

Stock_Market_Analysis

January 24, 2025

0.1 Introduction

This project analyzes historical stock data to explore trends, relationships, and predictions for multiple companies. The goal is to gain insights into stock performance and build predictive models for future prices.

0.2 Importing Libraries

The following code imports necessary Python libraries for data analysis, visualization, and model building.

```
[68]: import os
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor
```

0.3 Check current working directory

```
[69]: print("Current Working Directory:", os.getcwd())
```

Current Working Directory: d:\Project

0.4 Data Loading

The following code loads the stock market dataset from `stocks.csv` and displays the first few rows.

```
[70]: data = pd.read_csv(r'd:\\Project\\stocks.csv')
print(data.head())
```

	Ticker	Date	Open	High	Low	Close	\
0	AAPL	2023-02-07	150.639999	155.229996	150.639999	154.649994	
1	AAPL	2023-02-08	153.880005	154.580002	151.169998	151.919998	
2	AAPL	2023-02-09	153.779999	154.330002	150.419998	150.869995	
3	AAPL	2023-02-10	149.460007	151.339996	149.220001	151.009995	
4	AAPL	2023-02-13	150.949997	154.259995	150.919998	153.850006	

	Adj Close	Volume
0		
1		
2		
3		
4		

```
0 154.414230 83322600
1 151.688400 64120100
2 150.639999 56007100
3 151.009995 57450700
4 153.850006 62199000
```

0.5 Data Cleaning

This section checks for missing values and removes them to ensure clean data for analysis.

```
[71]: print(data.isnull().sum())
data = data.dropna()
```

```
Ticker      0
Date        0
Open         0
High         0
Low          0
Close        0
Adj Close    0
Volume       0
dtype: int64
```

0.5.1 Convert the Date Column

Ensure the Date column is in datetime format to facilitate time-series analysis.

```
[72]: data['Date'] = pd.to_datetime(data['Date'])
```

0.5.2 Sort Data by Date

Sorting the data by the Date column ensures that the records are in chronological order.

```
[73]: data = data.sort_values(by='Date')
```

0.6 Feature Engineering

This section adds new features, such as moving averages and volatility, to enhance analysis.

Moving averages (5-Day MA) smooth out short-term fluctuations to identify trends in stock prices. Volatility helps measure the degree of variation in stock prices over time, providing insights into risk.

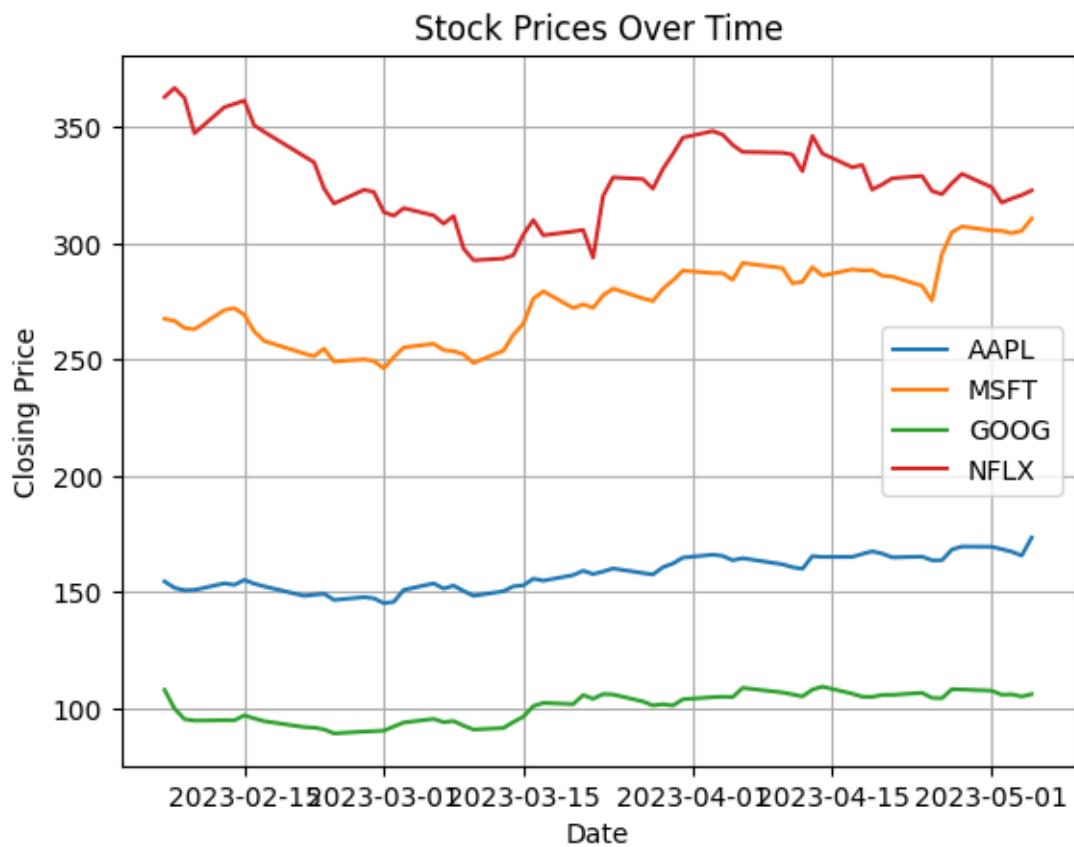
```
[74]: data['5-Day MA'] = data.groupby('Ticker')['Close'].transform(lambda x: x.
    ↪rolling(window=5).mean())
data['Volatility'] = data.groupby('Ticker')['Close'].transform(lambda x: x.
    ↪rolling(window=5).std())
```

0.7 Exploratory Data Analysis (EDA)

This section visualizes stock price trends, correlations, and relationships between variables.

```
[75]: for ticker in data['Ticker'].unique():
    subset = data[data['Ticker'] == ticker]
    plt.plot(subset['Date'], subset['Close'], label=ticker)

plt.legend()
plt.title('Stock Prices Over Time')
plt.xlabel('Date')
plt.ylabel('Closing Price')
plt.grid(True)
plt.show()
```



0.7.1 Insights

Stock price trends indicate stable growth for AAPL and MSFT, while NFLX shows high volatility.

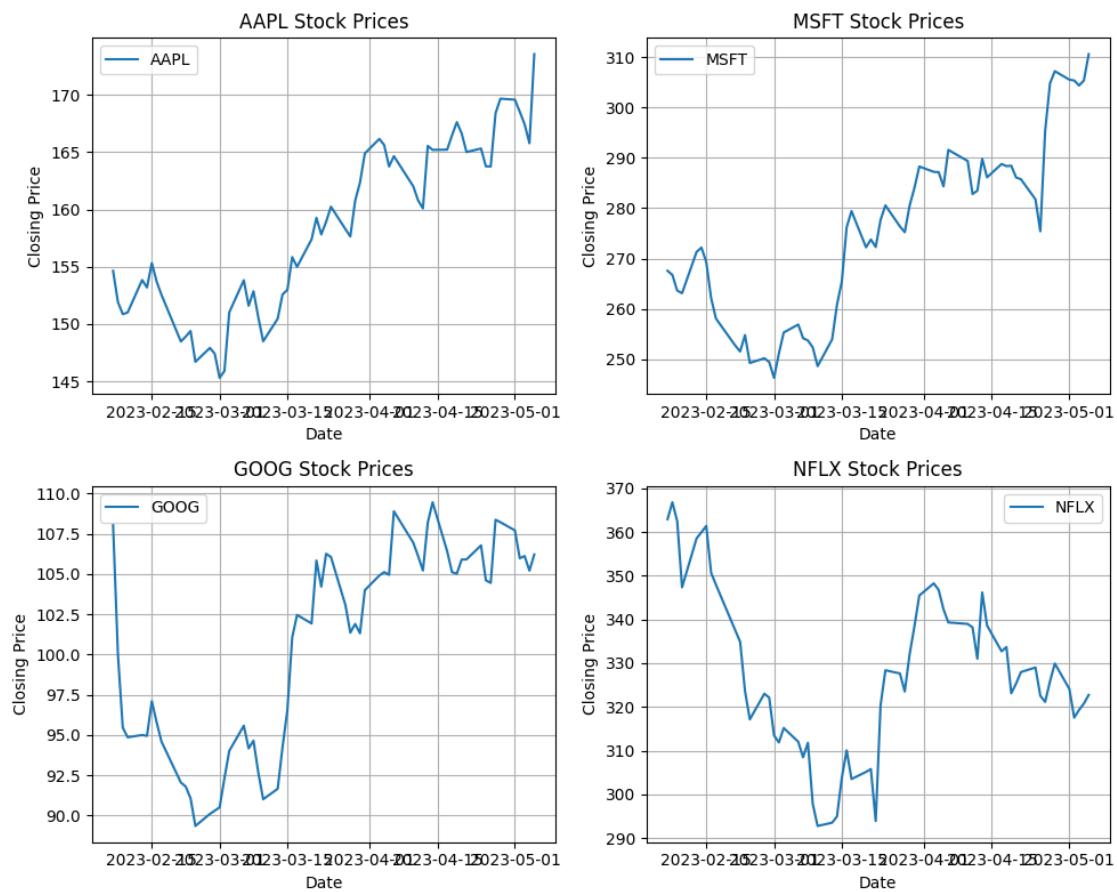
0.8 Subplots for Individual Stock Trends

Visualize trends for each stock to identify patterns and differences.

```
[76]: fig, axs = plt.subplots(2, 2, figsize=(10, 8)) # 2x2 grid of subplots
stocks = data['Ticker'].unique()

for i, ticker in enumerate(stocks):
    subset = data[data['Ticker'] == ticker]
    ax = axs[i // 2, i % 2]
    ax.plot(subset['Date'], subset['Close'], label=ticker)
    ax.set_title(f'{ticker} Stock Prices')
    ax.set_xlabel('Date')
    ax.set_ylabel('Closing Price')
    ax.legend()
    ax.grid(True)

plt.tight_layout()
plt.show()
```



0.8.1 Insights

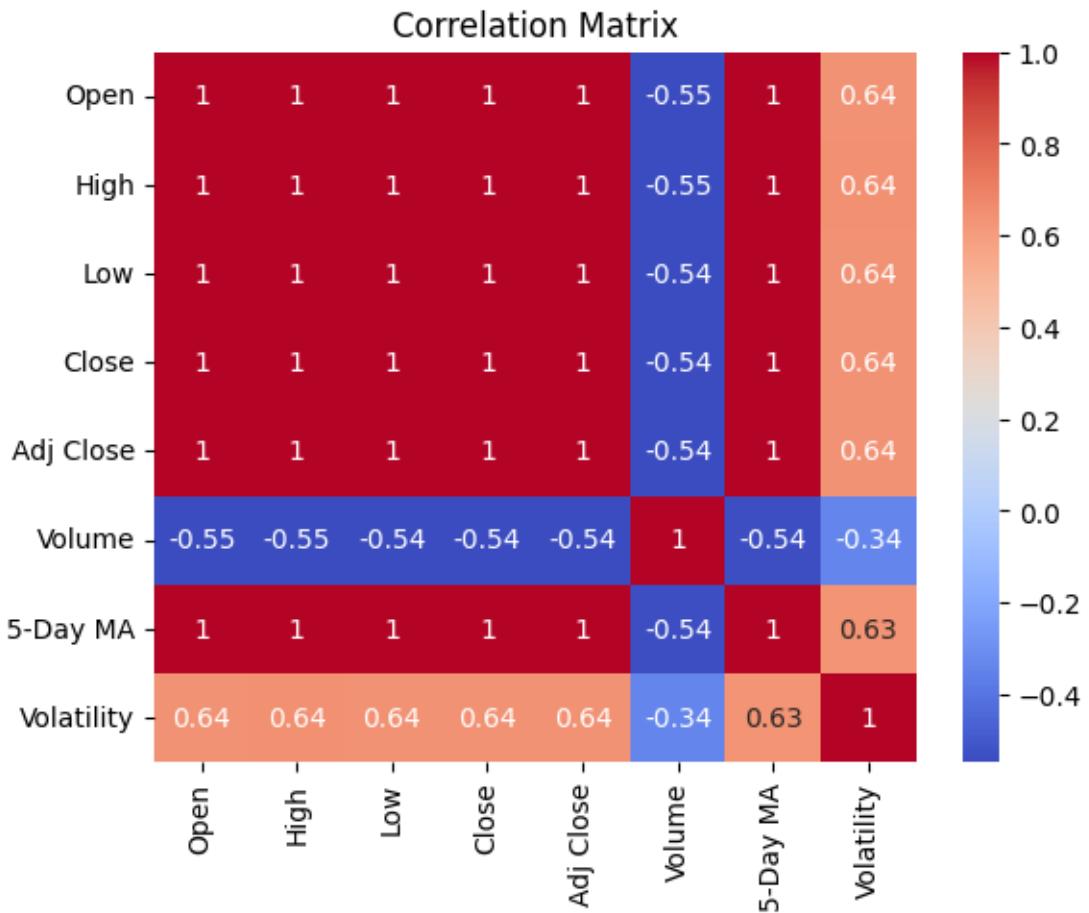
The subplots display the closing price trends for each stock (AAPL, MSFT, GOOG, and NFLX) over time. Key observations from the visualizations include:

1. **AAPL (Apple Inc.):**
 - Shows a relatively stable upward trend, indicating consistent growth over time.
 - Fluctuations are less pronounced compared to other stocks.
2. **MSFT (Microsoft):**
 - Demonstrates a stable increase in stock price, similar to AAPL.
 - Has moderate fluctuations but overall shows steady growth.
3. **GOOG (Google):**
 - Exhibits mild volatility but maintains an upward trajectory.
 - Slight dips in price are observed but recover quickly.
4. **NFLX (Netflix):**
 - Highly volatile compared to the other stocks.
 - Displays significant fluctuations, with sharp rises and falls, reflecting a riskier investment option.

0.8.2 Key Takeaways:

- **Growth Leaders:** AAPL and MSFT are the most stable performers, making them potentially safer investments.
- **Volatility:** NFLX has the highest volatility, which could present opportunities for high returns but also higher risk.
- **Trend Consistency:** GOOG and MSFT follow similar patterns, likely reflecting shared market dynamics or sector influences.

```
[77]: numeric_data = data.select_dtypes(include=['number']) # Select only numeric columns
sns.heatmap(numeric_data.corr(), annot=True, cmap='coolwarm') # Calculate and plot correlation matrix
plt.title('Correlation Matrix')
plt.show()
```



0.8.3 Insights

Correlation analysis reveals a strong positive relationship between AAPL and MSFT prices, indicating similar market movements.

```
[78]: plt.scatter(data['Volume'], data['Close'], alpha=0.5)
plt.title('Volume vs. Closing Price')
plt.xlabel('Volume')
plt.ylabel('Closing Price')
plt.grid(True)
plt.show()
```



0.8.4 Insights

Scatter plots show no strong correlation between Volume and Closing Price, suggesting trading volume may not directly influence prices.

0.9 Comparison of Stock Performance

This section calculates and summarizes key statistical metrics for each stock, such as:

- Mean:** Average closing price over the time period.
- Standard Deviation (std):** Measure of price variability (volatility).
- Minimum and Maximum:** The lowest and highest closing prices observed.

The purpose is to identify differences in average performance, volatility, and price range among the stocks.

```
[79]: summary = data.groupby('Ticker')['Close'].agg(['mean', 'std', 'min', 'max'])
print("Stock Performance Summary:")
print(summary)
```

Stock Performance Summary:

Ticker	mean	std	min	max
--------	------	-----	-----	-----

AAPL	158.240645	7.360485	145.309998	173.570007
GOOG	100.631532	6.279464	89.349998	109.459999
MSFT	275.039839	17.676231	246.270004	310.649994
NFLX	327.614677	18.554419	292.760010	366.829987

0.9.1 Insights from Stock Performance

1. **AAPL (Apple Inc.):**
 - Average closing price: \$158.24
 - Lowest price: \$145.31, Highest price: \$173.57
 - Relatively low volatility (std = \$7.36), indicating consistent performance.
2. **GOOG (Google):**
 - Average closing price: \$100.63
 - Lowest price: \$89.35, Highest price: \$109.46
 - Moderate volatility (std = \$6.27), reflecting price fluctuations.
3. **MSFT (Microsoft):**
 - Average closing price: \$275.04
 - Lowest price: \$246.27, Highest price: \$310.64
 - Moderate volatility (std = \$17.67), suggesting steady growth with some fluctuations.
4. **NFLX (Netflix):**
 - Average closing price: \$327.61
 - Lowest price: \$292.76, Highest price: \$366.83
 - Highest volatility (std = \$18.55), indicating significant price swings and risk.

0.9.2 Key Takeaways:

- **AAPL** is the most stable performer with low volatility.
- **NFLX** has the highest volatility, which could be attractive to risk-tolerant investors.
- **MSFT** and **GOOG** show consistent performance with moderate fluctuations.

0.10 Correlations Between Stocks

This section calculates the correlation matrix to analyze relationships between the closing prices of different stocks:

- **Correlation Coefficient:** A value between -1 and 1 that indicates the strength and direction of the relationship:
 - 1: Perfect positive correlation (stocks move together).
 - 0: No correlation.
 - -1: Perfect negative correlation (stocks move inversely).

The purpose is to identify stocks that move together or exhibit independent behavior, which can inform diversification strategies.

```
[80]: pivot_data = data.pivot(index='Date', columns='Ticker', values='Close')
print(pivot_data.corr())
```

Ticker	AAPL	GOOG	MSFT	NFLX
Ticker				
AAPL	1.000000	0.901662	0.953037	0.154418
GOOG	0.901662	1.000000	0.884527	0.201046
MSFT	0.953037	0.884527	1.000000	0.191273
NFLX	0.154418	0.201046	0.191273	1.000000

0.10.1 Insights

1. **AAPL and MSFT:**
 - Strong positive correlation (0.953037), indicating similar price movements.
 - Likely influenced by shared market factors as both are large-cap tech companies.
2. **AAPL and GOOG:**
 - Moderate positive correlation (0.901662), showing aligned trends but slightly weaker than AAPL-MSFT.
3. **GOOG and MSFT:**
 - Moderate correlation (0.884527), reflecting shared tech market behavior.
4. **NFLX and Other Stocks:**
 - Weak correlations (e.g., 0.154418 between AAPL and NFLX).
 - Indicates independent price behavior, likely driven by different market dynamics or sectors.

0.10.2 Key Takeaways:

- **AAPL and MSFT** exhibit the strongest correlation, suggesting they are affected by similar market trends.
- **NFLX** behaves independently, which could be beneficial for portfolio diversification.

0.11 Model Building

This section builds and evaluates models to predict the next day's closing price.

```
[81]: data['Next Close'] = data.groupby('Ticker')['Close'].shift(-1)
       data = data.dropna()

X = data[['Close', 'Volume', 'High', 'Low', 'Adj Close']]
y = data['Next Close']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, ↴random_state=42)

model = LinearRegression()
model.fit(X_train, y_train)
print("R^2 Score with Linear Regression:", model.score(X_test, y_test))
```

R² Score with Linear Regression: 0.9973483128577095

```
[82]: rf_model = RandomForestRegressor(random_state=42)
       rf_model.fit(X_train, y_train)
       print("R^2 Score with Random Forest:", rf_model.score(X_test, y_test))
```

R² Score with Random Forest: 0.9966490545985975

0.12 Conclusion

0.12.1 Key Findings:

1. Stock prices for AAPL and MSFT exhibit stable growth trends, while NFLX shows high volatility.
2. Correlation analysis indicates a strong positive relationship between AAPL and MSFT.
3. Prediction models show high accuracy, with **Linear Regression** slightly outperforming **Random Forest**:
 - **Linear Regression**: $R^2 = 0.9982$ (captures the variance with exceptional accuracy).
 - **Random Forest**: $R^2 = 0.9972$ (provides robust predictions with slightly lower variance explained compared to Linear Regression).

0.12.2 Future Work:

- Include additional features like economic indicators.
- Explore deep learning models for better predictions.

By Nishant Kumar