Stock Data Analysis and Visualization Project

Objective

The objective of this project was to extract historical stock data for multiple companies using the Yahoo Finance API, save the data in Excel format, and create visualizations to analyze stock price movements. This project highlights my skills in Python programming, data analysis, and data visualization.

Technologies and Tools

- **Programming Language**: Python
- **Libraries Used**:
  - `yfinance`: To fetch historical stock data from Yahoo Finance.
  - `pandas`: For data manipulation and storing data in Excel files.
  - `matplotlib`: For creating visualizations of stock price trends.
  - `openpyxl`: For handling Excel file outputs.

Project Workflow

1. Stock Data Retrieval

- Used the `yfinance` library to extract daily historical data for five major companies:
  1. **Apple (AAPL)**
  2. **Microsoft (MSFT)**
  3. **Amazon (AMZN)**
  4. **NVIDIA (NVDA)**
  5. **Tesla (TSLA)**

- Timeframe: February 1, 2025, to February 28, 2025.

2. Excel File Generation

- Leveraged the `pandas` library to store the extracted stock data for each company in separate Excel files.
- Ensured that the files included key columns like Open, High, Low, Close, Adj Close, and Volume.

3. Data Visualization

- Created line charts using the `matplotlib` library to visualize the daily closing prices for each stock within the given timeframe.
- Customized graphs with appropriate titles, axis labels, legends, and gridlines for improved readability.

4. Error Handling and Debugging

- Resolved dependency issues, such as missing the `openpyxl` library, by installing it with `pip install openpyxl`.
- Debugged errors related to incorrect column references by verifying column names with `pandas.DataFrame.columns`.

Code Workflow

Example Code: Data Extraction and Excel Generation

```
import yfinance as yf
import pandas as pd

# Define parameters
stock_symbol = "AAPL"  # Change symbol for other companies
start_date = "2025-02-01"
end_date = "2025-02-28"

# Fetch stock data
stock_data = yf.download(stock_symbol, start=start_date, end=end_date)

# Save to Excel
excel_filename = f"{stock_symbol}_stock_data.xlsx"
stock_data.to_excel(excel_filename)
print(f"{stock_symbol} stock data saved to {excel_filename}")
```

Example Code: Data Visualization

```
import matplotlib.pyplot as plt

# Plot closing prices
plt.plot(stock_data['Close'], color='blue', label=f'{stock_symbol} Closing Price')
plt.title(f'{stock_symbol} Stock Price Movements (Feb 2025)')
plt.xlabel('Date')
plt.ylabel('Price')
plt.legend()
plt.grid()
plt.show()
```

Insights Gained

1. Developed a deeper understanding of **stock market trends** and **financial data visualization**.
2. Strengthened proficiency in Python libraries such as `yfinance`, `pandas`, and `matplotlib`.
3. Gained experience in error handling for dependency management (e.g., `openpyxl` for Excel file handling).
4. Identified key insights, such as stock performance trends for each company during February 2025.

Applications of the Project

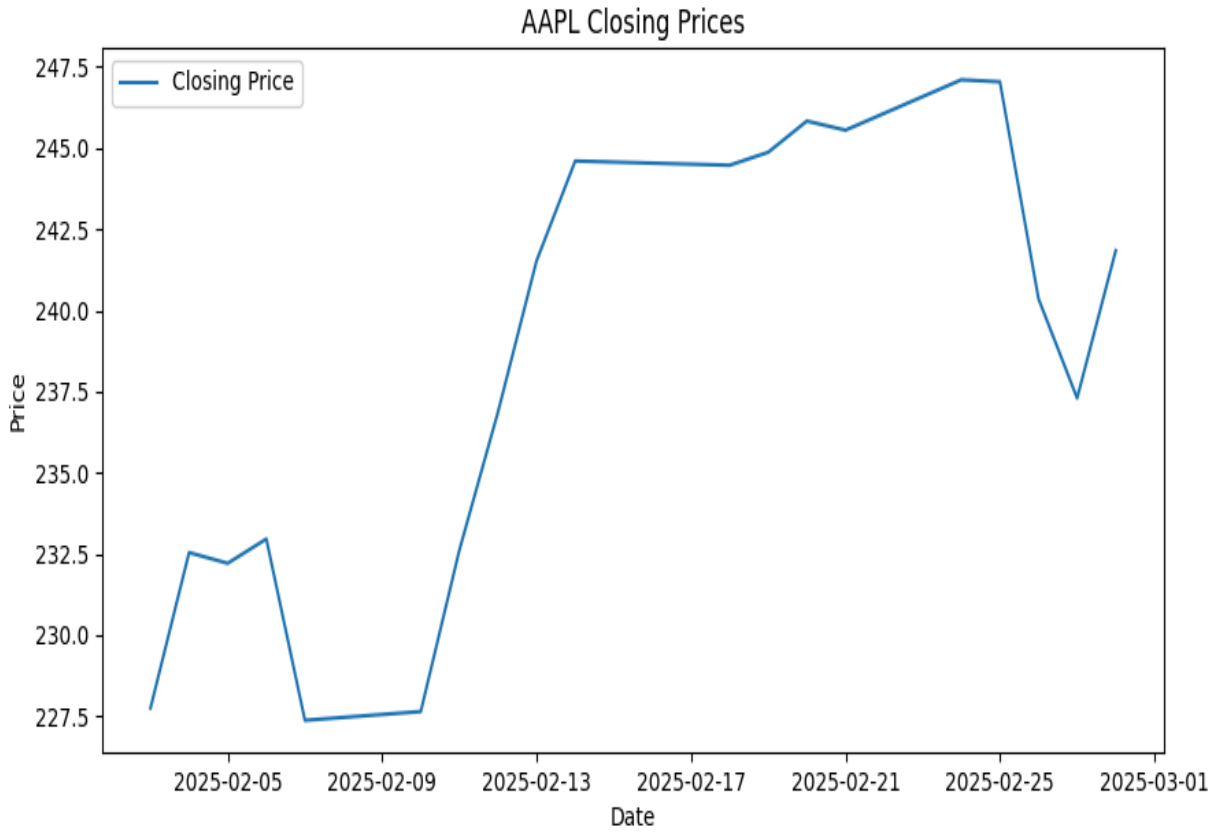
- **Portfolio Analysis**: Helps investors track historical stock performance to make informed decisions.
- **Financial Reporting**: Provides structured insights into stock market trends.
- **Skill Development**: Demonstrates practical knowledge of Python for data analysis and visualization tasks.

Future Improvements

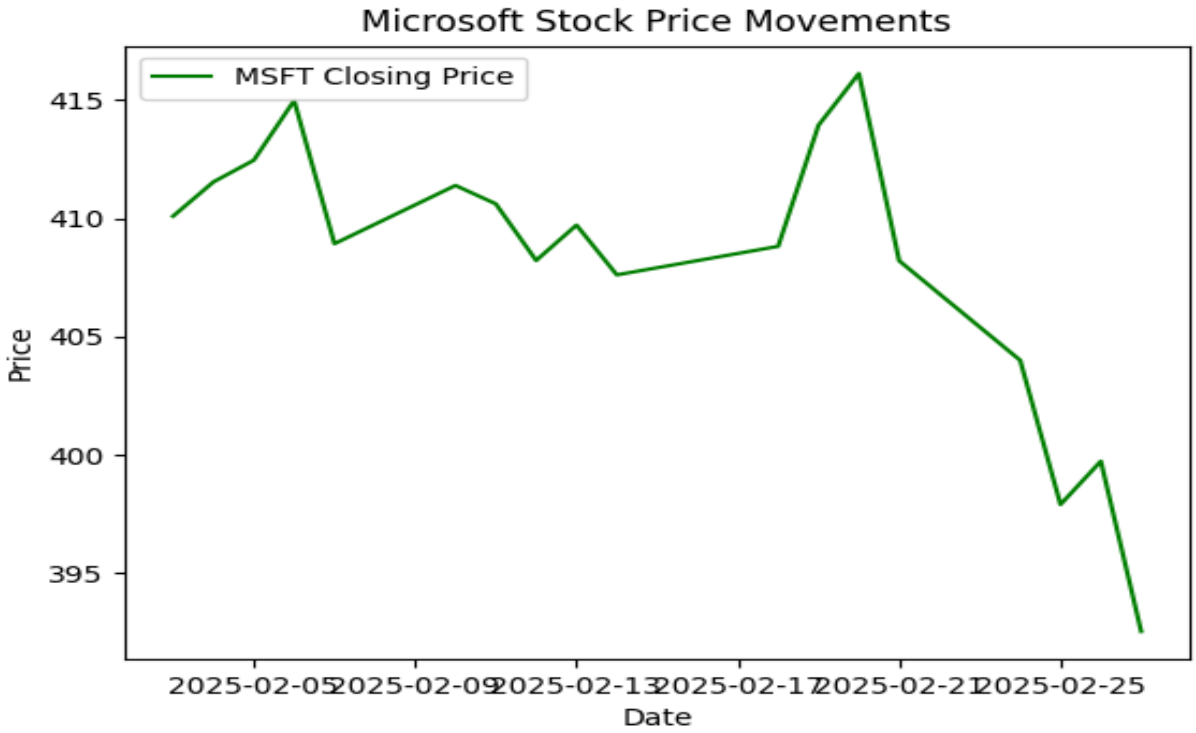
- Incorporate advanced data visualization tools like `seaborn` or `plotly` for richer insights.
- Expand the analysis to include stock correlation and market comparisons.
- Automate the workflow to schedule periodic stock data retrieval and reporting.

Output

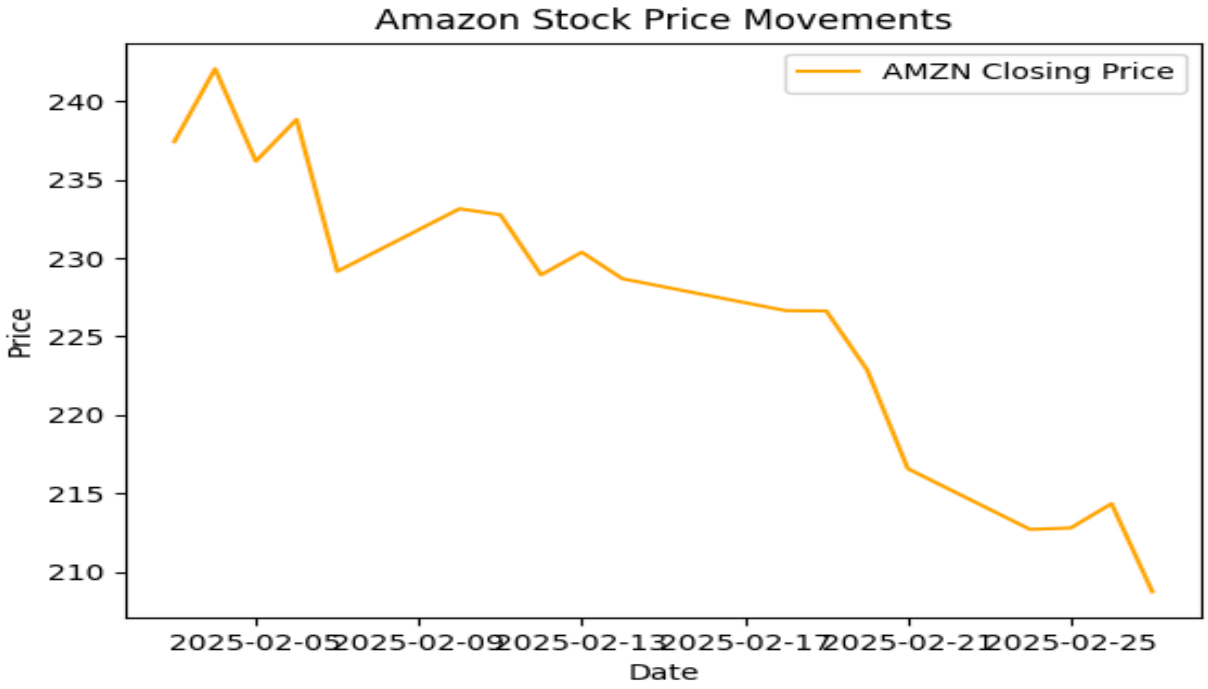
- **Apple (AAPL)**



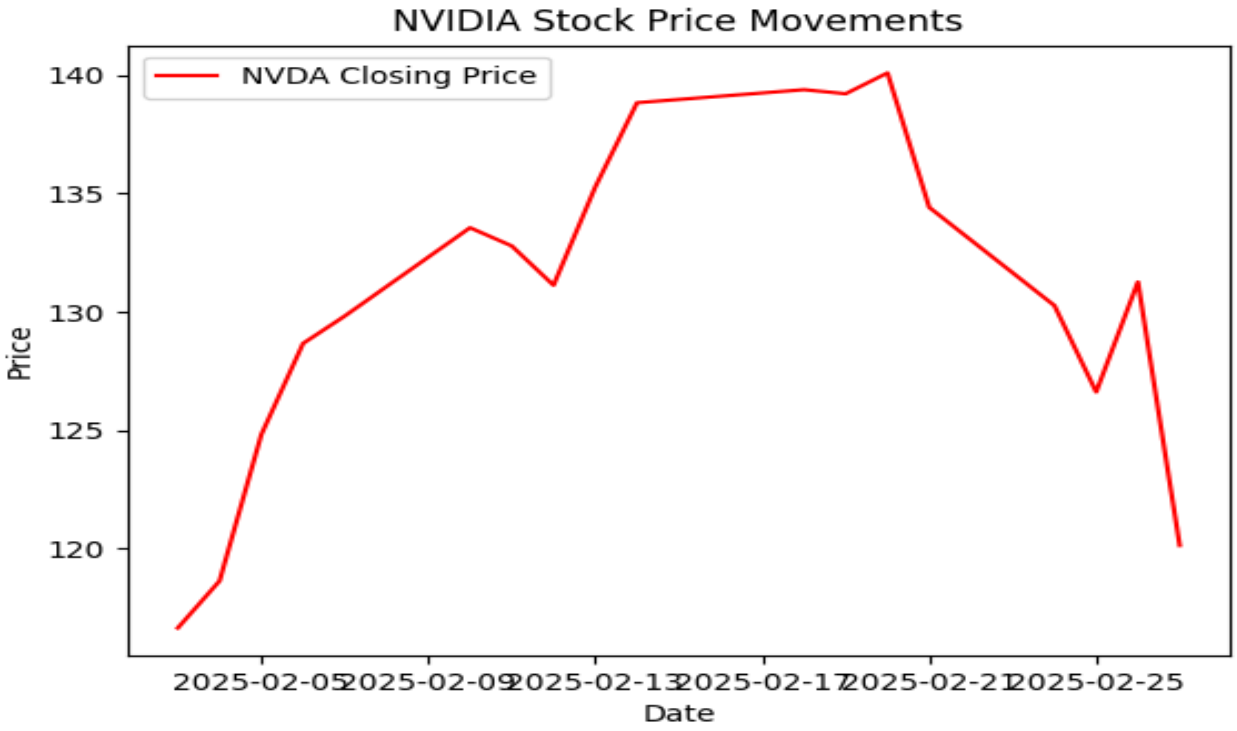
- **Microsoft (MSFT)**



- **Amazon (AMZN)**



- **NVIDIA (NVDA)**



- **Tesla (TSLA)**