# PREPARAÇÃO DOS DADOS POR ChatGPT (Open AI)

Uma análise de dados pode envolver várias etapas e técnicas, dependendo do objetivo específico e do conjunto de dados em questão. No entanto, a parte principal de uma análise de dados é a análise propriamente dita, onde os dados são explorados e interpretados para extrair insights valiosos. Isso inclui:

Preparação de dados: Antes de serem analisados, os dados precisam ser limpos, tratados e transformados para serem adequados para análise.

Análise Exploratória: Nessa etapa, os dados são explorados para encontrar padrões, tendências e relações. Isso pode incluir a visualização de dados, estatísticas descritivas e análise de grupos.

Análise Preditiva: Se o objetivo é prever um resultado futuro, essa etapa pode envolver técnicas de aprendizado de máquina e modelagem estatística.

Comunicação de resultados: A análise é inútil se os resultados não forem comunicados e interpretados corretamente para as pessoas certas. A comunicação eficaz dos resultados é crucial para garantir que as descobertas da análise sejam entendidas e aplicadas.

Tomada de decisão: A análise de dados deve ser usada para ajudar a tomar decisões informadas e para apoiar a estratégia de negócios.

Aqui estão alguns exemplos de como cada etapa da análise de dados pode ser implementada em Python:

## Códigos

### Preparação de dados:

import pandas as pd

**#carregando dados de um arquivo CSV**

data = pd.read\_csv("data.csv")

**#eliminando valores faltantes**

data.dropna(inplace=True)

**#convertendo uma coluna para outro tipo**

data["column\_name"] = data["column\_name"].astype(float)

### Análise Exploratória:

import seaborn as sns

import matplotlib.pyplot as plt

**#plotando um histograma**

sns.histplot(data["column\_name"])

plt.show()

**#calculando a correlação entre duas colunas**

corr = data.corr()

print(corr)

**#plotando um gráfico de dispersão**

sns.scatterplot(x="coluna\_x", y="coluna\_y", data=data)

plt.show()

### Análise Preditiva:

from sklearn.linear\_model import LinearRegression

from sklearn.model\_selection import train\_test\_split

**#separando as variáveis de entrada e saída**

X = data[["coluna\_x", "coluna\_y"]]

y = data["coluna\_z"]

#dividindo os dados em conjunto de treino e teste

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2)

**#treinando o modelo de regressão linear**

model = LinearRegression()

model.fit(X\_train, y\_train)

**#fazendo previsões**

y\_pred = model.predict(X\_test)

### Comunicação de resultados:

import matplotlib.pyplot as plt

**#criando um gráfico de barras**

plt.bar(data["coluna\_x"], data["coluna\_y"])

plt.xlabel("Coluna X")

plt.ylabel("Coluna Y")

plt.title("Título do gráfico")

plt.show()

**#criando um relatório em PDF**

from fpdf import FPDF

pdf = FPDF()

pdf.add\_page()

pdf.set\_font("Arial", size=12)

pdf.cell(200, 10, txt="Resultados da Análise de Dados", ln=1, align="C")

pdf.cell(200, 10, txt=f"Acurácia do modelo: {accuracy:.2f}", ln=2, align="C")

pdf.output("report.pdf")

### Tomada de decisão:

**#criando um gráfico de pizza**

labels = data["coluna\_x

sizes = data["coluna\_y"]

fig1, ax1 = plt.subplots() ax1.pie(sizes, labels=labels, autopct='%1.1f%%', shadow=True, startangle=90) ax1.axis('equal') **# Equal aspect ratio ensures that pie is drawn as a circle.**

plt.show()

**#usando a análise para tomar uma decisão estratégica**

if data["coluna\_x"].mean() > data["coluna\_y"].mean():

print("A coluna X tem valores médios maiores do que a coluna Y, então devemos investir mais nessa área.")

else: print("A coluna Y tem valores médios maiores do que a coluna X, então devemos investir mais nessa área.")

**Esses são apenas alguns exemplos de como cada etapa da análise de dados pode ser implementada em Python. A complexidade e as técnicas específicas usadas podem variar dependendo do conjunto de dados e do objetivo da análise.**