Rajalakshmi Engineering College (An Autonomous Institution) Rajalakshmi Nagar, Thandalam- 602105

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

AD23632 - Framework for Data Visualization and Analytics

Mini Project: INDIA STOCK MARKET DATA ANALYSIS AND VISUALIZATION

Report submitted by

REGISTRATION NUMBER: 231501077

STUDENT NAME : KISHANTH RAJA M

YEAR : 2023-2027

SUBJECT CODE : AD23632

Table of Contents

Chapter 1: Abstract

This project focuses on analyzing the relationship between India's key economic indicators and stock market performance. Using the dataset *India_Stock_Market_Data.xlsx*, which contains yearly data on GDP Growth, Inflation, Exchange Rate (INR/USD), and the BSE SENSEX index, interactive dashboards were created in Power BI and Tableau.

The objective was to visually interpret how macroeconomic factors such as GDP growth and inflation influence the stock market and currency movement over time. The dashboards enable trend analysis, pattern identification, and decision-making support for investors and policy analysts.

Through visualization and analysis, the study highlights that periods of stable inflation and consistent GDP growth often align with stronger SENSEX performance and a relatively stable rupee.

Chapter 2: Introduction

In a developing economy like India, financial markets play a vital role in shaping economic development and reflecting investor sentiment. The performance of the Indian stock market, represented by the **BSE SENSEX**, is closely linked to key economic indicators such as GDP growth, inflation, and exchange rate fluctuations. These indicators collectively influence capital flows, consumer confidence, and foreign investment decisions.

Understanding the relationship between these indicators is crucial for investors, policymakers, and economists. For instance, higher GDP growth typically signals a thriving economy, attracting investment and pushing stock prices upward. On the other hand, rising inflation and currency depreciation can lead to uncertainty and reduced investment activity.

This project aims to visualize and analyze these interconnected factors using historical data. By leveraging the analytical power of **Power BI** and **Tableau**, complex numerical data is transformed into visually interpretable formats. Such visualization not only aids in understanding long-term trends but also supports data-driven decision-making for investors and analysts in the Indian financial ecosystem.

Chapter 3: Dataset Description

The dataset titled **"India_Stock_Market_Data.xlsx"** provides annual data on India's key macroeconomic and financial indicators. It forms the foundation for understanding how fluctuations in economic performance influence stock market behaviour and currency strength.

Attributes included in the dataset:

- **Year:** Represents the calendar year of observation.
- **GDP Growth (%):** Indicates the annual percentage change in India's Gross Domestic Product, measuring the pace of economic growth.
- **Inflation (%):** Represents the annual inflation rate, calculated based on changes in consumer prices. It reflects the overall cost of living and purchasing power.
- **Exchange Rate (INR/USD):** Shows the average annual value of the Indian Rupee against the US Dollar, highlighting the strength or weakness of the domestic currency.
- **SENSEX:** Represents the average or closing value of the Bombay Stock Exchange (BSE) SENSEX index, which tracks the performance of 30 major companies across sectors.

This dataset spans multiple years, capturing periods of economic expansion and slowdown. It enables comparative and relational analysis between key economic variables. The dataset was used as the base for creating dashboards that visualize patterns such as GDP–Inflation correlation, currency depreciation trends, and stock market movements.

Chapter 4: Objective

To analyze and visualize how **GDP Growth**, **Inflation**, and **Exchange Rate** impact India's **BSE SENSEX** using **Python**, **Power BI**, and **Tableau**, converting raw economic data into meaningful, data-driven insights.

The major objectives of the study are as follows:

1. **To examine the relationship between macroeconomic indicators and market performance:**
   This objective focuses on studying how GDP Growth (%), Inflation (%), and Exchange Rate (INR/USD) affect the behaviour of the SENSEX index. The goal is to identify whether economic growth, price stability, and currency fluctuations have a direct or indirect influence on stock market trends.

2. **To clean, preprocess, and structure the dataset for analysis:**
   The dataset *India_Stock_Market_Data.xlsx* is prepared for analysis by standardizing column names, correcting data types, handling missing values, and ensuring accuracy. A well-prepared dataset ensures that the subsequent analysis and visualizations are reliable and error-free.

3. **To conduct exploratory data analysis (EDA) using Python:**
   Through Python libraries such as pandas, numpy, matplotlib, and seaborn, data patterns, trends, and correlations are examined. Visualization tools are used to identify the interactions among economic variables and their influence on market behaviour.

4. **To create interactive dashboards for visualization and interpretation:**
   Using Power BI and Tableau, dynamic dashboards are designed to visualize GDP Growth, Inflation, Exchange Rate, and SENSEX trends. These dashboards allow users to interactively explore year-wise data, compare indicators, and interpret relationships between variables effectively.

5. **To derive insights that support economic and financial understanding:**
   The final objective is to interpret data-driven findings that highlight how macroeconomic stability influences stock market growth. The insights can help investors, analysts, and policymakers understand the broader economic environment and make informed decisions.

Chapter 5: Methodology

The project methodology involves multiple stages — data preparation, visualization, and interpretation — to ensure accuracy and clarity of insights.

a. Data Preparation

- The dataset was loaded into Excel and verified for completeness.
- Data types were standardized: *Year* as numeric, *Inflation (%)* and *GDP Growth (%)* as float values.
- Missing or inconsistent values were checked and corrected where necessary.
- Duplicates were removed to maintain data integrity.

b. Data Visualization

- The cleaned dataset was imported into **Power BI** and **Tableau**.
- Different visualization techniques were applied to highlight relationships between variables.
- Dashboards were designed using charts such as area graphs, line plots, bar charts, tree maps, and pie charts.

c. Analytical Framework

- Economic theory was used to interpret visual patterns.
- Comparative trends between years were analysed to understand the impact of inflation or currency depreciation on GDP and stock indices.
- Findings were cross-verified between Power BI and Tableau outputs to ensure consistency.

d. Interpretation and Reporting

- Insights derived from dashboards were summarized and analysed in a written format.
- Visualizations were compiled into an interactive story format for easier understanding.

Chapter 6: Python Implementation

Python serves as the core environment for performing data preprocessing, feature engineering, and exploratory data analysis in this project. Libraries such as **pandas**, **numpy**, **matplotlib**, and **seaborn** were employed for data handling, statistical summarization, and visualization. The process began with importing the *India_Stock_Market_Data.xlsx* file, standardizing column names, converting values to numeric formats, and systematically treating missing entries through forward or backward filling to ensure data consistency.

Feature engineering was then carried out to derive new indicators, including **Real GDP Estimate** (GDP Growth – Inflation) and **SENSEX per USD** (SENSEX divided by the exchange rate), to better understand how macroeconomic performance interacts with market outcomes. Rolling averages and year-over-year percentage changes were also computed to capture medium-term volatility and growth trends.

Visualization formed the analytical core of this implementation. **Line plots** were used to display GDP growth, inflation, and SENSEX movement over time, while **scatter plots** highlighted relationships between inflation and market indices. **Correlation heatmaps** further revealed the strength and direction of dependencies among all economic variables. These visual tools made the findings more interpretable and supported evidence-based conclusions.

All generated plots were saved and exported for integration into **Power BI** and **Tableau** dashboards, ensuring alignment between Python-based EDA and business intelligence reporting. In conclusion, Python provided a **robust, transparent, and reproducible analytical framework** that transformed raw economic data into structured insights, serving as the foundation for effective data visualization and decision-making.

Chapter 7: Power BI Dashboard

Power BI was used to develop a structured and interactive dashboard that visually represents the dataset. The dashboard components include:

- **Area Chart:** Displays combined GDP Growth (%) and Inflation (%) trends across years, highlighting parallel or contrasting movements.

- **Bar Chart:** Illustrates GDP Growth by Year, allowing quick comparison of annual performance.

- **Tree map:** Groups GDP Growth by Inflation categories, visually emphasizing periods of economic strength or weakness under different inflation conditions.

- **Combo Chart:** Depicts the relationship between Exchange Rate (INR/USD) and GDP Growth, showing how currency value fluctuates with economic expansion or contraction.

- **Pie Chart:** Represents the distribution of inflation rates across years, helping identify high or low inflation periods.

- **Area Chart (Dual Axis):** Plots the Exchange Rate against the SENSEX index, highlighting potential correlations between currency movements and market performance.

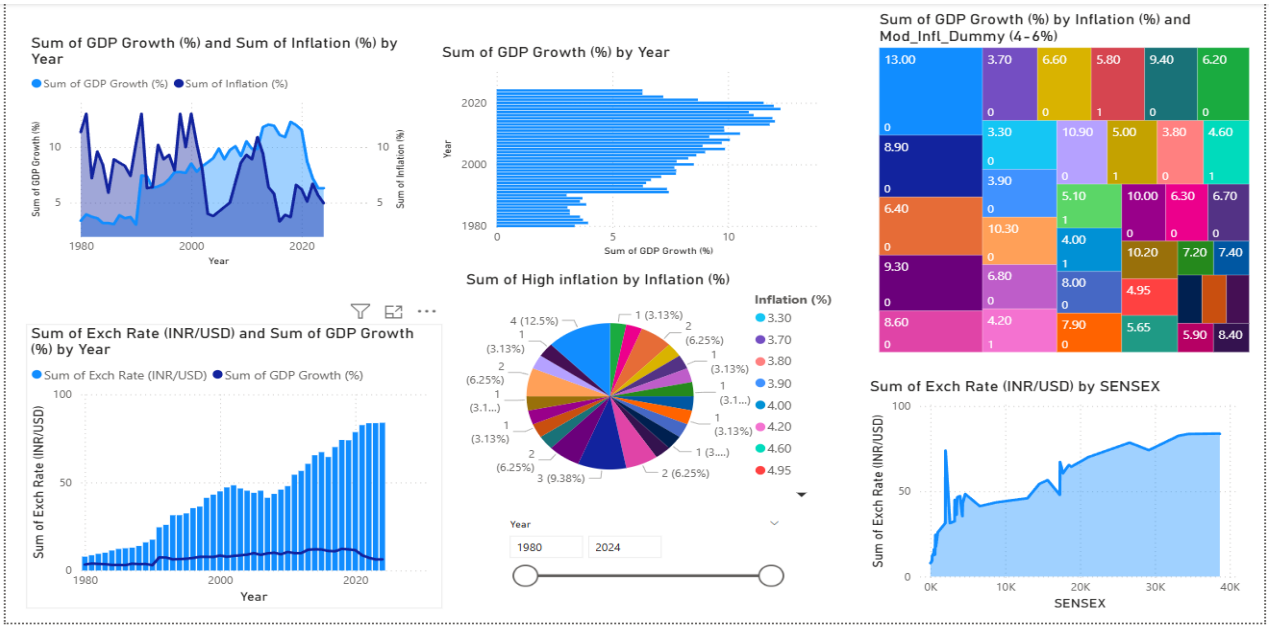


Fig 7.1: Power BI Dashboard

Chapter 8: Tableau Dashboard

Line Chart (Dual Axis)

- Used for inflation rate (%) and GDP growth (%) over the years.
- Shows long-term trends, cycles, and the relationship between these two key macroeconomic variables.

Bar Chart

- Used for exchange rate (INR/USD) across years.
- Highlights the steady depreciation of the currency, with color intensity indicating faster rate changes during economic events.

Scatter Plot

- Plots GDP growth (%) on the x-axis and SENSEX returns (%) on the y-axis, with each point representing a year.
- Visualizes correlation and dispersion between economic growth and stock market performance, and identifies outlier years.

Area/Line Chart (Dual Y Axis)

- Used for SENSEX vs exchange rate (INR/USD) and GDP growth vs inflation.
- Area charts emphasize cumulative trends, such as SENSEX growth or inflation cycles, while line overlays allow direct comparison across variables and highlight peaks and drawdowns over time.

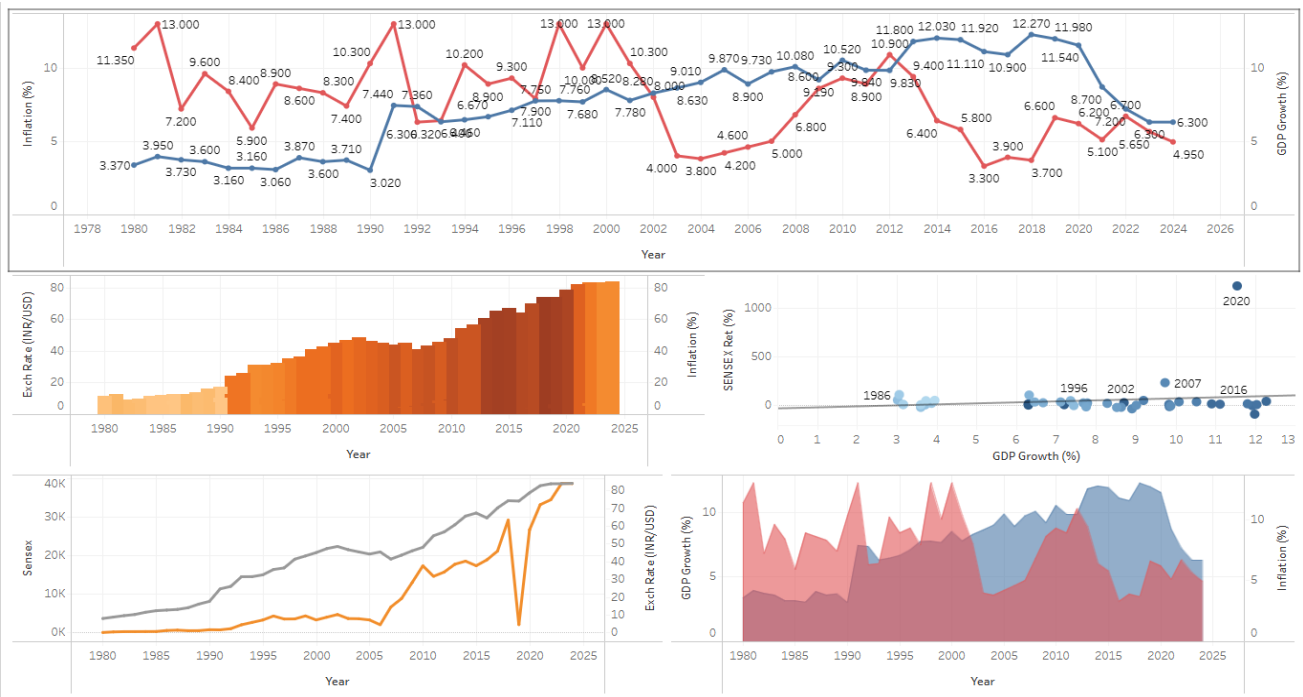


Fig 8.1: Tableau Dashboard

Chapter 9: Analysis and Discussions

The combined visual analysis from Power BI and Tableau reveals significant insights into India's economic landscape:

1. **GDP and Stock Market Relationship:**
   A positive correlation exists between GDP growth and SENSEX movement. Economic expansion drives market optimism and capital inflows.
2. **Inflation Impact:**
   Moderate inflation supports market growth, while high inflation dampens investment confidence and reduces purchasing power.
3. **Exchange Rate Influence:**
   The Rupee has shown gradual depreciation over time. However, controlled depreciation aligns with sustained GDP growth, supporting exports and balanced growth.
4. **Macroeconomic Stability:**
   Years with stable GDP growth, manageable inflation, and steady exchange rates coincide with consistent SENSEX appreciation.
5. **Volatility Patterns:**
   Global crises or domestic disruptions reflect sharp drops in GDP growth and market value, confirming market sensitivity to economic conditions.

Overall, the analysis demonstrates the interconnected nature of economic indicators and their combined effect on stock market performance.

Chapter 10: Conclusion

The project on **India Stock Market Data Analysis and Visualization** provided valuable insights into the country's economic and financial performance using a structured dataset and advanced visualization tools. By organizing and interpreting data from *India_Stock_Market_Data.xlsx*, the study successfully highlighted the dynamic relationship between India's macroeconomic indicators—namely GDP Growth, Inflation, Exchange Rate (INR/USD), and the BSE SENSEX index. The approach emphasized how fluctuations in these factors collectively shape the nation's financial outlook and investor sentiment.

Through the use of **Power BI** and **Tableau**, the project translated numerical data into intuitive and interactive dashboards. These visualizations effectively represented multi-year trends, making it easier to observe correlations and patterns that are often difficult to identify through raw data. The dashboards demonstrated how strong GDP growth generally coincides with rising stock market values, while high inflation or currency depreciation tends to slow down market performance. The visual approach therefore not only improved interpretability but also showcased the potential of data analytics tools for economic monitoring and research.

Overall, this project demonstrates that visualization-based analytics can serve as a powerful method for understanding complex macroeconomic relationships in emerging markets like India. The integration of GDP, inflation, and currency data into a unified view of market performance supports data-driven decision-making for investors, policy planners, and academic researchers. With further extension to real-time and sectoral data, this framework can evolve into a comprehensive financial intelligence model that continuously tracks and forecasts India's economic health.

Chapter 11: Future Scope

This project provides a foundation for understanding the relationship between India's macroeconomic indicators and stock market performance. However, there remains considerable potential to expand and enhance the analysis in future work. The following points outline possible directions for further development:

1. **Integration of higher-frequency data:**
   The current dataset contains annual data, which limits the ability to analyse short-term market fluctuations. Future studies can include monthly or quarterly data to capture finer trends, identify seasonal patterns, and assess immediate market responses to economic changes.

2. **Inclusion of sectoral indices and broader financial metrics:**
   Expanding the dataset to incorporate sector-specific indices such as **NIFTY IT, NIFTY BANK, and NIFTY PHARMA** will enable comparison of sectoral performance against macroeconomic indicators. Additionally, incorporating metrics like trading volume, market capitalization, and foreign institutional investments (FII) can provide a deeper understanding of market behaviour.

3. **Application of predictive and machine learning models:**
   Advanced analytical models such as Linear Regression, ARIMA, LSTM, or Random Forests can be implemented to forecast future SENSEX movements based on historical data. Machine learning can also help identify hidden patterns and non-linear relationships between macroeconomic factors and market performance.

4. **Comparative and global analysis:**
   The framework can be extended to include data from other emerging economies for cross-country comparisons. Analyzing how countries like China, Brazil, or Indonesia respond to similar macroeconomic conditions would add a global dimension to the study.

Chapter 12: Appendix

10.1 Python Code

```
pip install pandas numpy matplotlib seaborn plotly scikit-learn
```

Import required libraries

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import MinMaxScaler
```

Check data link by loading head

```
# 1️⃣ Load the Dataset (.xlsx)
# ------------------------------
# Replace with your actual Excel filename
df = pd.read_excel("/content/India_Stock_Market_Data.xlsx")

print("✅ Data Loaded Successfully!")
print(df.head())
```

```
✅ Data Loaded Successfully!
   Year  GDP Growth (%)  SENSEX  Inflation (%)  Exch Rate (INR/USD)  \
0  1980            3.37    0.00          11.35                 7.86
1  1981            3.95  148.25          13.00                 8.66
2  1982            3.73  214.56           7.20                 9.46
3  1983            3.60  222.14           9.60                10.10
4  1984            3.16  236.21           8.40                11.36

   SENSEX_Ret (%)  Mod_Infl_Dummy (4-6%)  High_Infl_Dummy (>6%)
0             NaN                      0                      1
1           48.25                      0                      1
2           44.73                      0                      1
3            3.53                      0                      1
4            6.33                      0                      1
```

Data Cleaning

```
# a. Check missing values
print("\n Missing Values:\n", df.isnull().sum())

# Fill missing numeric values with column mean
df.fillna(df.mean(numeric_only=True), inplace=True)

# b. Remove duplicates
duplicates = df.duplicated().sum()
print(f"\n Duplicates Found: {duplicates}")
df.drop_duplicates(inplace=True)

# c. Ensure correct data types
df['Year'] = df['Year'].astype(int)

# d. Clean column names
df.columns = df.columns.str.strip()
```

```
🧹 Missing Values:
 Year                    0
GDP Growth (%)           0
SENSEX                   0
Inflation (%)            0
Exch Rate (INR/USD)      0
SENSEX_Ret (%)           1
Mod_Infl_Dummy (4-6%)    0
High_Infl_Dummy (>6%)    0
dtype: int64

🗑️ Duplicates Found: 0
```

Data Normalization

```
# a. Check missing values
print("\n Missing Values:\n", df.isnull().sum())

# Fill missing numeric values with column mean
df.fillna(df.mean(numeric_only=True), inplace=True)

# b. Remove duplicates
duplicates = df.duplicated().sum()
print(f"\n Duplicates Found: {duplicates}")
df.drop_duplicates(inplace=True)

# c. Ensure correct data types
df['Year'] = df['Year'].astype(int)

# d. Clean column names
df.columns = df.columns.str.strip()
```

Data inspection and analysis

```
print("\n🔍 Dataset Info:")
print(df.info())
print("\n📊 Descriptive Statistics:")
print(df.describe())
#Filter: High Inflation Years (>6%)
high_infl = df[df['Inflation (%)'] > 6]
print(f"\n🔥 Years with Inflation > 6%:\n{high_infl[['Year',
    'Inflation (%)']]}")
#Example Stats: SENSEX
mean_sensex = df['SENSEX'].mean()
median_sensex = df['SENSEX'].median()
mode_sensex = df['SENSEX'].mode()[0]
range_sensex = df['SENSEX'].max() - df['SENSEX'].min()
variance_sensex = df['SENSEX'].var()
std_sensex = df['SENSEX'].std()
print(f"""
📊 SENSEX Summary:
Mean: {mean_sensex:.2f}
Median: {median_sensex:.2f}
Mode: {mode_sensex:.2f}
Range: {range_sensex:.2f}
Variance: {variance_sensex:.2f}
Std Dev: {std_sensex:.2f}
""")
```

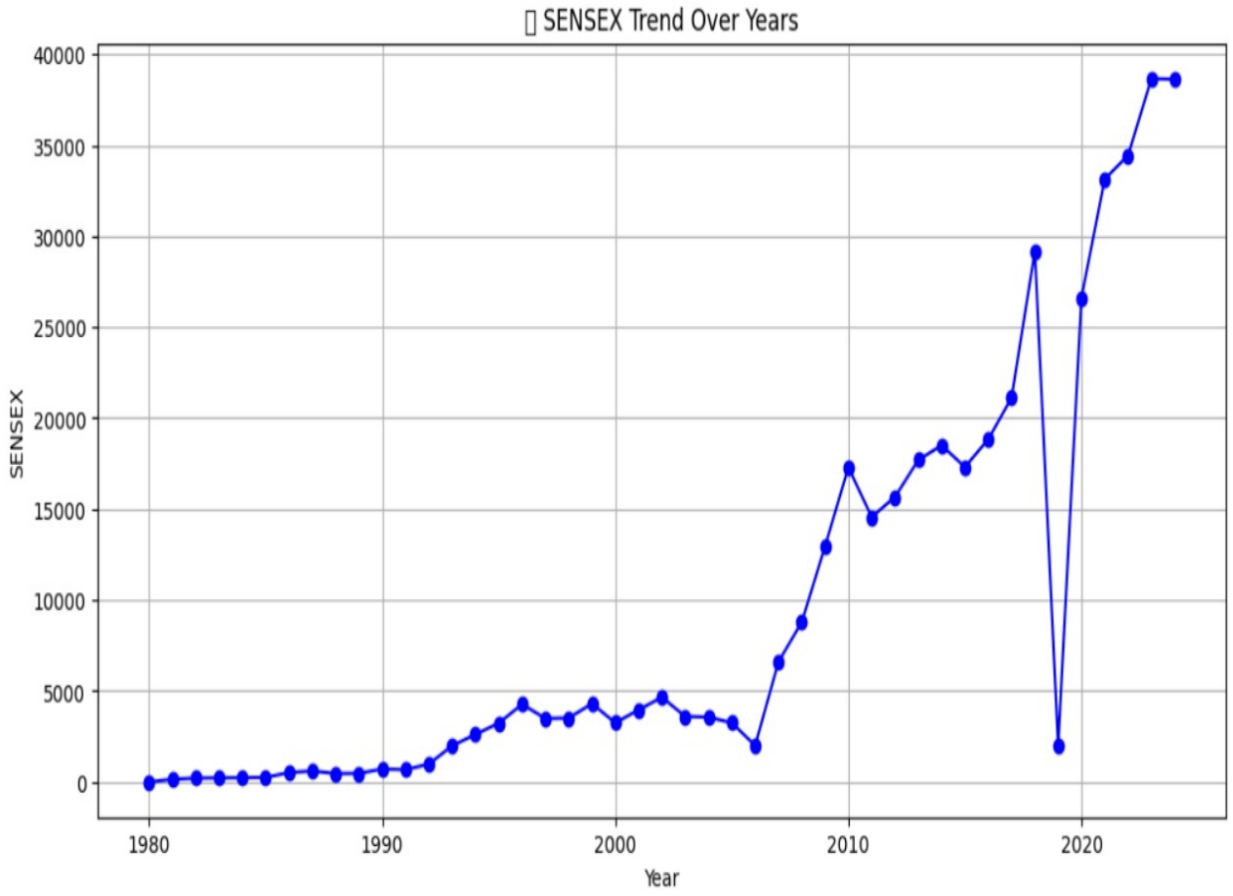
```
🔍 Dataset Info:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 45 entries, 0 to 44
Data columns (total 8 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   Year                    45 non-null     int64
 1   GDP Growth (%)          45 non-null     float64
 2   SENSEX                  45 non-null     float64
 3   Inflation (%)           45 non-null     float64
 4   Exch Rate (INR/USD)     45 non-null     float64
 5   SENSEX_Ret (%)          45 non-null     float64
 6   Mod_Infl_Dummy (4-6%)   45 non-null     int64
 7   High_Infl_Dummy (>6%)   45 non-null     int64
dtypes: float64(5), int64(3)
memory usage: 2.9 KB
None
```

```
📈 Descriptive Statistics:
              Year  GDP Growth (%)        SENSEX  Inflation (%)  \
count    45.000000       45.000000     45.000000      45.000000
mean   2002.000000        7.666889   9450.758000       7.756667
std      13.133926        2.921712  11475.182398       2.711730
min    1980.000000        3.020000      0.000000       3.300000
25%    1991.000000        6.300000    713.600000       5.800000
50%    2002.000000        7.760000   3562.310000       7.900000
75%    2013.000000        9.840000  17291.100000       9.400000
max    2024.000000       12.270000  38667.330000      13.000000

       Exch Rate (INR/USD)  SENSEX_Ret (%)  Mod_Infl_Dummy (4-6%)  \
count            45.000000       45.000000               45.00000
mean             42.979556       45.789318                0.20000
std              23.598036      184.979590                0.40452
min               7.860000      -93.090000                0.00000
25%              24.520000       -0.100000                0.00000
50%              44.100000       12.270000                0.00000
75%              60.580000       33.400000                0.00000
max              83.900000     1220.760000                1.00000

       High_Infl_Dummy (>6%)
count              45.000000
mean                0.711111
std                 0.458368
min                 0.000000
25%                 0.000000
50%                 1.000000
75%                 1.000000
max                 1.000000
```

```
📊 SENSEX Summary:
Mean: 9450.76
Median: 3562.31
Mode: 0.00
Range: 38667.33
Variance: 131679811.06
Std Dev: 11475.18
```

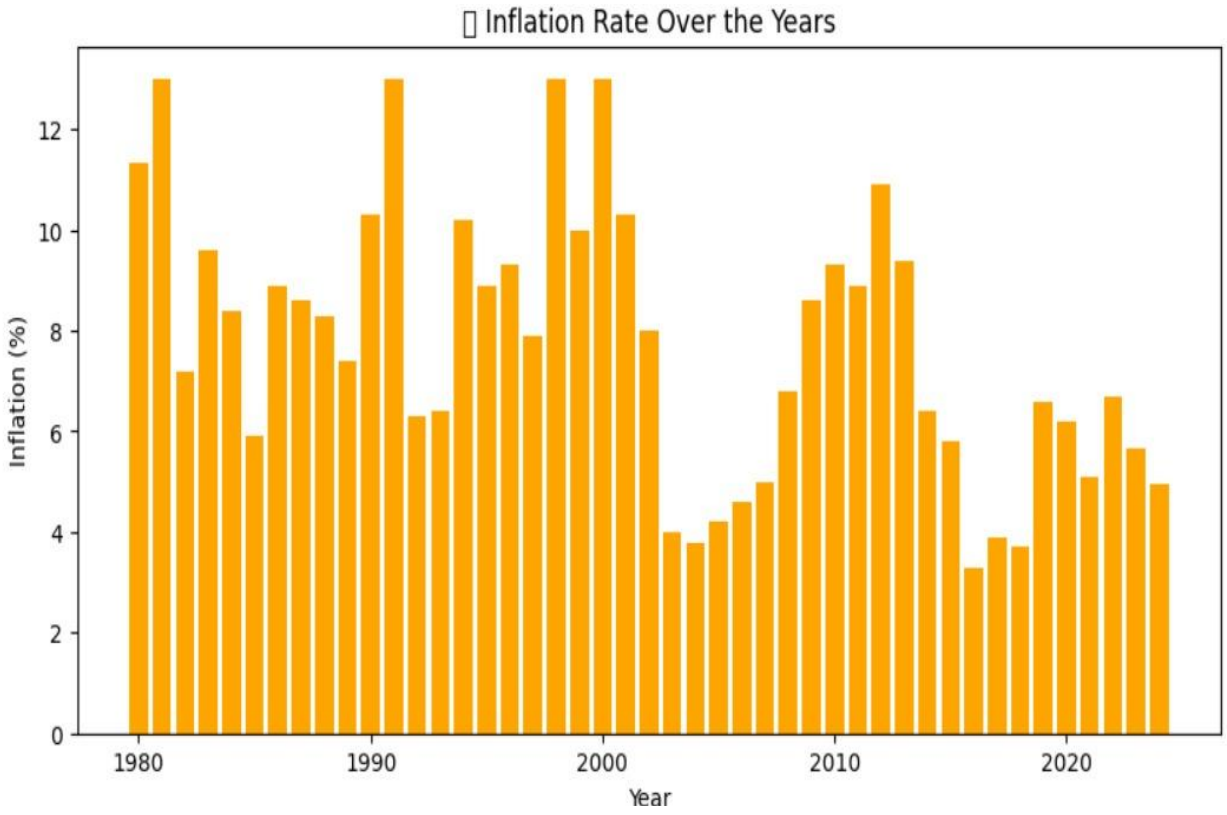
Visualizations –Line Chart

```
plt.figure(figsize=(12, 6))
plt.plot(df['Year'], df['SENSEX'], marker='o', color='blue')
plt.title("SENSEX Trend Over Years")
plt.xlabel("Year")
plt.ylabel("SENSEX")
plt.grid(True)
plt.show()
```



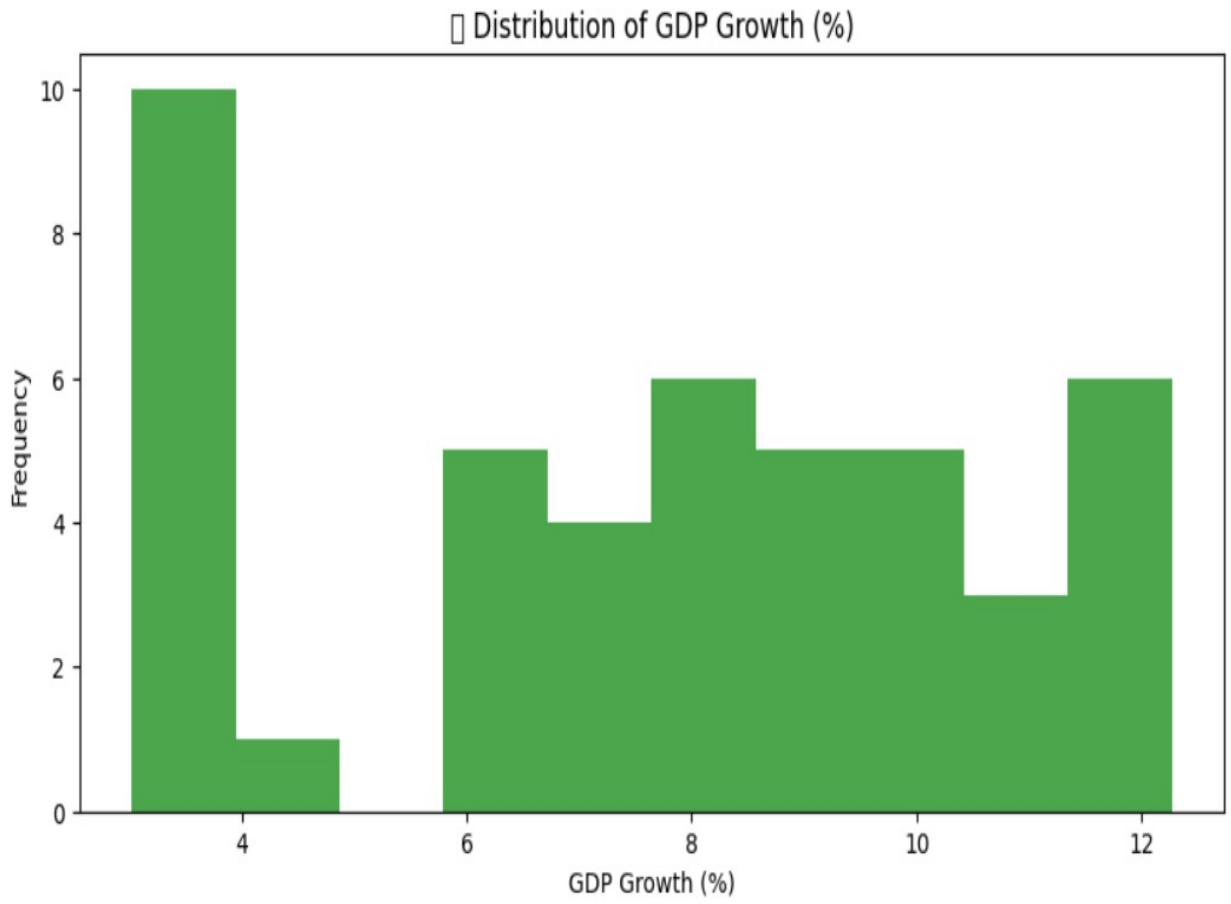
Bar Chart

```
plt.figure(figsize=(10,5))
plt.bar(df['Year'],df['Inflation(%)']
,color='orange')
plt.title("Inflation Rate Over the
Years")
plt.xlabel("Year")
plt.ylabel("Inflation (%)")
plt.show()
```



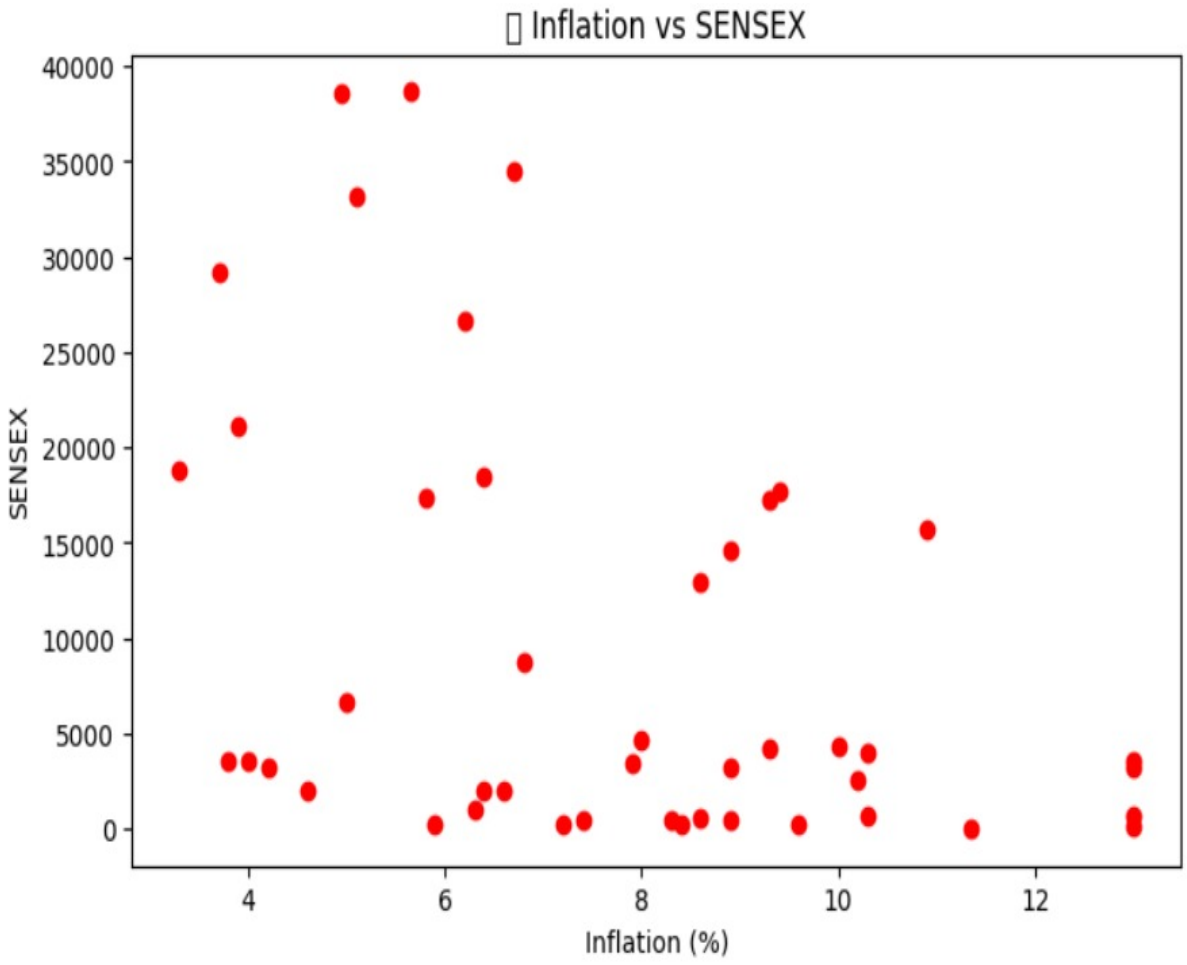
Histogram

```
plt.figure(figsize=(10, 5))

plt.hist(df['GDP Growth (%)'], bins=10, color='green', alpha=0.7)

plt.title("📊Distribution of GDP Growth (%)")

plt.xlabel("GDP Growth (%)")

plt.ylabel("Frequency")
plt.show()
```



Scatter Plot

```
plt.figure(figsize=(8, 5))
plt.scatter(df['Inflation (%)'], df['SENSEX'], color='red')

plt.title("Inflation vs SENSEX")

plt.xlabel("Inflation (%)")
plt.ylabel("SENSEX")
plt.show()
```



Heatmap

```
plt.figure(figsize=(8, 5))

sns.heatmap(df.corr(numeric_only=True), annot=True, cmap='coolwarm',
fmt=".2f")
plt.title("Correlation Heatmap")

plt.show()
```