**Step 1: Importing Necessary Libraries**

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

**Step 2: Loading the Data**

# Load the data from a CSV file

file\_path = 'path\_to\_your\_csv\_file.csv' # replace with the actual path to your CSV file

data = pd.read\_csv(file\_path)

# Display the first few rows of the dataset

data.head()

**Step 3: Data Exploration**

# Get basic information about the dataset

data.info()

# Get summary statistics

data.describe()

# Check for missing values

missing\_values = data.isnull().sum()

print(missing\_values)

# Visualize missing values

plt.figure(figsize=(10, 6))

sns.heatmap(data.isnull(), cbar=False, cmap='viridis')

plt.title('Missing Values Heatmap')

plt.show()

# Check for duplicate rows

duplicate\_rows = data.duplicated().sum()

print(f'Duplicate rows: {duplicate\_rows}')

# Visualize the distribution of data for a few sectors

sectors\_to\_plot = ['METALLURGICAL INDUSTRIES', 'MINING', 'POWER'] # example sectors

for sector in sectors\_to\_plot:

plt.figure(figsize=(10, 6))

sns.lineplot(data=data, x='Year', y=sector)

plt.title(f'{sector} Over Time')

plt.xlabel('Year')

plt.ylabel('Value')

plt.grid(True)

plt.show()

**Step 4: Data Manipulation**

# Fill missing values (example: forward fill)

data.fillna(method='ffill', inplace=True)

# Creating new columns for analysis

data['Total'] = data.sum(axis=1)

# Grouping data by specific criteria (example: year)

grouped\_data = data.groupby('Year').sum()

# Reset index if needed

grouped\_data.reset\_index(inplace=True)

# Sorting the data by a specific column

sorted\_data = data.sort\_values(by='Total', ascending=False)

# Filtering data (example: only rows where 'METALLURGICAL INDUSTRIES' > 1000)

filtered\_data = data[data['METALLURGICAL INDUSTRIES'] > 1000]

# Creating new features (example: percentage change)

data['Metallurgical\_Percent\_Change'] = data['METALLURGICAL INDUSTRIES'].pct\_change()

# Displaying the modified dataset

data.head()

**Step 5: Visualization**

# Heatmap of sector performance over years

plt.figure(figsize=(16, 12))

sns.heatmap(data.set\_index('Year').T, cmap='coolwarm', annot=True, fmt='.2f')

plt.title('Sector Performance Heatmap')

plt.show()

# Bar plot for top sectors in a specific year

year\_to\_plot = '2005-06' # example year

data\_year = data[['Sector', year\_to\_plot]].sort\_values(by=year\_to\_plot, ascending=False)

plt.figure(figsize=(14, 8))

sns.barplot(data=data\_year, x='Sector', y=year\_to\_plot)

plt.title(f'Top Sectors in {year\_to\_plot}')

plt.xticks(rotation=90)

plt.show()

# Treemap for investment distribution across sectors

import squarify

year\_to\_plot = '2005-06' # example year

sizes = data[year\_to\_plot].values

labels = data['Sector'].values

colors = plt.cm.tab20c(np.linspace(0, 1, len(sizes)))

plt.figure(figsize=(12, 8))

squarify.plot(sizes=sizes, label=labels, color=colors, alpha=0.8)

plt.title(f'Investment Distribution Across Sectors in {year\_to\_plot}')

plt.axis('off')

plt.show()

**Step 6: Saving the Modified Data**

# Save the modified dataset to a new CSV file

modified\_file\_path = 'path\_to\_save\_modified\_csv\_file.csv' # replace with the desired path to save the CSV file

data.to\_csv(modified\_file\_path, index=False)