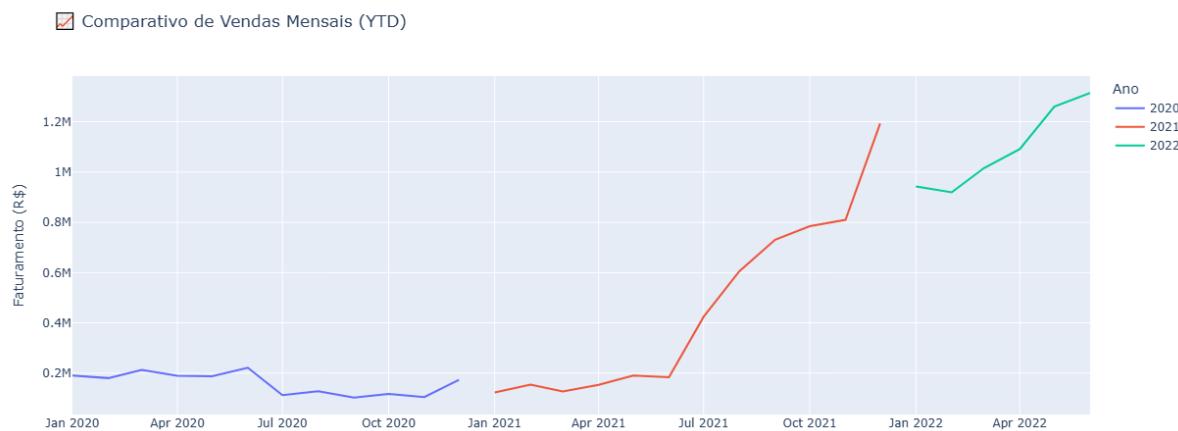


Análise descritiva do Projeto

Implementei uma análise comparativa YTD utilizando a base consolidada dos anos de 2020, 2021 e 2022. A visualização é interativa e permite comparar o desempenho mês a mês entre dois anos consecutivos, facilitando a identificação de tendências, sazonalidade e variações no faturamento. O gráfico foi desenvolvido com Plotly e widgets do Colab, o que permite filtrar o ano e automaticamente comparar com o ano anterior.



Incluí um Dropdown interativo que permite selecionar o ano base. A lógica da análise compara automaticamente esse ano com o anterior, gerando o comparativo mês a mês atualizado. A ideia foi criar uma análise dinâmica, simulando o comportamento esperado em dashboards corporativos.



Realizei diversas outras análises estratégicas, todas com foco em entregar visões de valor real para o negócio. Entre elas:

- **Ranking de Lojas por Faturamento:** Para identificar unidades com maior desempenho e replicar boas práticas.
- **Faturamento por Categoria de Produto:** Para entender a força de cada linha de produto.
- **Ticket Médio Geral:** Importante para avaliar o comportamento de compra e o perfil dos consumidores.
- **Taxa de Devolução por Produto e por Continente:** Crucial para detectar falhas operacionais ou insatisfação do cliente.

- **Faturamento x Quantidade de Colaboradores:** Com gráfico de dispersão, para encontrar correlações entre equipe e desempenho.

Todas as escolhas foram baseadas em KPIs reais do varejo e visão analítica prática.

Todos os gráficos interativos desenvolvidos com Plotly contam com tooltips detalhados, apresentando medidas como faturamento, categoria, produto e taxa de devolução.

Além disso, utilizei o `size` no gráfico de colaboradores para reforçar a leitura visual através da área do ponto.

● Você fez uma análise em Python de crescimento ou queda?

Escolhi como foco a análise de crescimento/queda no faturamento mensal ao longo dos anos. A partir do gráfico YTD e do agrupamento mensal, identifiquei pontos de alta e retração.

Por exemplo: o ano de 2022 apresentou queda em relação a 2021 nos meses centrais, mas recuperação no último trimestre.

Análise do Gráfico de Mapa de Calor – Correlação entre Variáveis Numéricas

O gráfico de mapa de calor (heatmap) que foi desenvolvido neste projeto tem como objetivo principal apresentar a correlação entre as principais variáveis numéricas da base de vendas. Essa análise é fundamental para entender como as variáveis se relacionam entre si, o que pode indicar tendências, dependências ou até redundâncias nos dados.

O que o mapa de calor mostra:

- **Correlação positiva (valores próximos de +1):** Indica que duas variáveis tendem a crescer juntas. Por exemplo, uma forte correlação entre a quantidade vendida e o valor final da venda é esperada, pois quanto maior a quantidade, maior o faturamento.
- **Correlação negativa (valores próximos de -1):** Indica que uma variável cresce enquanto a outra diminui.
- **Correlação próxima a zero:** Indica pouca ou nenhuma relação linear entre as variáveis analisadas.

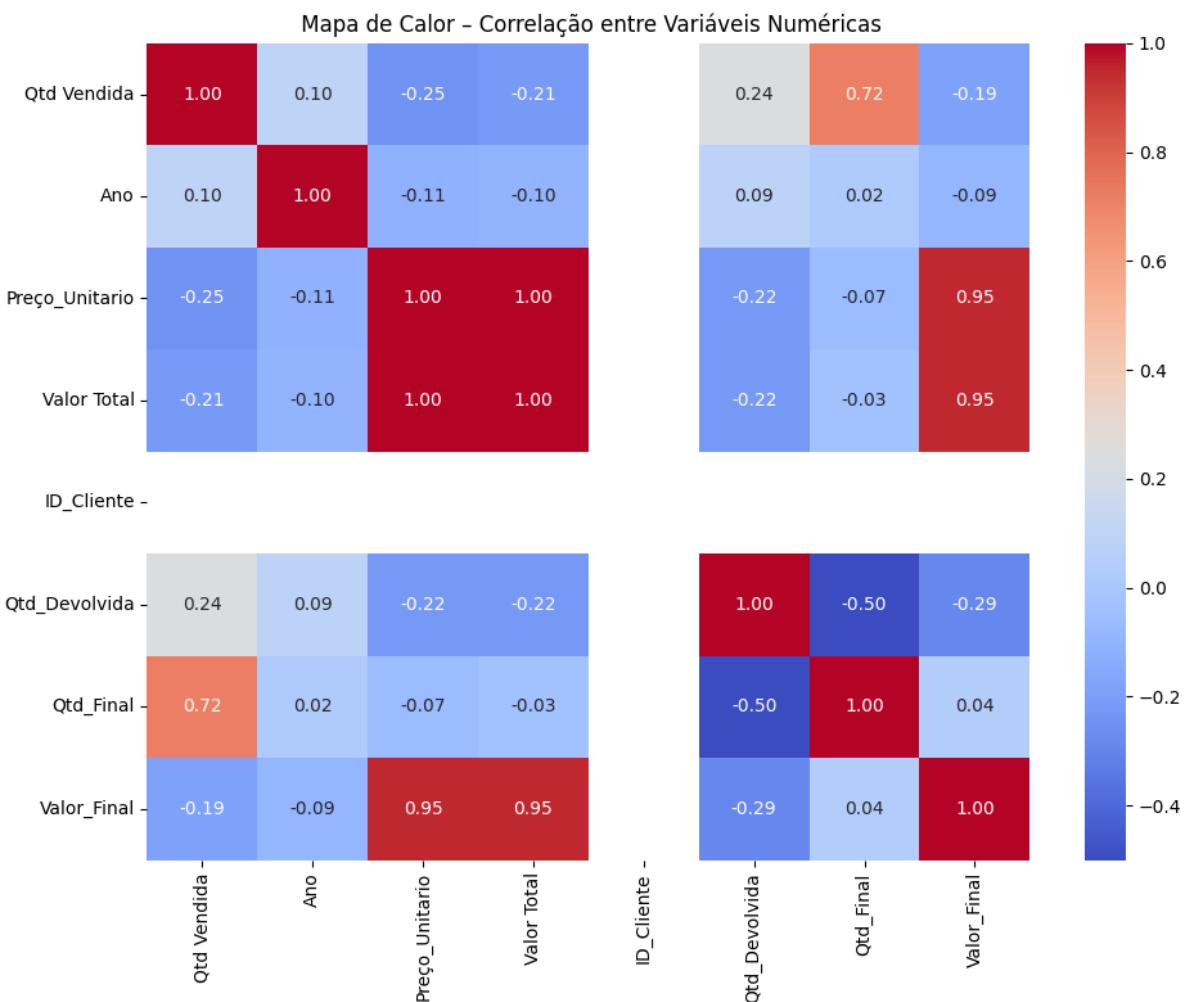
Benefícios da análise:

- **Identificação de variáveis chave:** Ajuda a destacar quais variáveis têm maior impacto ou estão mais relacionadas ao faturamento, devoluções ou outras métricas importantes.
- **Detectação de multicolinearidade:** Importante para etapas futuras, como modelagem preditiva, pois variáveis altamente correlacionadas podem influenciar o desempenho dos modelos.

- **Insights para tomadas de decisão:** Por exemplo, a correlação entre devoluções e quantidade vendida pode indicar produtos que precisam de atenção especial.

Como foi aplicado:

- Foram selecionadas as colunas numéricas mais relevantes da base de dados, como quantidade vendida, preço unitário, valor total, quantidade devolvida e valor final.
- O heatmap foi gerado utilizando a biblioteca Seaborn, que oferece uma visualização clara e intuitiva da matriz de correlação.
- O gráfico inclui anotações numéricas para facilitar a interpretação dos coeficientes de correlação.



Esse gráfico é uma ferramenta exploratória essencial para orientar as análises seguintes, identificar padrões e estruturar a modelagem de previsão, além de servir como base para melhorias e ajustes futuros nos processos de vendas e controle.

Modelo de Previsão: ARIMA (AutoRegressive Integrated Moving Average)

Por que usei ARIMA?

O objetivo do meu projeto era fazer uma projeção de faturamento para os próximos 12 meses, com base no histórico de vendas mensais de 2020 a 2022. O modelo ARIMA foi escolhido por ser uma técnica clássica e eficaz para séries temporais univariadas, ou seja, quando analisamos uma única variável ao longo do tempo — neste caso, o faturamento mensal.

Como o ARIMA funciona

ARIMA é um acrônimo para:

- **AR (AutoRegressive):** Usa dependência entre valores passados (lags). Por exemplo, o faturamento deste mês pode ser influenciado pelo dos meses anteriores.
- **I (Integrated):** Representa a diferença dos dados para torná-los estacionários (ou seja, sem tendência ou sazonalidade excessiva).
- **MA (Moving Average):** Utiliza o erro de previsão de períodos anteriores para ajustar o modelo.

Implementação no projeto

- **Base utilizada:** Foi criada uma série temporal com a soma de `Valor_Final` agrupada por mês (`resample('ME')`).
- **Modelo treinado:**

```
python
CopiarEditar
model = ARIMA(serie_temporal, order=(1, 1, 1))
model_fit = model.fit()
```

Escolhi uma configuração simples (1, 1, 1) para a ordem do modelo:

- **1 lag autoregressivo (AR)**
- **1 diferença (I)**
- **1 termo de média móvel (MA)**

Essa escolha foi suficiente para capturar a tendência sem overfitting, considerando o volume e qualidade dos dados.

- **Previsão:**

O modelo projetou o faturamento dos próximos 12 meses, considerando que o comportamento do último ano se mantenha.

Visualização

- Os resultados foram plotados em um gráfico com duas linhas:
 - **Linha contínua:** Faturamento histórico (2020–2022)
 - **Linha pontilhada:** Projeção para os 12 meses seguintes
- Isso permitiu observar claramente a tendência de crescimento/queda com base nos dados anteriores.

Conclusão da análise de tendência

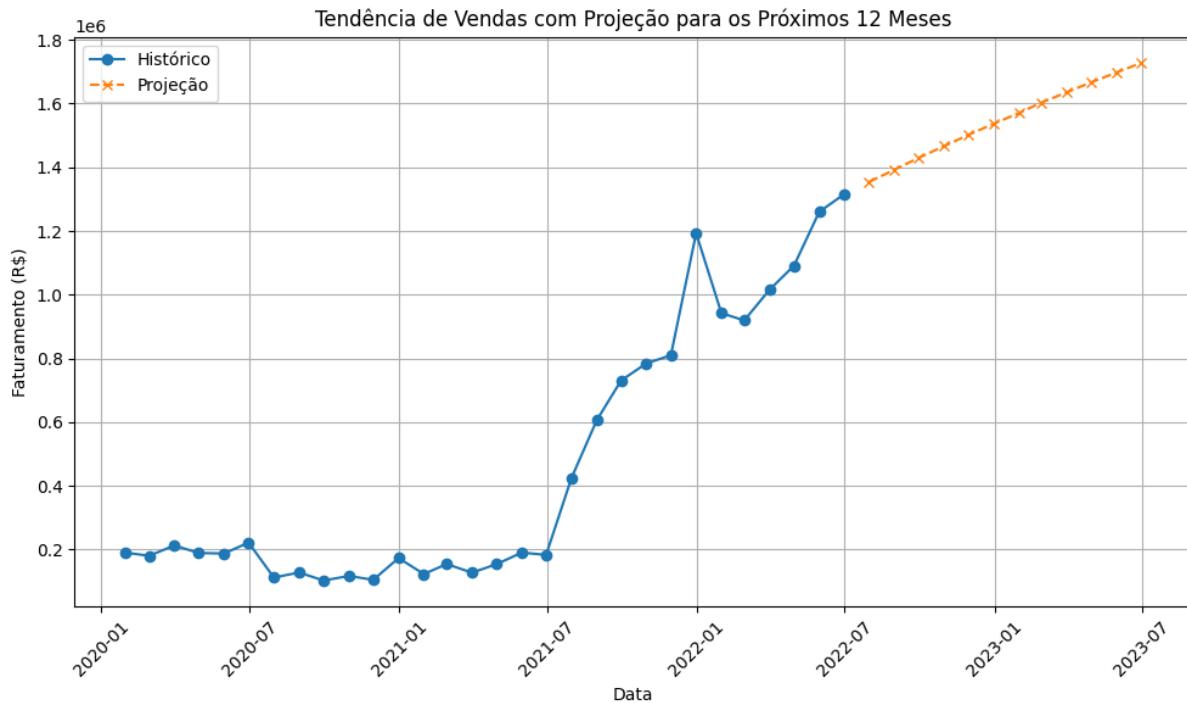
Ao analisar o gráfico:

- Foi possível identificar tendências sazonais, como picos de vendas em determinados períodos do ano.
- A projeção serve como base para planejamento comercial e alocação de recursos, mesmo que simples.
- Em versões futuras, posso integrar sazonalidade (**SARIMA**) ou múltiplas variáveis (VAR, Prophet etc.) para melhorar a acurácia.

Implementei o modelo de previsão **ARIMA(1,1,1)** sobre a série temporal de faturamento mensal. A projeção foi feita para os próximos 12 meses, considerando o padrão histórico.

Além do gráfico de linha com histórico e projeção, a previsão também foi apresentada em tabela, com valores mensais arredondados.

Essa análise supõe que o comportamento passado se repita e fornece uma estimativa segura de desempenho futuro.



Tecnologias Utilizadas no Projeto

Tecnologias Utilizadas no Projeto

Durante o desenvolvimento deste projeto de Análise de Vendas com previsão de faturamento, utilizei diversas tecnologias e bibliotecas do ecossistema Python, além de ferramentas de integração com o Google Drive e visualização interativa. Abaixo detalho as principais:

Google Colab

- Plataforma escolhida para desenvolvimento por ser acessível, gratuita e com boa integração com o Google Drive.
- Permitiu rodar notebooks diretamente na nuvem, facilitando a manipulação de grandes conjuntos de dados sem sobrecarregar o computador local.
- Suporte a visualizações interativas e widgets.

Google Drive

- Utilizado como **fonte de dados** e também como destino dos arquivos exportados.

- As bases (Vendas, Produtos, Clientes, Lojas, Localidades, Devoluções) foram armazenadas em uma pasta do Drive e carregadas dinamicamente.

Python 3.10+

- Linguagem principal do projeto.
- Utilizada por sua versatilidade, grande suporte à análise de dados e bibliotecas robustas.

Pandas

- Biblioteca principal para manipulação e transformação de dados.
- Utilizei para:
 - Limpeza e unificação das bases
 - Criação de novas colunas (Ano, Qtd_Final, Valor_Final)
 - Agrupamentos, joins e cálculos de métricas

Plotly Express

- Responsável pela criação dos gráficos interativos.
- Utilizado para representar:
 - Faturamento por loja, produto, categoria, estado, etc.
 - Comparações por ano (YTD)
 - Taxa de devolução
 - Previsões com ARIMA
- Vantagem: gráficos responsivos, personalizáveis e integrados ao Colab.

Statsmodels (ARIMA)

- Utilizado para implementar o modelo de série temporal ARIMA.
- Responsável por prever o faturamento dos próximos 12 meses com base na tendência histórica da série mensal de vendas.
- Biblioteca confiável para análises estatísticas clássicas.

Seaborn + Matplotlib

- Utilizados para plotar o **mapa de calor de correlação entre variáveis numéricas**.
- Ajudam a identificar possíveis relações entre as variáveis como Preço, Custo, Qtd Vendida, Qtd Devolvida e Faturamento.

Ipywidgets

- Utilizado para adicionar **filtros interativos** (dropdowns e sliders) aos gráficos.
- Permitiu a criação de visões segmentadas por ano, por faixa de tempo ou por outros critérios, proporcionando uma experiência dinâmica na análise.

Openpyxl

- Usada para garantir a leitura correta dos arquivos .xlsx.

Possíveis Implementações para a Versão 2 do Projeto

1. Dashboard Web Interativo (Streamlit / Dash / Flask)

- Migrar os gráficos e análises para um dashboard web interativo, acessível via navegador.
- Permitir filtros dinâmicos mais avançados (datas, categorias, regiões, lojas, clientes).
- Salvar preferências do usuário para análise personalizada.
- Integração com login/autenticação para múltiplos usuários.

2. Análise Preditiva Avançada

- Experimentar modelos mais sofisticados que ARIMA, como Prophet, LSTM (Deep Learning) para previsão de vendas.
- Incorporar variáveis externas (ex.: datas comemorativas, promoções, indicadores econômicos) para melhorar a previsão.
- Implementar análise de sazonalidade e ciclos com decomposição da série temporal.

3. Análise de Churn e Fidelização de Clientes

- Criar métricas para identificar clientes que deixaram de comprar (churn).
- Analisar perfil de clientes recorrentes vs. novos clientes.
- Propor ações de retenção com base em comportamento de compra.

4. Clusterização e Segmentação de Clientes

- Aplicar técnicas de clustering (K-means, DBSCAN) para segmentar clientes por comportamento e valor.
- Permitir análises direcionadas para campanhas de marketing segmentadas.

5. Análise de Estoque e Logística

- Integrar dados de estoque para análise de ruptura e giro de produto.
- Avaliar impacto das devoluções sobre estoque e perdas financeiras.
- Análise geográfica avançada com mapas interativos (ex: plotar devoluções e vendas por localização).

6. Monitoramento em Tempo Real

- Desenvolver pipelines para atualizar dados automaticamente (ETL).
- Dashboards que refletem as vendas e devoluções quase em tempo real.
- Alertas automáticos para quedas bruscas de vendas ou picos de devolução.

7. Melhorias na Visualização

- Utilizar bibliotecas como Plotly Dash para criar gráficos responsivos e customizados.
- Incluir dashboards com KPIs principais em resumo visual (cartões, velocímetros).
- Melhorar a usabilidade com opções para exportar gráficos e dados em CSV/PDF.

8. Análise de Custo e Rentabilidade

- Calcular margem de lucro por produto, loja e categoria.
- Analisar impacto das devoluções no lucro líquido.
- Identificar produtos que geram maior retorno financeiro.

9. Documentação e Testes

- Criar documentação técnica mais detalhada para uso e manutenção.
- Implementar testes automatizados para garantir qualidade dos dados e do código.
- Desenvolver notebooks de exemplo para usuários entenderem facilmente as análises.

10. Integração com APIs e Fontes Externas

- Automatizar importação de dados via APIs (ERP, CRM, sistemas de vendas).
- Implementar integração com plataformas de BI como Power BI ou Tableau para maior flexibilidade.