

Nome: Lucas dos Santos Siqueira.

2º Semestre de Análise e Desenvolvimento de Sistemas.

Análise de dados utilizando Python e IA



Prompt da Imagem

“The official Python logo in the center, glowing in blue and yellow, surrounded by futuristic technology elements: holograms, neon circuits, digital code streams, floating data panels, and a high-tech atmosphere. The background is sleek and modern, with a cybernetic aura that emphasizes innovation, programming, and artificial intelligence. The logo should be clear and central, while the surrounding effects highlight its importance in technology”.

Introdução

Linguagem de programação Python:

Considerada uma das linguagens de programação mais utilizadas pelos os desenvolvedores (TIOBE) , o Python se destaca na área de análise de dados de forma congruente por diversos fatores: sua versatilidade, sua acessibilidade (por ser uma linguagem considerada simples porém que supre abrangentemente as necessidades) e também possui uma biblioteca vasta de para análise de dados pronta para uso.

As principais ferramentas são:

- **Pandas:** Pandas é uma biblioteca do Python usada para análise e manipulação de dados. Ela oferece duas estruturas básicas e principais: Séries (vetor com índice) e DataFrame (tabelas de dados, como no Excel). Com a ferramenta, é possível ler dados de arquivos (CSV, Excel, SQL etc.), limpar, transformar, agrupar e visualizar informações com facilidade. É uma ferramenta excelente para quem trabalha com dados, por conta da sua simplicidade e eficiência. Ideal para análises, relatórios e preparação de dados para machine learning.
- **Matplotlib:** Matplotlib é uma biblioteca do Python usada para criação de gráficos e visualizações de dados. Ela permite gerar gráficos como linhas, barras, histogramas, dispersão (scatter), pizza, entre outros. Utilizando o Matplotlib, é possível transformar dados numéricos em visualizações claras e informativas, sendo muito usada em relatórios, análises exploratórias e apresentações. É uma ferramenta essencial para quem deseja visualizar dados de forma profissional no Python.
- **NumPy:** O Numpy é uma biblioteca fundamental do Python para cálculos numéricos e científicos. Ela oferece suporte a arrays e matrizes multidimensionais, além de funções matemáticas de alto desempenho. Com o NumPy, é possível fazer operações vetoriais, álgebra linear, transformações e estatísticas de forma rápida e eficiente. Sendo assim, é essencial para quem trabalha com ciência de dados, machine learning ou engenharia.

Todas as ferramentas acima são muito utilizadas de forma simultânea por analistas, já que quando são combinadas, garantem eficiência e qualidade em como os dados serão expostos.

Planejamento da Análise

Como base de dados, serão utilizadas planilhas de dados extraídas diretamente de um sistema de autoridade própria, chamado HubCheff, que foi desenvolvido inicialmente como um sistema de gestão voltada para hamburguerias, com foco em controle de pedidos, estoque, insumos, e organização operacional. No entanto, ao longo do seu desenvolvimento e testes práticos, observou-se que a sua estrutura robusta e flexível permite que ele seja utilizado também em diversos outros segmentos do setor alimentício.

Decidiu-se também que para a resolução da atividade será utilizada a IA generativa ChatGPT devido à sua grande evolução e facilidade de acesso. Assim, os dados das tabelas de produtos e insumos providenciados pelo HubCheff serão direcionados para a IA.

A estrutura das tabelas que serão utilizadas para a atividade foi criada originalmente no software MySQL Workbench, e será o ideal para a realização da atividade, abaixo estão listadas as estruturas das tabelas “Produto” e “Insumo” utilizadas.

tbl_produto	
Campos	Descrição
id	Identificador unitário do produto
nome	Nome do produto
status	Se o produto está ativo ou inativo
categoria	Categoria do produto (ex: lanche, bebida etc.)
medida	Unidade de medida (ex: unidade, litro, kg)
preco_custo	Custo para produzir/comprar o produto
preco_venda	Preço de venda ao consumidor

tbl_insumo	
Campos	Descrição
id	Identificador unitário do produto
nome	Nome do insumo
categoria	Categoria do produto (ex: carne, embutidos, etc.)
medida	Unidade de medida (ex: unidade, litro, kg)
estoque	Quantidade atual em estoque
estq_minimo	Quantidade mínima recomendada para reposição
preco_custo	Custo por unidade do insumo
custo_kg/l	Custo por quilo ou litro (dependendo da unidade)

Para a tabela de produtos, o ChatGPT irá observar a tabela com a intenção de realizar as seguintes análises:

1. Análise na margem de lucro por produto.

O objetivo é verificar a rentabilidade dos produtos, sendo possível fazer as seguintes observações:

- $\text{Lucro unitário} = \text{preco_venda} - \text{preco_custo}$
- $\text{Margem de lucro (\%)} = (\text{lucro} / \text{preco_venda}) \times 100$
- Produto mais e menos lucrativo

2. Analisar produtos inativos.

O objetivo é entender a proporção e características dos produtos desativados sendo possível fazer as seguintes observações:

- Quantidade e percentual de produtos ativos e inativos.
- Categorias com mais produtos inativos.

3. Analisar categorias de produtos existentes.

O objetivo é explorar a distribuição dos produtos por categoria, sendo possível fazer as seguintes observações:

- Número de produtos por categoria.
- Média de lucro por categoria

Para a tabela de insumos, o ChatGPT irá realizar as seguintes análises:

1. Análise de estoque crítico.

O objetivo é identificar insumos com estoque abaixo do mínimo, sendo possível realizar as seguintes observações:

- Percentual de insumos em situação crítica.
- Categorias mais afetadas.
- Custo total dos insumos abaixo do mínimo.

2. Análise de custo por insumo.

O objetivo é verificar quais insumos mais impactam financeiramente o negócio, sendo possível realizar as seguintes observações:

- Custo total por insumo = estoque × custo_por_kg_l
- Top 5 insumos mais caros (em valor total de estoque)

Assim o objetivo é mostrar como a integração entre sistemas de gestão e ferramentas de inteligência artificial pode potencializar o uso de dados operacionais, trazendo benefícios concretos para negócios do setor alimentício.

Criação do Prompt para a IA

A formulação dos prompts segue uma estrutura clara para que a IA entenda o que precisa ser feito, dividida em Contexto, Objetivo, Dados de entrada, O que o código deve fazer e Ferramentas.

Prompt 1 – Análise de Produtos: foca em analisar produtos de uma hamburgueria a partir de uma planilha do HubCheff. O contexto informa que se trata de preços, categorias e status dos produtos. O objetivo é calcular margem de lucro, identificar produtos ativos/inativos e analisar distribuição por categoria. Os dados de entrada fornecem exemplos e formatos de cada campo. O que o código deve fazer inclui cálculos de lucro, estatísticas de status e categoria e geração de gráficos. As ferramentas indicadas são Pandas e Matplotlib. Essa formulação garante que a IA produza um código funcional, com análise financeira e visualizações.

Prompt 2 – Análise de Insumos: foca em insumos, avaliando estoque e custo. O contexto mostra que os dados são de ingredientes. O objetivo é identificar insumos críticos, categorias mais afetadas e custos totais. Os dados de entrada indicam os campos e formatos. O que o código deve fazer envolve cálculos de estoque crítico, percentuais, custos, ranking dos mais caros e gráficos. Pandas e Matplotlib são usados como ferramentas. Com essa estrutura, a IA consegue gerar

um código que calcula valores, organiza dados e produz visualizações de forma clara. Abaixo estão listados os prompts utilizados para a realização da atividade.

Prompt 1 – Análise de Produtos

Contexto

Sou um estudante e preciso de um código em Python para fazer uma análise de dados de produtos, utilizando os dados de uma planilha exportada do sistema HubCheff, que é usado para gerenciar hamburguerias e outros negócios do setor alimentício.

Objetivo

Quero realizar uma análise de margem de lucro dos produtos, além de identificar quais produtos estão ativos ou inativos e como eles estão distribuídos entre as categorias.

Dados de entrada

Abaixo segue a amostra dos dados da planilha com 10 produtos fictícios:

- **Produto 1:**
Nome: X-Burguer | Status: ativo | Categoria: Lanche | Medida: un | Preço de custo: 6.50 | Preço de venda: 14.00
- **Produto 2:**
Nome: X-Salada | Status: ativo | Categoria: Lanche | Medida: un | Preço de custo: 7.00 | Preço de venda: 15.50
- **Produto 3:**
Nome: X-Bacon | Status: inativo | Categoria: Lanche | Medida: un | Preço de custo: 8.00 | Preço de venda: 16.00
- **Produto 4:**
Nome: Coca-Cola 350ml | Status: ativo | Categoria: Bebida | Medida: lata | Preço de custo: 3.00 | Preço de venda: 6.50
- **Produto 5:**
Nome: Coca-Cola 600ml | Status: inativo | Categoria: Bebida | Medida: garrafa | Preço de custo: 4.00 | Preço de venda: 8.00
- **Produto 6:**
Nome: Batata Média | Status: ativo | Categoria: Acompanhamento | Medida: porção | Preço de custo: 4.00 | Preço de venda: 10.00

- **Produto 7:**

Nome: Batata Grande | Status: ativo | Categoria: Acompanhamento | Medida: porção | Preço de custo: 5.00 | Preço de venda: 12.00

- **Produto 8:**

Nome: X-Egg | Status: ativo | Categoria: Lanche | Medida: un | Preço de custo: 7.50 | Preço de venda: 15.00

- **Produto 9:**

Nome: X-Tudo | Status: ativo | Categoria: Lanche | Medida: un | Preço de custo: 9.00 | Preço de venda: 18.00

- **Produto 10:**

Nome: Guaraná Lata | Status: ativo | Categoria: Bebida | Medida: lata | Preço de custo: 2.80 | Preço de venda: 6.00

O que o código deve fazer

- Calcular o lucro unitário ($\text{preço_venda} - \text{preço_custo}$)
- Calcular a margem de lucro (%)
- Identificar o produto mais e menos lucrativo
- Contar quantos produtos estão ativos e inativos e calcular o percentual
- Contar quantos produtos existem por categoria
- Calcular a média de lucro por categoria
- Gerar um gráfico de barras com o lucro unitário de cada produto
- Gerar um gráfico de pizza com a proporção de produtos ativos e inativos

Ferramentas a serem usadas

Utilize as bibliotecas Pandas para análise de dados e Matplotlib para gerar os gráficos.

Prompt 2 – Análise de Insumos

Contexto:

Sou um estudante e preciso de um código em Python para analisar uma planilha de insumos que foi gerada pelo sistema HubCheff, utilizado na gestão de restaurantes e hamburguerias.

Objetivo:

Quero realizar uma análise para identificar quais insumos estão com estoque abaixo do mínimo recomendado e entender quais insumos representam o maior custo de estoque.

Dados de entrada:

Abaixo segue uma amostra com 10 registros fictícios da tabela de insumos:

Insumo 1:

Nome: Pão Brioche | Categoria: Padaria | Medida: un | Estoque: 40 | Estoque mínimo: 50 | Preço de custo: 0.60 | Custo por kg/l: 10.00

Insumo 2:

Nome: Carne Bovina 150g | Categoria: Carne | Medida: kg | Estoque: 25 | Estoque mínimo: 30 | Preço de custo: 28.00 | Custo por kg/l: 28.00

Insumo 3:

Nome: Bacon | Categoria: Embutidos | Medida: kg | Estoque: 10 | Estoque mínimo: 15 | Preço de custo: 22.00 | Custo por kg/l: 22.00

Insumo 4:

Nome: Cebola | Categoria: Hortifruti | Medida: kg | Estoque: 60 | Estoque mínimo: 40 | Preço de custo: 4.00 | Custo por kg/l: 4.00

Insumo 5:

Nome: Batata Congelada | Categoria: Congelados | Medida: kg | Estoque: 20 | Estoque mínimo: 25 | Preço de custo: 12.00 | Custo por kg/l: 12.00

Insumo 6:

Nome: Alface | Categoria: Hortifruti | Medida: kg | Estoque: 30 | Estoque mínimo: 20 | Preço de custo: 3.00 | Custo por kg/l: 3.00

Insumo 7:

Nome: Tomate | Categoria: Hortifruti | Medida: kg | Estoque: 18 | Estoque mínimo: 25 | Preço de custo: 5.00 | Custo por kg/l: 5.00

Insumo 8:

Nome: Queijo Cheddar | Categoria: Laticínios | Medida: kg | Estoque: 12 | Estoque mínimo: 15 | Preço de custo: 18.00 | Custo por kg/l: 18.00

Insumo 9:

Nome: Presunto | Categoria: Embutidos | Medida: kg | Estoque: 9 | Estoque mínimo: 10 | Preço de custo: 19.00 | Custo por kg/l: 19.00

Insumo 10:

Nome: Ovo | Categoria: Granja | Medida: un | Estoque: 60 | Estoque mínimo: 50 | Preço de custo: 0.50 | Custo por kg/l: 12.00

O que o código deve fazer:

- Identificar os insumos que estão com estoque abaixo do mínimo

- Calcular o percentual de insumos em situação crítica
- Agrupar por categoria e mostrar quais são as mais afetadas
- Custo total dos insumos abaixo do mínimo.
- Calcular o custo total em estoque por insumo (estoque × custo_kg/l)
- Identificar os 5 insumos mais caros em valor total de estoque
- Gerar um gráfico de barras com os valores totais de cada insumo
- Gerar um gráfico de barras apenas com os insumos em estoque crítico

Ferramentas a serem usadas:

Utilize as bibliotecas Pandas para análise de dados e Matplotlib para visualização gráfica.

Análise e Apresentação dos Resultados

Tabela Produtos

Após a execução do prompt para a análise da tabela dos produtos, a IA fez a seguinte resposta:

“ Beleza! Aqui está o código completo em Python que realiza a análise de produtos conforme seu prompt, usando pandas e matplotlib.

```
import pandas as pd
import matplotlib.pyplot as plt

# Dados dos produtos
dados = {
    'nome': ['X-Burguer', 'X-Salada', 'X-Bacon', 'Coca-Cola 350ml', 'Coca-Cola 600ml',
            'Batata Média', 'Batata Grande', 'X-Egg', 'X-Tudo', 'Guaraná Lata'],
    'status': ['ativo', 'ativo', 'inativo', 'ativo', 'inativo',
              'ativo', 'ativo', 'ativo', 'ativo', 'ativo'],
    'categoria': ['Lanche', 'Lanche', 'Lanche', 'Bebida', 'Bebida',
                  'Acompanhamento', 'Acompanhamento', 'Lanche', 'Lanche', 'Bebida'],
    'medida': ['un', 'un', 'un', 'lata', 'garrafa',
              'porção', 'porção', 'un', 'un', 'lata'],
    'preco_custo': [6.50, 7.00, 8.00, 3.00, 4.00, 4.00, 5.00, 7.50, 9.00, 2.80],
    'preco_venda': [14.00, 15.50, 16.00, 6.50, 8.00, 10.00, 12.00, 15.00, 18.00,
                    6.00]
}

# Criando DataFrame
df = pd.DataFrame(dados)
```

```

# Cálculos de lucro unitário e margem de lucro (%)
df['lucro_unitario'] = df['preco_venda'] - df['preco_custo']
df['margem_lucro_perc'] = (df['lucro_unitario'] / df['preco_venda']) * 100

# Produto mais e menos lucrativo
produto_mais_lucrativo = df.loc[df['lucro_unitario'].idxmax()]
produto_menos_lucrativo = df.loc[df['lucro_unitario'].idxmin()]

# Quantidade e percentual de produtos ativos e inativos
status_counts = df['status'].value_counts()
status_percent = (status_counts / len(df)) * 100

# Contagem de produtos por categoria
categoria_counts = df['categoria'].value_counts()

# Média de lucro por categoria
media_lucro_categoria = df.groupby('categoria')['lucro_unitario'].mean()

# Exibindo resultados
print("Produto mais lucrativo:")
print(produto_mais_lucrativo[['nome', 'lucro_unitario', 'margem_lucro_perc']])
print("\nProduto menos lucrativo:")
print(produto_menos_lucrativo[['nome', 'lucro_unitario', 'margem_lucro_perc']])
print("\nQuantidade de produtos por status:")
print(status_counts)
print("\nPercentual de produtos por status:")
print(status_percent.round(2))
print("\nNúmero de produtos por categoria:")
print(categoria_counts)
print("\nMédia do lucro unitário por categoria:")
print(media_lucro_categoria.round(2))

# Gráfico de barras: lucro unitário por produto
plt.figure(figsize=(10,6))
plt.bar(df['nome'], df['lucro_unitario'], color='skyblue')
plt.title('Lucro Unitário por Produto')
plt.ylabel('Lucro Unitário (R$)')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()

# Gráfico de pizza: proporção de produtos ativos vs inativos
plt.figure(figsize=(6,6))

```

```
plt.pie(status_counts, labels=status_counts.index, autopct='%1.1f%%',
colors=['green', 'red'])
plt.title('Proporção de Produtos Ativos e Inativos')
plt.show()
```

O que esse código faz:

- Cria um DataFrame com os dados que você forneceu.
- Calcula lucro unitário e margem de lucro.
- Identifica os produtos mais e menos lucrativos.
- Conta quantos produtos estão ativos e inativos, e calcula o percentual.
- Conta produtos por categoria e calcula a média de lucro por categoria.
- Exibe gráficos: barras para lucro unitário e pizza para proporção de status.

Se quiser, posso te ajudar a rodar ou adaptar esse código! Quer que eu explique algum ponto? “.

Explicação do Código por Linhas

Linhas 1-2

Importa as bibliotecas necessárias:

- pandas (apelidado de pd), para manipular e analisar dados em tabelas.
- matplotlib.pyplot (apelidado de plt), para criar gráficos.

Linhas 5-16

Cria arrays com os dados fictícios dos produtos, onde cada chave é o nome da coluna (ex: 'nome', 'status') e o valor é uma lista com os dados daquela coluna.

Linha 19

Converte os dados em um DataFrame do pandas, que é uma estrutura de tabela (como uma planilha Excel) para facilitar manipulação e análise.

Linha 22

Cria uma nova coluna no DataFrame chamada lucro_unitario, que é o preço de venda menos o preço de custo para cada produto.

Linha 23

Cria outra coluna chamada margem_lucro_perc, que calcula a margem de lucro percentual para cada produto. Ou seja, o lucro unitário dividido pelo preço de venda, multiplicado por 100 para obter porcentagem.

Linhas 26-27

df['lucro_unitario'].idxmax() retorna o índice do produto com maior lucro unitário.

df.loc[...] pega a linha completa do DataFrame correspondente a esse índice.

Mesma lógica para o produto com menor lucro.

Linhas 30-31

value_counts() conta quantas vezes cada valor aparece na coluna status (ativo, inativo).

Calcula o percentual desses status em relação ao total de produtos.

Linha 34

Conta quantos produtos existem em cada categoria.

Linha 37

Agrupa os produtos pela categoria e calcula a média do lucro unitário para cada grupo.

Linhas 40-51

Exibe no console o produto mais lucrativo, mostrando nome, lucro unitário e margem percentual.

Exibe o produto menos lucrativo.

Imprime contagens, percentuais, quantidade por categoria e média de lucro por categoria.

round(2) arredonda os valores para 2 casas decimais para facilitar a leitura.

Linhas 54-60

Cria um gráfico de barras para o lucro unitário por produto.

Linhas 63-66

Cria um gráfico de pizza para a proporção de produtos ativos e inativos.

Resultado da Análise dos Dados da Tabela Produtos

O código foi executado no website chamado Google Colab, visto que é um ambiente de desenvolvimento em python que não necessita de instalação de componentes localmente, facilitando a utilização e manipulação da linguagem no desenvolvimento da atividade.

Após executar o código, ele imprime a seguinte resposta:

“

Produto mais lucrativo:

```
nome      X-Tudo
lucro_unitario      9
margem_lucro_perc      50
Name: 8, dtype: object
```

Produto menos lucrativo:

```
nome      Coca-Cola 600ml
lucro_unitario      4
margem_lucro_perc      50
Name: 4, dtype: object
```

Quantidade de produtos por status:

```
ativo  8
inativo 2
Name: status, dtype: int64
```

Percentual de produtos por status:

```
ativo  80.0
inativo 20.0
Name: status, dtype: float64
```

Número de produtos por categoria:

```
Lanche      5
Bebida      3
Acompanhamento  2
Name: categoria, dtype: int64
```

Média do lucro unitário por categoria:

```
categoria
Acompanhamento  6.5
Bebida          3.9
Lanche          7.0
Name: lucro_unitario, dtype: float64
```

“

Analisando esses dados, é possível entender que:

1. Lucro dos produtos

- O produto mais lucrativo é o X-Tudo, com R\$ 9,00 de lucro por unidade e uma margem de lucro de 50%.
- O menos lucrativo é a Coca-Cola 600ml, com R\$ 4,00 de lucro e também 50% de margem.

2. Status dos produtos

- 80% dos produtos estão ativos (8 de 10).
- 20% estão inativos (2 de 10).

Muitos produtos para venda, com poucos itens inativos.

3. Categorias

A maioria dos produtos está na categoria Lanche (5 produtos).

- Bebidas vêm em seguida com 3 produtos.
- Acompanhamentos têm apenas 2 produtos.

O cardápio é mais focado em lanches, já que provém de uma hamburgueria.

4. Média de Lucro por Categoria

- Lanche tem a maior média de lucro unitário (R\$ 7,00).
- Acompanhamento vem logo atrás com R\$ 6,50.
- Bebidas têm lucro médio mais baixo (R\$ 3,90).

Normalmente a venda de lanches gera mais lucro, enquanto as bebidas, são vendidas como acompanhamentos.

Gráficos

Foram criados dois gráficos para essa tabela por meio biblioteca matplotlib.pyplot (apelidado de plt): Um gráfico de barras para o lucro unitário por produto e um gráfico de pizza para a proporção de produtos ativos e inativos.

Como não foi de agrado ter que baixar o python localmente e utilizar algum software gráfico como o VS Code para imprimir os gráficos, foi utilizado então o website Google Colab, no qual permite a execução de códigos em tempo real sem ter componentes instalados localmente.

Gráfico - 1

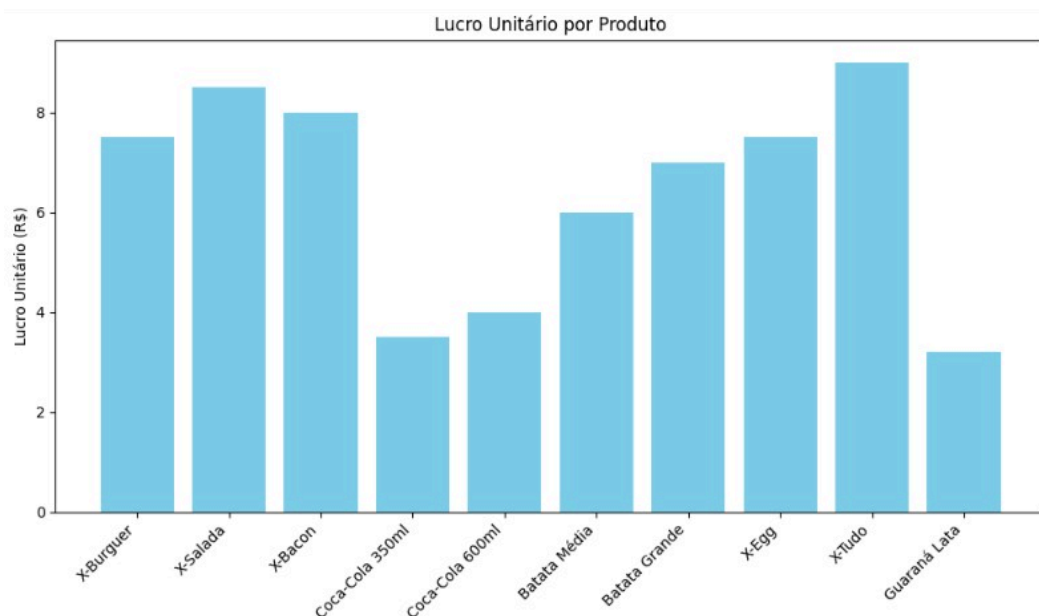
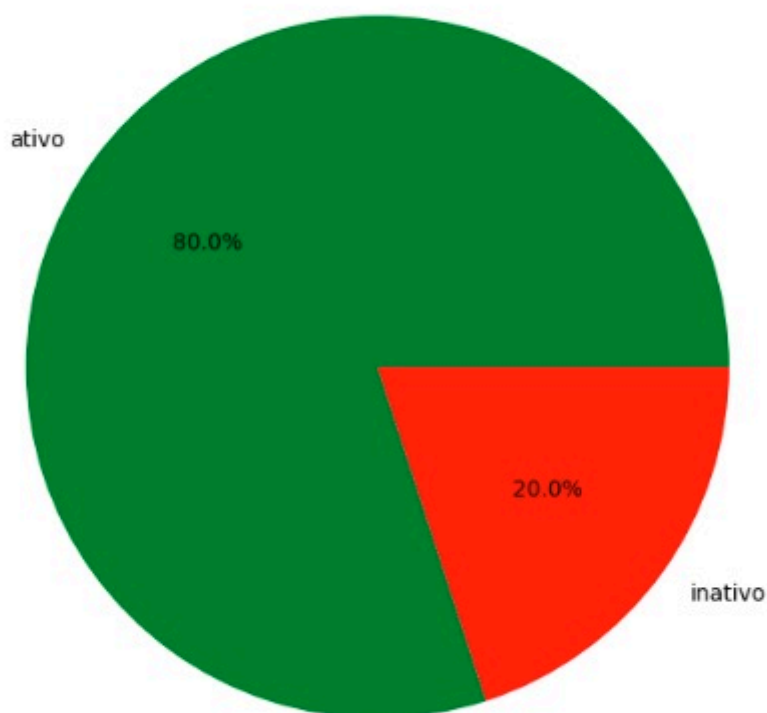


Gráfico - 2

Proporção de Produtos Ativos e Inativos



Analizando os Gráficos

Gráfico - 1

Esse gráfico mostra uma barra para cada produto, com a altura da barra representando quanto de lucro cada unidade vendida gera. Fácilmente de comparar

os produtos, O X-Tudo será a barra mais alta, com R\$ 9,00 de lucro por unidade, é o produto mais lucrativo. Produtos como Coca-Cola 350ml, Guaraná Lata, e Batata Média terão barras bem mais baixas (lucro menor). X-Salada, X-Egg e X-Burguer têm barras altas, indicando bons lucros.

Gráfico - 2

Esse gráfico mostra um círculo dividido em fatias, representando a quantidade relativa de produtos ativos e inativos no total. A maior fatia (80%) será verde e representará os produtos ativos. Uma fatia menor (20%) será vermelha, representando os inativos. Ou seja, o gráfico mostra que a maioria dos produtos está ativa.

Tabela Insumos

Após a execução do prompt para a análise da tabela dos insumos, a IA fez a seguinte resposta:

“ Claro! Abaixo está o código Python completo que atende a todas as exigências do Prompt 2 – Análise de Insumos, utilizando os dados fornecidos e as bibliotecas pandas e matplotlib:

✓ Código Python – Análise de Insumos

```
import pandas as pd
```

```
import matplotlib.pyplot as plt
```

```
# Dados dos insumos
```

```
dados_insumos = {  
    'nome': [  
        'Pão Brioche', 'Carne Bovina 150g', 'Bacon', 'Cebola', 'Batata Congelada',  
        'Alface', 'Tomate', 'Queijo Cheddar', 'Presunto', 'Ovo'  
    ],  
    'categoria': [  
        'Padaria', 'Carne', 'Embutidos', 'Hortifruti', 'Congelados',  
        'Hortifruti', 'Hortifruti', 'Laticínios', 'Embutidos', 'Granja'  
    ],  
    'medida': [  
        'un', 'kg', 'kg', 'kg', 'kg',  
        'kg', 'kg', 'kg', 'kg', 'un'  
    ],  
    'estoque': [  
        40, 25, 10, 60, 20,  
        30, 18, 12, 9, 60  
    ],  
    'estoque_minimo': [  
        50, 30, 15, 40, 25,  
        20, 25, 15, 10, 50  
    ],  
    'preco_custo': [  
        0.60, 28.00, 22.00, 4.00, 12.00,  
        3.00, 5.00, 18.00, 19.00, 0.50  
    ],  
    'custo_kg_l': [  
        10.00, 28.00, 22.00, 4.00, 12.00,  
        3.00, 5.00, 18.00, 19.00, 12.00  
    ]  
}
```

```

# Criando o DataFrame
df = pd.DataFrame(dados_insumos)

# Custo total em estoque por insumo (estoque * custo por kg/l)
df['custo_total_estoque'] = df['estoque'] * df['custo_kg_l']

# Insumos em situação crítica (estoque < estoque_minimo)
df['critico'] = df['estoque'] < df['estoque_minimo']
insumos_criticos = df[df['critico'] == True]

# Percentual de insumos críticos
percentual_criticos = (len(insumos_criticos) / len(df)) * 100

# Custo total dos insumos críticos
custo_total_criticos = insumos_criticos['custo_total_estoque'].sum()

# Agrupamento por categoria dos insumos críticos
categorias_criticas = insumos_criticos['categoria'].value_counts()

# Top 5 insumos mais caros em valor total de estoque
top_5_caros = df.nlargest(5, 'custo_total_estoque')

# ----- Resultados no console -----
print("Insumos em estoque crítico:")
print(insumos_criticos[['nome', 'estoque', 'estoque_minimo']])

print("\nPercentual de insumos em situação crítica:
{:.2f}%".format(percentual_criticos))

print("\nCategorias mais afetadas por estoque crítico:")
print(categorias_criticas)

print("\nCusto total dos insumos abaixo do mínimo:
R${:.2f}".format(custo_total_criticos))

print("\nTop 5 insumos mais caros (valor total em estoque):")
print(top_5_caros[['nome', 'custo_total_estoque']])

# ----- Gráficos -----

# Gráfico de barras – Custo total em estoque por insumo
plt.figure(figsize=(10,6))
plt.bar(df['nome'], df['custo_total_estoque'], color='orange')

```

```
plt.xticks(rotation=45, ha='right')
plt.title('Custo Total em Estoque por Insumo')
plt.ylabel('Custo Total (R$)')
plt.tight_layout()
plt.show()
```

Gráfico de barras – Apenas insumos críticos

```
plt.figure(figsize=(10,6))
plt.bar(insumos_criticos['nome'], insumos_criticos['custo_total_estoque'], color='red')
plt.xticks(rotation=45, ha='right')
plt.title('Custo Total de Insumos em Estoque Crítico')
plt.ylabel('Custo Total (R$)')
plt.tight_layout()
plt.show()
```

✓ O que o código faz

- Importação de bibliotecas = Usa pandas para manipular os dados e matplotlib.pyplot para os gráficos.
- Criação da tabela de dados = Simula os dados da planilha com 10 insumos.
- Identificação de insumos críticos = Verifica se o estoque está abaixo do mínimo e cria uma coluna booleana (True ou False).
- Cálculo do percentual de críticos = Percentual de insumos com estoque crítico.
- Custo total por insumo = Calcula o custo total de cada insumo baseado no estoque.
- Agrupamento por categoria = Mostra quais categorias têm mais insumos críticos.
- Top 5 insumos mais caros = Ordena pelos mais caros em valor de estoque.
- Gráficos = Gera dois gráficos: um geral e um só com os críticos.

Se você quiser que eu te explique linha por linha como no outro prompt, é só falar. Posso também converter esse resultado para Word ou PDF se necessário. “

Explicação do Código por Linhas

Linhas 1-2

Importa as bibliotecas necessárias:

- pandas (apelidado de pd), para manipular e analisar dados em tabelas.
- matplotlib.pyplot (apelidado de plt), para criar gráficos.

Linhas 5-34

Cria arrays com os dados fictícios dos insumos, onde cada chave é o nome da coluna (ex: 'nome', 'categoria', etc) e o valor é uma lista com os dados daquela coluna.

Linha 37

Converte os dados em um DataFrame do pandas, que é uma estrutura de tabela (como uma planilha Excel) para facilitar manipulação e análise.

Linha 40

Cria uma nova coluna no DataFrame chamada `custo_total_estoque`. Calcula o custo total de cada insumo em estoque: multiplicando a quantidade em estoque pelo custo por kg/l.

Linha 43-44

Cria uma nova coluna no DataFrame chamada `critico`, Essa coluna terá `True` para os insumos com estoque abaixo do mínimo e `False` caso contrário.

Filtra o DataFrame para manter apenas os insumos com estoque crítico (onde `critico` é `True`). Onde o resultado é salvo na variável `insumos_criticos`.

Linhas 47

Calcula o percentual de insumos em situação crítica.

`len(insumos_criticos)` = número de insumos abaixo do mínimo.

`len(df)` = número total de insumos

Multiplica por 100 para obter o valor em porcentagem.

Linhas 50

Cria uma nova coluna no DataFrame chamada "`custo_total_criticos`" com o custo total em estoque de cada insumo.

Faz a multiplicação entre a quantidade (estoque) e o custo por kg/l (`custo_kg_l`).

Linha 53

Conta quantos insumos críticos há em cada categoria (ex: quantos da Hortifruti, Embutidos, etc).

Linha 56

Seleciona os 5 insumos com maior custo total em estoque usando a função `nlargest()`.

Linhas 59-70

Exibe os insumos críticos junto com seus estoques e estoques mínimos.

Mostra o percentual calculado de insumos abaixo do mínimo, com 2 casas decimais.

Exibe quantos insumos em estoque crítico existem por categoria.

Mostra o valor total (em reais) dos insumos críticos.

Exibe os 5 insumos com o maior custo total de estoque.

Linhas 75-81

Cria um gráfico de barras para o custo total por insumo.

Linhas 84-90

Cria um gráfico de barras para os insumos em estoque crítico.

Resultado da Análise dos Dados da Tabela Insumos

Como a tabela produto, o código foi executado no website Google Colab. Após executar o código, ele imprime a seguinte resposta:

“

Insumos em estoque crítico:

	nome	estoque	estoque_minimo
0	Pão Brioche	40	50
1	Carne Bovina 150g	25	30
2	Bacon	10	15
4	Batata Congelada	20	25
6	Tomate	18	25
7	Queijo Cheddar	12	15
8	Presunto	9	10

Percentual de insumos em situação crítica: 70.00%

Categorias mais afetadas por estoque crítico:

categoria

Embutidos 2

Padaria 1

Carne 1

Congelados 1

Hortifruti 1

Laticínios 1

Name: count, dtype: int64

Custo total dos insumos abaixo do mínimo: R\$2037.00

Top 5 insumos mais caros (valor total em estoque):

	nome	custo_total_estoque
9	Ovo	720.0
1	Carne Bovina 150g	700.0
0	Pão Brioche	400.0
3	Cebola	240.0
4	Batata Congelada	240.0

“

Analisando esses dados, é possível entender que:

1. Insumos em estoque crítico

7 dos 10 insumos estão com estoque abaixo do mínimo necessário, representando 70% do total. Isso indica risco operacional significativo, podendo afetar diretamente a produção ou entrega de produtos.

2. Categorias Mais Afetadas

A categoria "Embutidos" é a mais impactada, com 2 insumos críticos: Bacon e Presunto.

- Outras categorias com 1 insumo crítico cada incluem:
- Padaria (Pão Brioche)
- Carne (Carne Bovina 150g)
- Congelados (Batata Congelada)
- Hortifruti (Tomate)
- Laticínios (Queijo Cheddar)

Isso mostra que o problema está distribuído, mas com maior concentração em produtos de origem animal ou perecíveis.

3. Custo Total dos Insumos Críticos

O valor total necessário para repor os itens abaixo do mínimo é de aproximadamente R\$ 2.037,00.

Esse número ajuda no planejamento financeiro: é o valor mínimo a ser investido para regularizar o estoque crítico.

4. Top 5 Insumos Mais Caros em Estoque

1. Ovo (720,00)
2. Carne Bovina 150g (700,00)
3. Pão Brioche (400,00)
4. Cebola (240,00)
5. Batata Congelada (240,00)

É possível apontar que 3 desses 5 também estão na lista de insumos críticos, o que eleva o risco financeiro caso faltem.

Gráficos

Também foram criados dois gráficos para essa tabela, dessa vez dois gráficos de barras, um para o custo total por insumo e outro para os insumos em estoque crítico.

Gráfico - 1

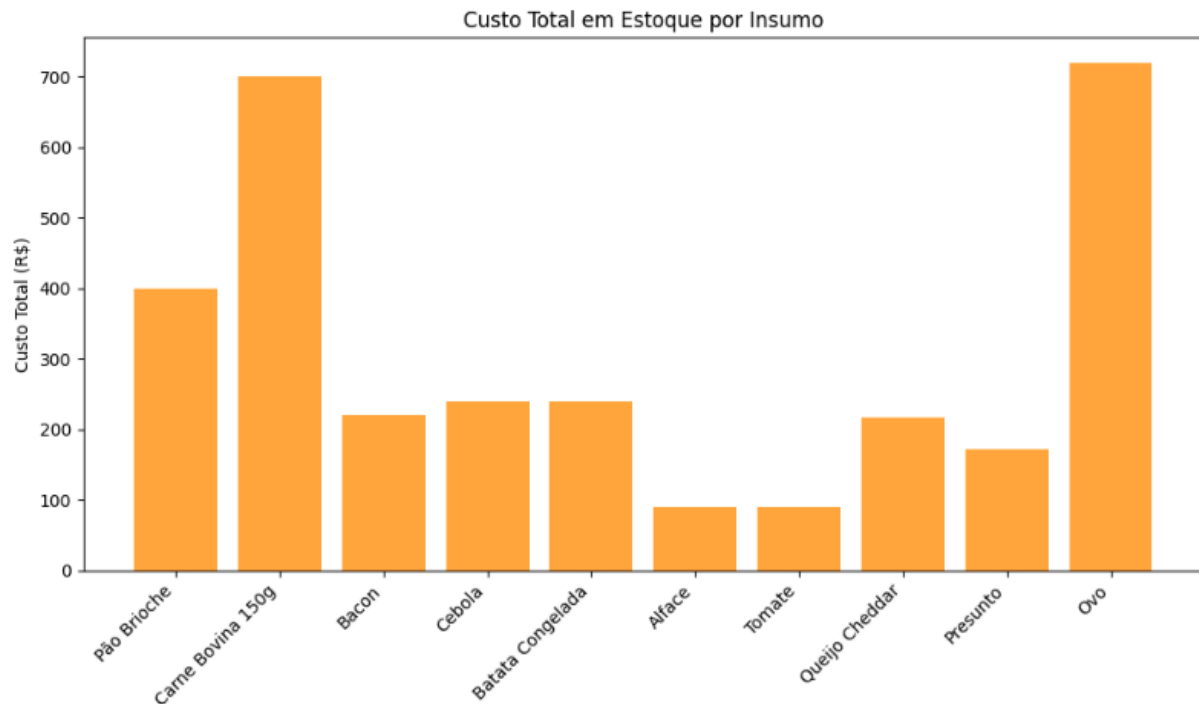
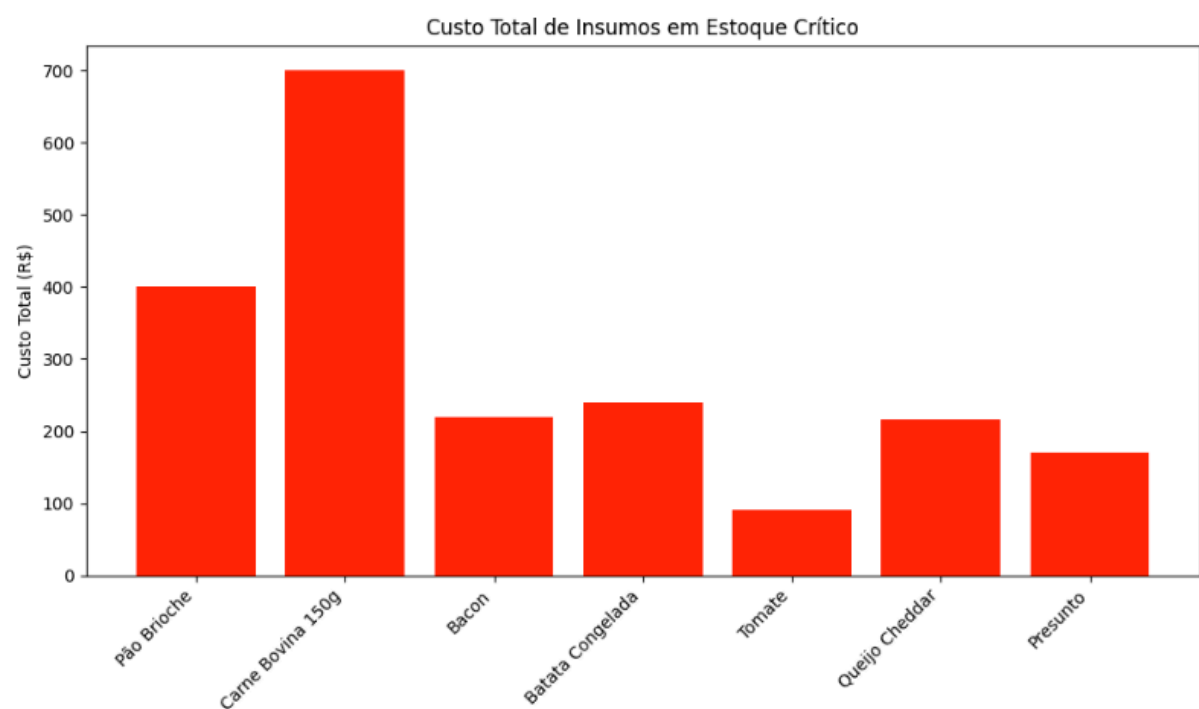


Gráfico - 2



Analisando os Gráficos

Gráfico - 1

Esse gráfico de barras exhibe para cada insumo o valor total investido em estoque. Ajudando a visualizar onde está o maior investimento. Isso pode indicar uma possível necessidade de atenção em controle de estoque. Os insumos seguem a ordem:

1. Ovo
2. Carne Bovina
3. Pão Brioche
4. Cebola
5. Batata Congelada
6. Presunto
7. Tomate
8. Alface

Gráfico - 2

Este gráfico foca apenas nos insumos cujo estoque está abaixo do mínimo exigido. O estoque crítico representa um problema significativo, especialmente porque os insumos mais caros estão entre os que faltam, o que quer dizer um grande impacto financeiro caso se perca o controle desses insumos. Sendo os insumos listados:

1. Carne Bovina
2. Pão Brioche
3. Batata Congelada
4. Queijo Cheddar
5. Presunto
6. Bacon
7. Tomate

Assim sendo de urgência uma reposição de estoque, visto que ingredientes essenciais estão em estado crítico para a confecção de mais hambúrgueres.

Conclusão

A linguagem Python mostrou-se essencial para a análise de dados por reunir simplicidade, versatilidade e uma ampla variedade de bibliotecas especializadas, como Pandas, NumPy e Matplotlib. Essa combinação permite transformar dados brutos em informações claras e relevantes, facilitando a tomada de decisões em diferentes áreas. Além disso, com o avanço das ferramentas de Inteligência Artificial, o processo de análise pode se tornar ainda mais ágil e assertivo, já que

algoritmos de IA auxiliam na detecção de padrões, previsões e automação de tarefas repetitivas. Dessa forma, Python, aliado à IA, representa um caminho cada vez mais estratégico para explorar e potencializar o valor dos dados.

Bibliografia

<https://www.tiobe.com/tiobe>

<https://pandas.pydata.org>

<https://matplotlib.org>

<https://numpy.org>