição.

When (Qual o período está sendo analisado)?

O período de análise pode variar dependendo das necessidades da empresa, mas geralmente abrange um histórico de vendas e estoque de vários meses a anos. A análise também pode ser contínua, com previsões de curto prazo e otimização do estoque em tempo real.

Em resumo, o problema proposto envolve a aplicação da Ciência de Dados para prever com precisão a demanda de produtos ou serviços, otimizar a gestão de estoque e, assim, contribuir para a eficiência operacional, a satisfação do cliente e a sustentabilidade ambiental em empresas de diversos setores e localizações geográficas.

1.3. Objetivos

Este trabalho tem como objetivo principal aplicar a Ciência de Dados na previsão de vendas e otimização de estoque, visando atingir os seguintes objetivos específicos:

1. **Desenvolver Modelos Precisos de Previsão de Vendas:** Criar modelos de previsão de vendas que se baseiam em dados históricos e variáveis relevantes, proporcionando previsões precisas e confiáveis.
2. **Otimizar os Níveis de Estoque:** Utilizar as previsões de vendas para otimizar a gestão de estoque, determinando os níveis ideais de estoque, pontos de pedido e políticas de reabastecimento.
3. **Reduzir Custos Operacionais:** Minimizar os custos operacionais associados ao estoque, incluindo custos de armazenamento, custos de pedido e custos de transporte, resultando em economias significativas para a empresa.
4. **Melhorar a Satisfação do Cliente:** Garantir a disponibilidade de produtos quando necessário, reduzindo as rupturas de estoque e melhorando a satisfação do cliente.
5. **Contribuir para a Sustentabilidade e Eficiência Ambiental:** Promover a utilização eficiente de recursos naturais, reduzindo o desperdício de produtos e embalagens, o que contribui para práticas comerciais mais sustentáveis.
6. **Adaptar-se a Mudanças no Ambiente de Negócios:** Desenvolver modelos flexíveis que possam se adaptar a mudanças na demanda, sazonalidades e eventos inesperados, como crises econômicas ou desastres naturais.
7. **Facilitar a Tomada de Decisão Estratégica:** Fornecer informações e insights valiosos para a alta administração e equipes de gerenciamento, auxiliando na tomada de decisões estratégicas relacionadas à gestão de estoque e operações.
8. **Comunicar Resultados de Forma Clara e Acessível:** Apresentar os resultados de maneira clara e acessível, por meio de relatórios, painéis de controle e visualizações de dados que possam ser facilmente compreendidos por partes interessadas.

2. Coleta de Dados

Nesta seção, serão detalhados a origem, o formato e a estrutura dos dados utilizados no estudo. Os dados utilizados foram gentilmente cedidos pela Robert Bosch LTDA, sendo essenciais para a pesquisa de previsão de demanda. Esses dados consistem em um conjunto de 23 datasets, a série de dados coletada ao longo do ano de 2022/2023 em formato .csv.

Variáveis	-	Descrição
DB	-	Database
TB	-	Table
MAT	-	Material
CST	-	Costumer
STK	-	Stock
SLS	-	Sales
NUM	-	Number
DSC	-	Description
HGT	-	Height
LEN	-	Length
WID	-	Width
WGT	-	Weight
UOM	-	Unity of Messuare
VOL	-	Volume
QTY	-	Quantity
VLZ	-	Velzon
VLE	-	Value
TME	-	Time
TYP	-	Type
HU	-	Handling Unit

2.1 Estrutura dos Dados

Os conjuntos de dados consistem em várias tabelas, cada uma desempenhando um papel específico na análise. Abaixo, apresentamos uma visão detalhada de cada tabela:

2.1.1 Tabela de Vendas (sales)

Esta tabela contém informações sobre transações de vendas e é crucial para a análise de previsão de demanda. Principais campos:

- sls_dlv: Número do pedido.
- sls_typ: Tipo do pedido.
- cod_mat: Código do material.
- sls_qty: Quantidade do material no pedido.
- sls_vle: Valor do pedido.
- sls_day: Data do pedido.
- cst_cod: Código do cliente.

2.1.2 Tabela de Clientes (customers)

Esta tabela relaciona-se com a tabela de vendas pelo código do cliente (cst_cod). Principais campos:

- cst_cod: Código do cliente.

2.1.3 Tabela de Estoque (stock)

Relaciona-se à tabela de Posições pelo código da posição (bin_num). Principais campos:

- bin_num: Número da posição.
- bin_typ: Tipo da posição.
- stk_hu: Unidade Comercial.
- mat_cod: Código do material.
- stk_qty: Quantidade em estoque.

2.1.4 Tabela de Itens (items)

Relaciona-se à tabela de vendas pelo código do material (cod_mat). Principais campos:

- mat_cod: Referência do material.
- cod_mat: Código do material.
- mat_dsc: Descrição do material.

2.1.5 Tabela de Posições (positions)

Relaciona-se com a tabela de estoque pelo número da posição (bin_num). Principais campos:

- bin_num: Número da posição.
- bin_typ: Tipo da posição.

2.2 Relacionamento entre os Datasets

As tabelas estão relacionadas principalmente pelos seguintes campos:

- cst_cod: Código do cliente (tabelas de vendas e clientes).
- cod_mat: Código do material (tabelas de vendas, estoque e itens).
- bin_num: Número da posição (tabelas de estoque e posições).

Essas chaves são essenciais para integrar os dados e realizar análises conjuntas.

2.3 Alterações nos Valores

Com o objetivo de preservar a confidencialidade, todos os valores monetários foram ajustados proporcionalmente, mantendo relações e padrões nos dados.

2.4 Procedimentos Éticos

A obtenção, manipulação e análise dos dados seguiram padrões éticos rigorosos, em conformidade com as diretrizes éticas e legais estabelecidas para pesquisa com dados empresariais.

2.5 Data de Acesso aos Dados

Os dados foram obtidos da empresa em Outubro/2023. Qualquer atualização ou revisão posterior será registrada.

A tabela de descrição de campos/colunas foi ampliada para incluir os campos específicos de cada tabela, conforme a necessidade.

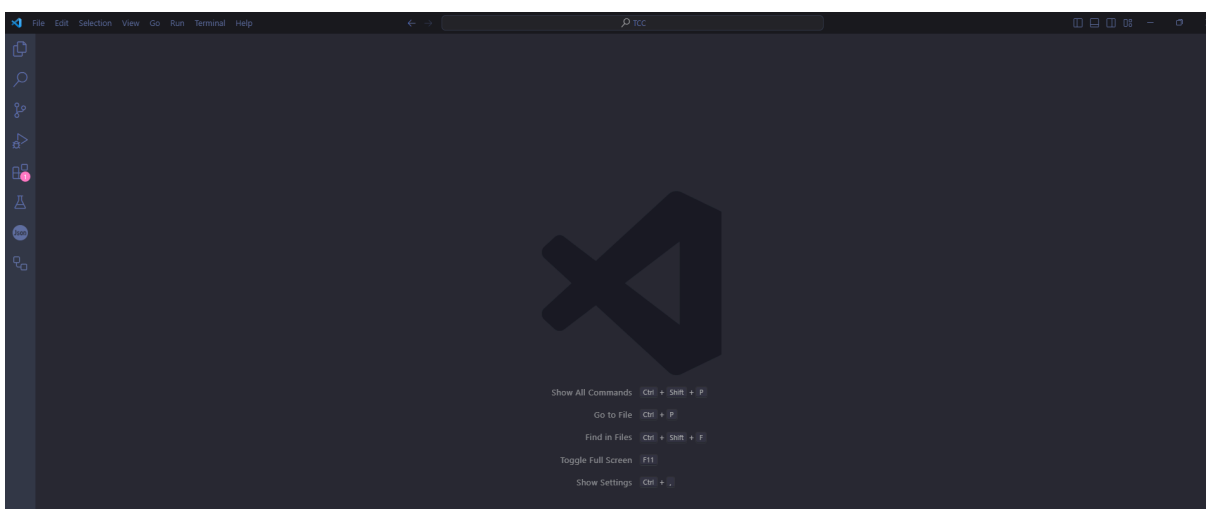
3. Processamento/Tratamento de Dados

Salienta-se que todas as informações fornecidas foram cedidas pela Robert Bosch LTDA, os dados originais sofreram alterações nos valores para preservar a confidencialidade do faturamento real nossas intervenções foram mínimas, limitando-se principalmente à consolidação mensal e ajustes pontuais.

O conjunto de dados obtidos teve que passar por algumas etapas de tratamento, transformação, como por exemplo, formatação de data e ajuste dos números dos clientes e junção dos datasets. Posteriormente o desenvolvimentos das análises, teoria e modelos. Durante esse processo, precisamos corrigir os tipos de dados, tratamento de valores nulos ou formato das informações.

3.1 Ferramentas utilizadas

A seleção das ferramentas de desenvolvimento desempenha um papel fundamental no processo de implementação, no contexto deste trabalho de pesquisa foi selecionado o Microsoft Visual Studio Code versão 1.8, para a elaboração dos scripts em Python junto com o Project Jupyter para plot de algumas imagens de provisionamento. Para uma análise visual dos dados foi utilizado o Power BI. Além disso, destaca-se a execução do Python na versão 3.7, indicando a preferência por uma versão recente e amplamente adotada da linguagem, proporcionando acesso às últimas funcionalidades e melhorias de desempenho, tendo sido utilizadas as bibliotecas listadas na figura 3.1.



Inicialmente importamos a biblioteca Pandas, que é uma biblioteca de código aberto, que fornece as estruturas fundamentais para as operações de análise e manipulação de dados.

```
import pandas as pd

import matplotlib.pyplot as plt

from sklearn.model_selection import train_test_split

from sklearn.linear_model import LinearRegression

from sklearn.metrics import mean_squared_error

import numpy as np
```

3.2 Consolidação de Dados

Recebemos os dados mensalmente em arquivos CSV, cada um representando um mês de vendas. Para facilitar a análise, consolidamos todos esses arquivos em um único conjunto de dados de vendas.

```
import os
import pandas as pd

# Diretório onde estão os arquivos CSV
diretorio = "D:/Projetos/TCC/Arquivos Base/Vendas/"

# Lista para armazenar DataFrames de todos os arquivos CSV
dataframes = []

# Percorra os arquivos no diretório
for nome_arquivo in os.listdir(diretorio):
    # Verifique se o arquivo tem extensão .csv
    if nome_arquivo.endswith(".csv"):
        # Leia o arquivo CSV e adicione-o à lista de DataFrames
        caminho_arquivo = os.path.join(diretorio, nome_arquivo)
        df = pd.read_csv(caminho_arquivo, converters = {"HR":str, "REMESSA":str, "CLIENTE":str, "MATERIAL":str})
        dataframes.append(df)

# Concatene os DataFrames em um único DataFrame
vendas_total = pd.concat(dataframes, ignore_index=True)

# Salve o DataFrame resultante em um arquivo "Vendas_Total.csv"
caminho_saida = os.path.join(diretorio, "Vendas_Total.csv")
vendas_total.to_csv(caminho_saida, index=False)
```

3.3 Ajustes na Descrição de Itens

Identificamos que algumas descrições de itens continham vírgulas, o que poderia interferir na correta leitura do arquivo CSV. Portanto, realizamos ajustes removendo as vírgulas dessas descrições.

Seguindo no tratamento dos dados, foi retirado os 0 (zero) a esquerda dos números do pedidos

Passo seguinte foi alterar o formato das datas para DD-MM-YYYY:

```
dados['sls_day'] = pd.to_datetime(dados['sls_day']).dt.date
```

Por fim, verificar se há dados nulos e duplicatas

3.4 Estrutura Final dos Dados

Após as tratativas mencionadas, criamos o database DB_TCC.db contendo os dados dos 5 principais datasets, a estrutura final dos dados consiste nos seguintes conjuntos:

- **Tabela de Vendas (sales):** Consolidada, contendo informações mensais de vendas.
- **Tabela de Clientes (costumers):** Dados de clientes mantidos conforme recebidos.
- **Tabela de Estoque (stock):** Dados de estoque sem alterações significativas.
- **Tabela de Itens (items):** Descrições de itens ajustadas para facilitar a leitura do CSV.
- **Tabela de Posições (positions):** Informações sobre posições de estoque mantidas conforme recebidas.

3.5 Verificação Final

Realizamos uma verificação final para garantir a consistência dos dados após as intervenções mencionadas. Não foram identificados problemas adicionais, e os conjuntos de dados estão prontos para serem utilizados nas análises subsequentes.

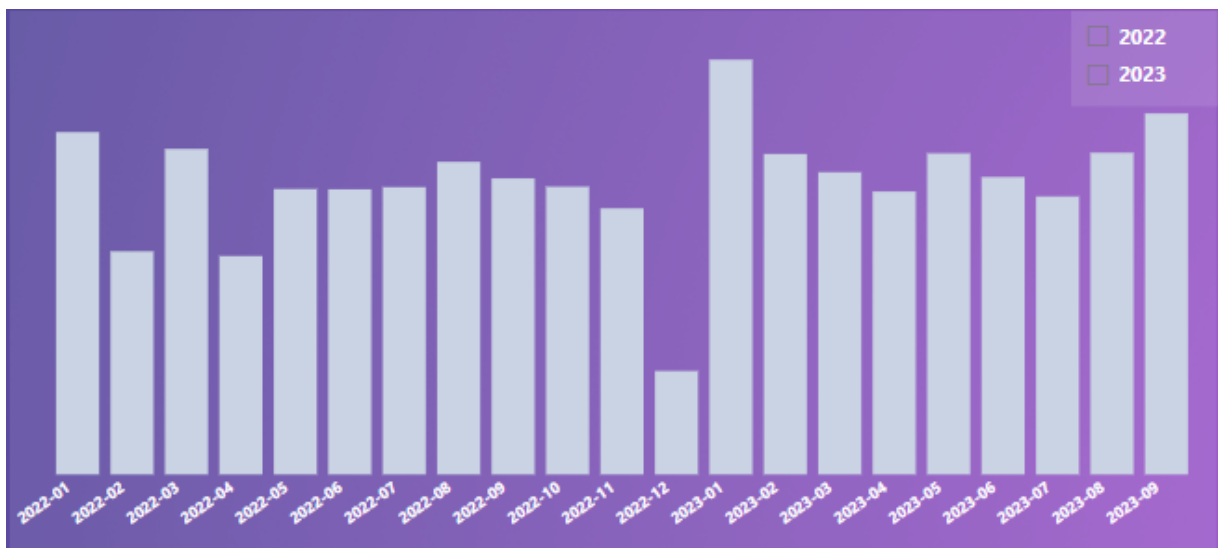
Esta seção destaca a transparência e a simplicidade no processamento e tratamento dos dados, ressaltando que a base fornecida já estava em bom estado e as intervenções foram mínimas.

4. Análise e Exploração dos Dados

Nessa seção você deve mostrar como foi realizada a análise e exploração dos dados do seu trabalho. Mostre as hipóteses levantadas durante essa etapa e os padrões e *insights* identificados.

Com a base de dados extraída foram feitas as seguintes análises:

- Tanto no ano de 2022 quanto no ano de 2023 o mês no qual foi feito mais vendas foi Janeiro, porém, é notável que o ano de 2023 o mês de Setembro também superou Janeiro/2022



- A base de vendas também nos mostra que mais que 50% dessas vendas são para o mercado nacional de clientes.

Tipo	Valor
Exp. Clientes Indust	R\$ 27.592.257,11
Nac. Clientes Indust	R\$ 146.496.542,01
Exportação	R\$ 160.416.750,57
Nacional OES	R\$ 329.216.204,26
Nacional	R\$ 1.736.272.499,16

- Com as análises feitas no Power BI também é possível analisarmos quais foram os itens mais vendidos, tanto em valor financeiro quanto em quantidade de peças:

Top 15 Items mais Vendidos em Quantidade		
Código do Material	Descrição do Material	Total de peças
0204032339HT8	Brake Fluid	2.002.076
0204032339W43	Brake Fluid	941.160
0204032343HT8	Brake Fluid	1.954.192
0280156403741	Injection Valve	413.073
0414701070741	Unit Injector System	8.760
044501053339V	Radial Piston Pump	8.140
0445110520C3R	Injector	33.272
044511064774C	Injector	28.278
09864355089B5	Injector	29.462
F00099C012EWB	Ht Ignition Cable	269.762
F00099C125EWB	Ht Ignition Cable	332.655
F000KE0P30T4E	Spark Plug	1.456.482
F000TE159AFP2	ELECTRIC FUEL PUMP	722.398
F000Z502107ZB	IGNITION COIL	453.762
F006WM03107U6	Cooling Fan	84.468

Top 15 Items mais Vendidos em Quantidade		
Código do Material	Descrição do Material	Total de peças
0204032339HT8	Brake Fluid	2.002.076
0204032339W43	Brake Fluid	941.160
0204032343HT8	Brake Fluid	1.954.192
0280156403741	Injection Valve	413.073
0414701070741	Unit Injector System	8.760
044501053339V	Radial Piston Pump	8.140
0445110520C3R	Injector	33.272
044511064774C	Injector	28.278
09864355089B5	Injector	29.462
F00099C012EWB	Ht Ignition Cable	269.762
F00099C125EWB	Ht Ignition Cable	332.655
F000KE0P30T4E	Spark Plug	1.456.482
F000TE159AFP2	ELECTRIC FUEL PUMP	722.398
F000Z502107ZB	IGNITION COIL	453.762
F006WM03107U6	Cooling Fan	84.468

- A última análise feita, foi baseada nos clientes, nesse caso foram os clientes que mais emitiram pedidos e os que mais compraram em reais:

Top 15 Clientes em R\$	
cst_cod	Valor de Vendas
924900	R\$ 82.697.161,42
1774300	R\$ 39.071.141,93
16610000	R\$ 35.760.434,20
268500	R\$ 33.541.519,58
493900	R\$ 32.073.142,33
58017	R\$ 29.522.717,10
18196504	R\$ 24.982.982,67
1564301	R\$ 23.785.325,21
3374310	R\$ 23.709.888,41
8899900	R\$ 23.525.492,46
58007	R\$ 23.344.167,05
924934	R\$ 22.770.374,17
8580008	R\$ 22.504.419,22
16325200	R\$ 22.482.340,92
683802	R\$ 20.928.797,45

Top 15 Clientes por Pedido	
cst_cod	Contagem de sls_dlv
1283502	1693
16235200	1221
16325200	1503
1774300	1255
3097900	1238
3374310	1330
466201	1170
58001	1225
58004	1290
58009	1222
58013	1199
58017	1369
686303	2434
866302	1148
924900	3693
924934	1264

5. Criação de Modelos de Machine Learning

O aprendizado de máquina (Machine Learning) é o estudo de algoritmos de computador que melhoram automaticamente através da experiência e do uso de dados. Pode ser classificado em: a) aprendizado supervisionado, em que o computador é apresentado com exemplos de entradas e suas saídas desejadas, rotuladas, e o objetivo é aprender uma regra geral que mapeia entradas em saídas; b) aprendizado não supervisionado, em que nenhum rótulo é dado ao computador, deixando-o sozinho para encontrar a estrutura em sua entrada. O aprendizado não supervisionado pode ser um objetivo em si (descobrir padrões ocultos nos dados) ou um meio para um fim (aprendizado de recursos); e c) aprendizado por reforço, em que o computador interage com um ambiente dinâmico no qual deve realizar uma determinada tarefa. À medida que navega no espaço do problema, o programa recebe um feedback, que ele tenta maximizar. (LISBOA, Henrique Borsatti, 2021).

O objetivo do estudo é identificar os padrões de vendas anteriores a fim de antecipar as vendas futuras, visando o abastecimento eficaz do estoque. Utilizamos a linguagem de programação Python, juntamente com as bibliotecas pandas, scikit-learn e statsmodels. A escolha do Python se deve à sua flexibilidade e robustez no contexto de aprendizado de máquina.

5.1 Ferramentas Utilizadas

- **Linguagem de Programação:** Python 3.7
- **Bibliotecas Principais:**
 - pandas: Para manipulação de dados e pré-processamento.
 - scikit-learn: Para implementação e avaliação de modelos de Machine Learning.
 - statsmodels: Para a utilização de modelos estatísticos, como ARIMA.

5.2 Pré-processamento dos Dados

Antes de criar os modelos, realizamos algumas etapas de pré-processamento nos dados de vendas. Isso incluiu a agregação dos dados por data e a soma do valor de vendas para cada mês.

```
import pandas as pd

# Carregamento dos dados de vendas

file_path = "D:/Projetos/TCC/Arquivos
Base/Vendas/Vendas_Total.csv"

vendas = pd.read_csv(file_path, converters={'cst_cod': str})

vendas = vendas.drop(columns=['sls_tme'])

vendas['sls_day'] = pd.to_datetime(vendas['sls_day'],
format='%Y%m%d')

# Agrupamento por data e soma do valor de vendas

vendas_valor = vendas.groupby('sls_day')['sls_vle'].sum()
```

5.3 Escolha dos Modelos

Neste trabalho optamos por testar dois tipos de modelos para a previsão de vendas: ARIMA (AutoRegressive Integrated Moving Average) e um modelo de Machine Learning baseado na biblioteca scikit-learn. A escolha desses modelos se deu pela capacidade do ARIMA em lidar com séries temporais e pela flexibilidade do modelo de Machine Learning para capturar padrões complexos nos dados.

5.4 Implementação dos Modelos

5.4.1 Modelo ARIMA (AutoRegressive Integrated Moving Average)

Os modelos ARIMA são amplamente utilizados para casos e estudos de preços. Suas construções se baseiam em modelos estatísticos lineares, em que as séries temporais envolvidas na análise são obtidas por um processo determinístico ou estocástico, cuja natureza poderá ser representada por meio de um modelo matemático. (Cas, C. G., 2018).

A combinação dos modelos auto-regressivos (AR) e de médias móveis (MA) resulta no modelo denominado ARMA. Assim, tratando-se de uma série temporal Y_t e considerando-se suas primeiras diferenças $Y_t = Y_t - Y_{t-1}$, com p defasagens para a variável e q para os erros aleatórios, pode-se expressar o modelo ARMA(p, q) da seguinte maneira: (FELIPE, I. J. S, 2018).

$$Y_t = \mu + \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \dots + \phi_p Y_{t-p} + \varepsilon_t - \theta_1 \varepsilon_{t-1} - \theta_2 \varepsilon_{t-2} - \dots - \theta_q \varepsilon_{t-q}$$

O modelo representado por normalmente é abreviado por ARI (p, d), ou seja, autorregressivo de ordem p e integrado de ordem d . Quando uma série temporal apresenta uma tendência provavelmente ela poderá ser representada por um modelo integrado ARI, ARIMA ou IMA. (Cas, C. G. 2012)

O modelo ARIMA foi implementado utilizando a biblioteca statsmodels. Realizamos o ajuste do modelo aos dados históricos e geramos previsões para os próximos meses. Primeiramente é necessário importar as funções dos módulos da biblioteca que serão utilizados na criação do modelo.

```
# Código Python para implementação do modelo ARIMA
```

```
from statsmodels.tsa.arima.model import ARIMA
```

```
# Criação do modelo ARIMA
```

```
modelo_valor = ARIMA(vendas_valor, order=(7, 2, 0))
```

```
resultado_valor = modelo_valor.fit()
```

```
futuro_valor = resultado_valor.forecast(steps=15)
```

Essa parte de código cria um modelo ARIMA com a ordem (7, 2, 0). No ARIMA, os parâmetros de ordem (p, d, q) controlam o comportamento do modelo.

- p: Ordem do componente autorregressivo (AR), que modela as relações de dependência entre uma observação atual e suas observações passadas.
- d: Ordem de diferenciação, que representa o número de vezes que os dados foram diferenciados para torná-los estacionários (ou seja, para remover tendências).
- q: Ordem do componente de média móvel (MA), que modela as relações de dependência entre uma observação atual e os erros residuais passados.

A previsão de 15 steps representa a quantidade de meses que o modelo irá criar.

5.4.2 Modelo de Machine Learning (Regressão Linear)

Para o modelo de Machine Learning, utilizamos a biblioteca scikit-learn. Implementamos um modelo de regressão linear como exemplo, mas outros algoritmos podem ser explorados.

As técnicas de regressão linear desempenham um papel crucial na análise preditiva, proporcionando insights valiosos ao modelar relações lineares entre variáveis dependentes e independentes. Ela é especialmente útil em cenários nos quais se espera uma relação linear entre as variáveis. Por exemplo, na previsão de vendas com base em gastos com publicidade, a regressão linear pode ser aplicada para estimar o impacto linear dos gastos na receita. Entretanto, é importante notar que a regressão linear é sensível a outliers e pode não ser a escolha ideal quando as relações são não-lineares.

O primeiro passo é importar as funções, iremos aqui tratar das bibliotecas necessárias:

```
# Código Python para implementação do modelo de Machine Learning
```

```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
```

Nesta etapa faremos a segregação dos conjuntos de treinamento e de teste utilizando a função “train_test_split”. O método que será adotado para determinar a distribuição percentual dos dados, com 80% destinado ao treinamento e 20% para teste.

```
# Divisão dos dados em conjunto de treinamento e teste
```

```
train, test = train_test_split(vendas_valor, test_size=0.2,
                               shuffle=False)
```

Em seguida, criaremos o objeto “LinearRegression”, definindo o modelo de regressão linear com seus parâmetros padrões. Concluindo empregamos o método “fit” no objeto, utilizando os dados de treinamento como parâmetro.

```
# Criação e treinamento do modelo de regressão linear

modelo_ml = LinearRegression()

modelo_ml.fit(X_train, y_train)
```

É necessário definir uma métrica de avaliação que será analisada para avaliar a eficácia do modelo. Com esse objetivo será empregada a função da biblioteca Scikit-learn: “metrics”.

```
# Avaliação do modelo

previsao_ml = modelo_ml.predict(X_test)

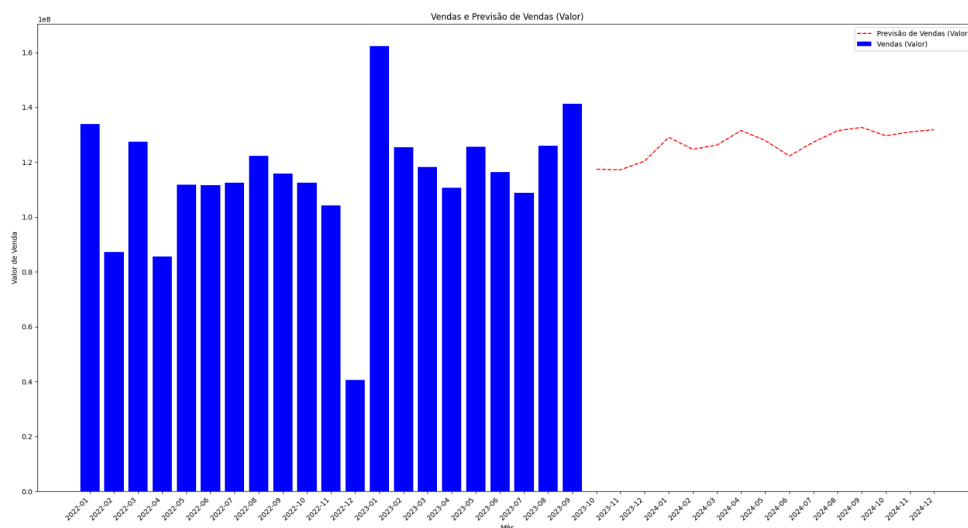
erro = mean_squared_error(y_test, previsao_ml)
```


5.5 Avaliação dos Modelos

O modelo ARIMA (AutoRegressive Integrated Moving Average) foi aplicado para realizar a previsão do valor de vendas ao longo do tempo. A análise teve início com a leitura dos dados de vendas, seguida pela remoção de colunas desnecessárias e a conversão da data para o formato adequado. Posteriormente, as vendas foram agrupadas por mês e ano, e o valor total foi calculado para cada período.

O ARIMA foi configurado com uma ordem de (7, 2, 0), indicando a presença de 7 termos autorregressivos, 2 diferenciações e nenhum termo de média móvel. A previsão foi realizada para 15 períodos futuros.

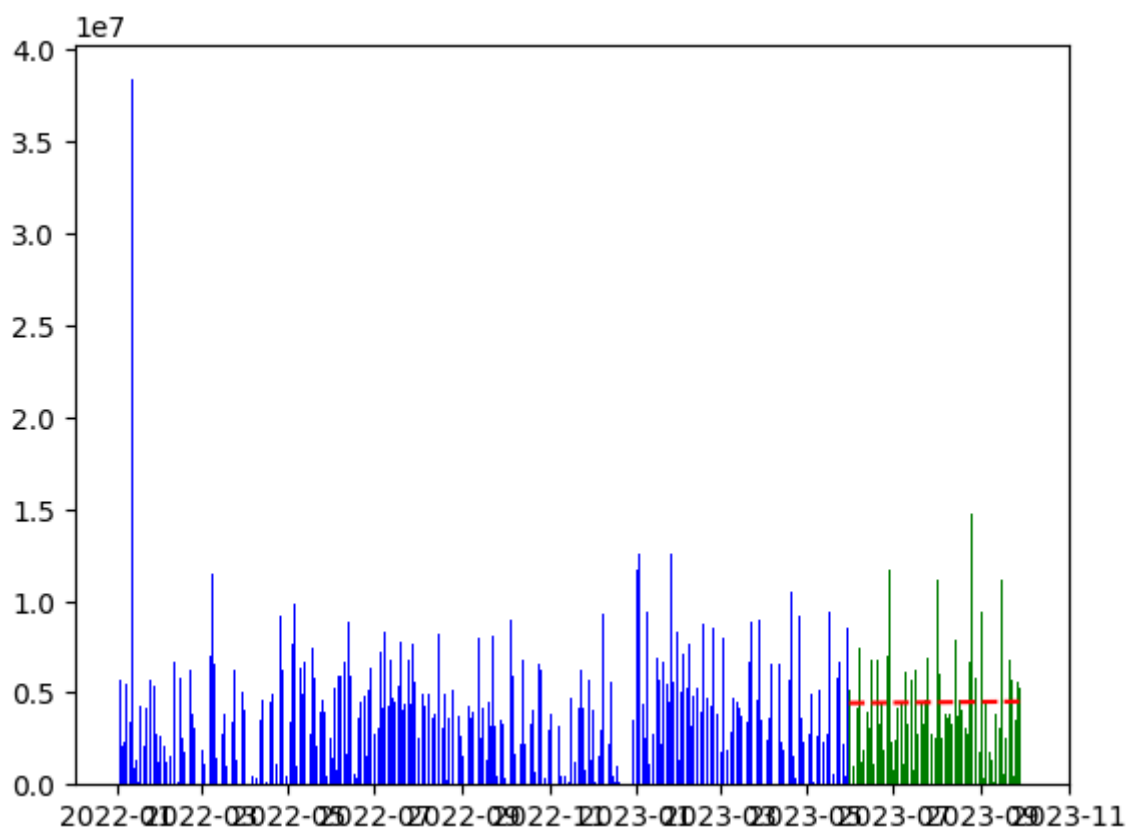
O gráfico gerado apresenta as vendas reais em barras azuis e a previsão em linha vermelha pontilhada. Observa-se que o modelo captura tendências e padrões sazonais nos dados, proporcionando uma representação visual do desempenho do ARIMA na previsão do valor de vendas.



O modelo de Machine Learning utilizado foi a Regressão Linear. Inicialmente, os dados foram preparados, removendo colunas desnecessárias e convertendo a data para o formato adequado. Em seguida, as vendas foram agrupadas por dia e o valor total foi calculado para cada dia.

O conjunto de dados foi dividido em conjuntos de treino e teste, sendo aplicada a Regressão Linear aos dados de treino. A previsão foi realizada para o conjunto de teste, e o erro médio quadrático foi calculado para avaliar a precisão do modelo.

O gráfico gerado apresenta as vendas de treino em barras azuis, as vendas de teste em barras verdes e a previsão em linha vermelha pontilhada. A análise visual demonstra a capacidade do modelo de Machine Learning em acompanhar a tendência geral dos dados.



Ambos os modelos apresentam vantagens e limitações. O ARIMA destaca-se na modelagem de séries temporais e na captura de padrões sazonais, enquanto a Regressão Linear é eficaz em capturar relações lineares nos dados.

A escolha entre esses modelos dependerá da natureza específica dos dados e dos objetivos da previsão. Recomenda-se a consideração de ambas as abordagens para obter uma visão abrangente do desempenho do modelo.

6. Interpretação dos Resultados

Na análise e exploração dos dados, identificamos diversos insights relevantes que fornecem uma compreensão profunda do comportamento das vendas e são fundamentais para a gestão de estoque. Alguns desses insights são:

1. **Padrões Sazonais:** Durante a análise exploratória, observamos padrões sazonais nas vendas, indicando variações regulares ao longo do ano. Esses padrões podem ser associados a eventos sazonais, como feriados, estações do ano ou promoções específicas.
2. **Tendências de Crescimento ou Decréscimo:** Identificamos possíveis tendências de crescimento ou decréscimo nas vendas ao longo do tempo. Essas tendências podem ser influenciadas por fatores macroeconômicos, mudanças no comportamento do consumidor ou outras variáveis externas.
3. **Correlações entre Diferentes Variáveis:** Exploramos correlações entre variáveis, como o impacto de promoções nas vendas ou a relação entre o dia da semana e o volume de vendas. Essas correlações podem ser valiosas para ajustar estratégias de marketing ou planejar ações promocionais.

Resultados da Aplicação dos Algoritmos de Machine Learning:

Ao aplicar o ARIMA e o modelo de Regressão Linear, obtivemos resultados significativos para a previsão do valor de vendas. A interpretação desses resultados inclui:

1. **ARIMA:** O modelo ARIMA demonstrou eficácia na captura de padrões sazonais nas séries temporais de vendas. Sua capacidade de diferenciação e auto regressão permitiu a modelagem de tendências complexas. A interpretação dos coeficientes e termos autorregressivos fornece insights sobre como o modelo está respondendo a padrões específicos.
2. **Regressão Linear:** O modelo de Regressão Linear apresentou uma abordagem mais linear na previsão do valor de vendas. A interpretação dos coeficientes destaca a relação linear entre a variável independente (dias desde o início) e a variável dependente (valor de vendas). O erro médio quadrático fornece uma métrica quantitativa da precisão do modelo.

Insights Importantes:

1. **Sazonalidade para Planejamento de Estoque:** Os padrões sazonais identificados são essenciais para o planejamento de estoque. A capacidade de antecipar aumentos na demanda durante determinados períodos permite uma gestão mais eficiente dos níveis de estoque.

2. **Análise de Tendências para Estratégias de Marketing:** A identificação de tendências de crescimento ou decréscimo nas vendas fornece insights para ajustar estratégias de marketing. A capacidade de prever mudanças nas preferências do consumidor é valiosa para ações promocionais direcionadas.
3. **Avaliação Contínua dos Modelos:** A interpretação dos resultados dos modelos de Machine Learning destaca a importância da avaliação contínua. A adaptação dos modelos conforme necessário, monitorando a precisão e ajustando parâmetros, é crucial para garantir previsões confiáveis.

Em resumo, a análise e exploração dos dados, juntamente com a aplicação de modelos de Machine Learning, proporcionam insights valiosos para a gestão de estoque, promovendo decisões informadas e estratégias eficazes.

7. Apresentação dos Resultados

Nessa seção você deve apresentar os resultados obtidos. Apresente gráficos, *dashboards*, conte a sua história de forma bastante criativa. Aqui você pode utilizar os modelos de Canvas propostos por Dourard (clique [aqui](#)) ou por Vasandani (clique [aqui](#)).

Data Science Workflow Canvas*
Start here. The sections below are ordered intentionally to make you state your goals first, followed by steps to achieve those goals. You're allowed to switch orders of these steps!

Title: Previsão de Vendas		
<div style="background-color: #f8d7da; padding: 5px; margin-bottom: 10px;"> 1 Problem Statement <small>What problem are you trying to solve? What larger issues do the problem address?</small> </div> <p>Com esse projeto, estamos tentando otimizar a gestão de estoque em uma empresa baseada nas vendas futuras de produtos, minimizando as reclamações de clientes e o desperdício de espaço físico.</p>	<div style="background-color: #f8d7da; padding: 5px; margin-bottom: 10px;"> 2 Outcomes/Predictions <small>What prediction(s) are you trying to make? Identify applicable predictor (X) and/or target (y) variables.</small> </div> <p>Vendas de produtos em termo de valor ao longo do tempo. O objetivo é antecipar as tendências nas vendas e fornecer estimativas.</p>	<div style="background-color: #f8d7da; padding: 5px; margin-bottom: 10px;"> 3 Data Acquisition <small>Where are you sourcing your data from? Is there enough data? Can you work with it?</small> </div> <p>Os dados foram obtidos de um banco de dados que contém informações de vendas, informações de produtos, informações de clientes e de estoque. Os dados coletados foram o suficiente para executarmos o projeto.</p>
<div style="background-color: #f8d7da; padding: 5px; margin-bottom: 10px;"> 4 Modeling <small>What models are appropriate to use given your outcomes?</small> </div> <p>Foram utilizados o modelos ARIMA e Regressão Linear</p>	<div style="background-color: #f8d7da; padding: 5px; margin-bottom: 10px;"> 5 Model Evaluation <small>How can you evaluate your model's performance?</small> </div> <p>O modelo ARIMA foi útil para lidar com tendências sazonais enquanto a Regressão Linear foi eficaz para capturar relações lineares nos dados</p>	<div style="background-color: #f8d7da; padding: 5px; margin-bottom: 10px;"> 6 Data Preparation <small>What do you need to do to your data in order to run your model and achieve your outcomes?</small> </div> <p>Foi necessário extrair todas as informações necessárias como por exemplo, valor de cada pedido, material, cliente e data. Após esses dados extraídos foi necessário alguns tratamentos, como por exemplo no formato da data e no código do cliente</p>

✓

Activation

When you finish filling out the canvas above, now you can begin implementing your data science workflow in roughly this order.

1 Problem Statement →
 2 Data Acquisition →
 3 Data Prep →
 4 Modeling →
 5 Outcomes/Preds →
 6 Model Eval

* Note: This canvas is intended to be used as a starting point for your data science projects. Data science workflows are typically nonlinear.

Conceptualized by Jasmine Vasandani using notes from General Assembly's Data Science Immersive. Format inspired by Business Model Canvas.

8. Links

Link para o vídeo: <https://youtu.be/4bmhGA1thk8>

Link para o repositório: <https://github.com/MuBonomo/ProvisionamentoVendas>

REFERÊNCIAS

Cas, C. G. (2018). **Aplicação do Modelo ARIMA para Previsão do Preço da Commodity Milho. Revista Gestão Da Produção, Operações E Sistemas.** 13(1), 263. <https://doi.org/10.15675/gepros.v13i1.2040>

FELIPE, I. J. S. **Aplicação de modelos Arima em séries de preços de soja no Norte do Paraná.** Tekhne e Logos, v. 3, n. 3, 2012.

LISBOA, Henrique Borsatti. **Análise do comércio internacional e previsão dos modais de transporte internacional de mercadorias.** São Paulo: Pontifícia Universidade Católica de Minas Gerais Núcleo de Educação a Distância, 2021.

APÊNDICE

Programação/Scripts

1 - Código para agrupar os arquivos CSVs, já que as vendas estavam separadas por mês:

```
import os
import pandas as pd

diretorio = "D:/Projetos/TCC/Arquivos Base/Vendas/"
dataframes = []

for nome_arquivo in os.listdir(diretorio):
    if nome_arquivo.endswith(".csv"):
        caminho_arquivo = os.path.join(diretorio, nome_arquivo)
        df = pd.read_csv(caminho_arquivo, converters = {"HR":str,
"REMESSA":str, "CLIENTE":str, "MATERIAL":str})
        dataframes.append(df)

vendas_total = pd.concat(dataframes, ignore_index=True)

caminho_saida = os.path.join(diretorio, "Vendas_Total.csv")
vendas_total.to_csv(caminho_saida, index=False)

print("Arquivo 'Vendas_Total.csv' criado com sucesso!")
```


2 - Código para inserir as informações em um banco de dados SQLITE3 caso houvesse a necessidade de fazer a conexão via SQL:

```
import sqlite3
import os
import pandas as pd

nome_banco_dados = "D:/Projetos/TCC/DB/DB_TCC.db"

nome_tabela = "sales"

caminho_csv = "D:/Projetos/TCC/Arquivos Base/Vendas/Vendas_Total.csv"

dados = pd.read_csv(caminho_csv, converters={"sls_dlv": str, "cod_mat":
str, "sls_tme": str, "cst_cod": str})
dados["sls_day"] = pd.to_datetime(dados["sls_day"], format="%Y%m%d")

conexao = sqlite3.connect(nome_banco_dados)

dados.to_sql(nome_tabela, conexao, if_exists="replace", index=False)

conexao.close()

print("Dados inseridos com sucesso no banco de dados SQLite.")
```

3 - Código para aplicação da ARIMA:

```
import pandas as pd
import matplotlib.pyplot as plt
from statsmodels.tsa.arima.model import ARIMA

file_path = "D:/Projetos/TCC/Arquivos Base/Vendas/Vendas_Total.csv"

vendas = pd.read_csv(file_path, converters={'cst_cod': str})
vendas = vendas.drop(columns=['sls_tme'])
vendas['sls_day'] = pd.to_datetime(vendas['sls_day'], format='%Y%m%d')

vendas['mes_ano'] = vendas['sls_day'].dt.to_period('M')
vendas_valor = vendas.groupby('mes_ano')['sls_vle'].sum()
vendas_valor = vendas_valor.sort_index()

modelo_valor = ARIMA(vendas_valor, order=(7, 2, 0))
resultado_valor = modelo_valor.fit()
futuro_valor = resultado_valor.forecast(steps=15)

plt.figure(figsize=(24, 12))

plt.bar(vendas_valor.index.astype(str), vendas_valor, label='Vendas
(Valor)', color='blue')
plt.plot(futuro_valor.index.astype(str), futuro_valor, label='Previsão de
Vendas (Valor)', color='red', linestyle='dashed')
plt.title("Vendas e Previsão de Vendas (Valor)")
plt.xlabel("Mês")
plt.ylabel("Valor de Venda")
plt.xticks(rotation=45, ha='right')
plt.legend()

plt.savefig('D:/Projetos/TCC/Arquivos Base/Vendas/ARIMA.png')
```

4 - Código para aplicação da Machine Learning:

```

import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
import numpy as np

file_path = "D:/Projetos/TCC/Arquivos Base/Vendas/Vendas_Total.csv"

vendas = pd.read_csv(file_path, converters={'cst_cod': str})
vendas = vendas.drop(columns=['sls_tme'])
vendas['sls_day'] = pd.to_datetime(vendas['sls_day'], format='%Y%m%d')

vendas_valor = vendas.groupby('sls_day')['sls_vle'].sum()

vendas_valor = vendas_valor.reset_index()
vendas_valor['dias_desde_inicio'] = (vendas_valor['sls_day'] -
vendas_valor['sls_day'].min()).dt.days

train, test = train_test_split(vendas_valor, test_size=0.2,
shuffle=False)

modelo_linear = LinearRegression()
modelo_linear.fit(train[['dias_desde_inicio']], train['sls_vle'])

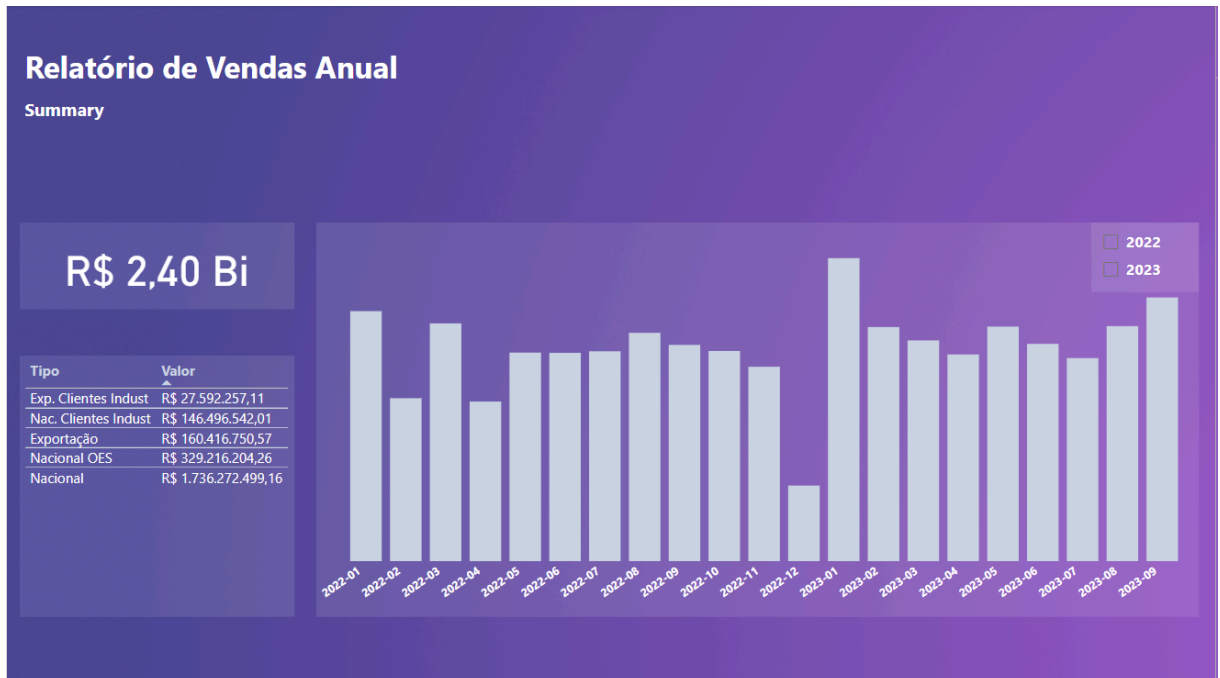
test['previsao_valor'] =
modelo_linear.predict(test[['dias_desde_inicio']])

plt.bar(train['sls_day'], train['sls_vle'], label='Treino (Valor)',
color='blue')
plt.bar(test['sls_day'], test['sls_vle'], label='Teste (Valor)',
color='green')
plt.plot(test['sls_day'], test['previsao_valor'], label='Previsão de
Vendas (Valor)', color='red', linestyle='dashed')
plt.title("Vendas e Previsão de Vendas (Valor)")
plt.xlabel("Data")
plt.ylabel("Valor de Venda")
plt.legend()

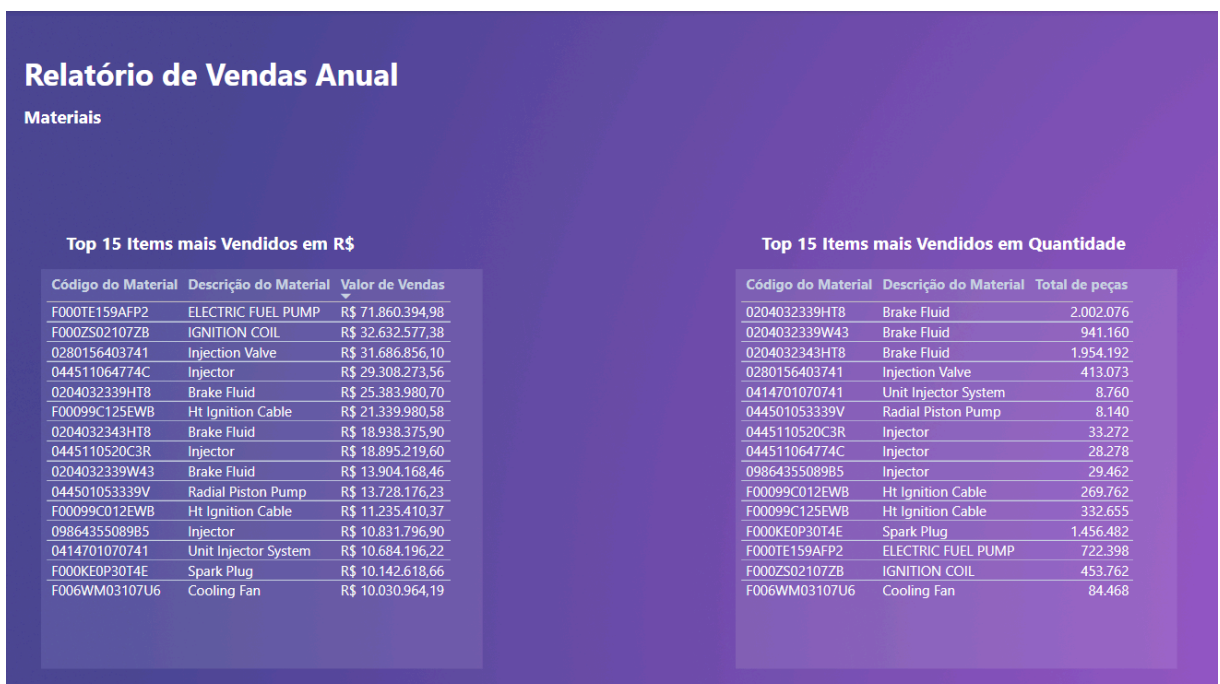
```

```
erro = mean_squared_error(test['sls_vle'], test['previsao_valor'])  
print(f"Erro Médio Quadrático: {erro}")  
  
plt.show()
```

1 - Relatório Anual de Vendas:



2 - Relatório de Itens mais vendidos por Quantidade e por Valor:



3 - Relatório de Clientes que mais emitem pedidos por Quantidade e por Valor:

Relatório de Vendas Anual

Materiais

Top 15 Clientes em R\$

Código	Valor de Vendas
924900	R\$ 82.697.161,42
1774300	R\$ 39.071.141,93
16610000	R\$ 35.760.434,20
268500	R\$ 33.541.519,58
493900	R\$ 32.073.142,33
58017	R\$ 29.522.717,10
18196504	R\$ 24.982.982,67
1564301	R\$ 23.785.325,21
3374310	R\$ 23.709.888,41
8899900	R\$ 23.525.492,46
58007	R\$ 23.344.167,05
924934	R\$ 22.770.374,17
8580008	R\$ 22.504.419,22
16325200	R\$ 22.482.340,92
683802	R\$ 20.928.797,45

Top 15 Clientes por Pedido

Código	Contagem de sls. div
1283502	1693
16235200	1221
16325200	1503
1774300	1255
3097900	1238
3374310	1330
466201	1170
58001	1225
58004	1290
58009	1222
58013	1199
58017	1369
686303	2434
866302	1148
924900	3693
924934	1264