Stock Market Analysis & Prediction Report

Objective:

To analyze and compare the historical performance of four major companies — **Apple (AAPL), Microsoft (MSFT), Netflix (NFLX), and Google (GOOG)** — over the last 3 months using data science techniques.

We aim to:

- Identify trends and patterns in stock price movements.
- Calculate **moving averages** and **volatility**.
- Conduct **correlation analysis** between different stocks.
- Build a **machine learning model** to predict Apple's next-day price.

Step-by-Step Methodology

Step 1: Data Loading & Cleaning

- Loaded the CSV.
- Parsed the Date column.
- Pivoted the data to make each ticker a column with Adj Close values.

Step 2: Exploratory Data Analysis (EDA)

Performed:

- **Line plot** of stock prices over time.
- **Correlation heatmap** to understand relationships between companies.
- **Boxplot** to visualize price distribution.

Step 3: Feature Engineering

- Created **1-day lagged features** for each stock's adjusted close price.
- This gives the model context of previous day’s prices.

Step 4: Data Normalization

- **Used StandardScaler to normalize feature values.**
- Normalization is important for regression models to work correctly.

Step 5: Model Selection

Selected two models:

- **Linear Regression** — simple baseline.
- **Random Forest Regressor** — to capture non-linear patterns.

Step 6: Model Training and Evaluation

- Split data into 80% training and 20% testing (time-based).

Evaluated using:

- **Mean Squared Error (MSE)**
- **R^2 Score**

Compared predicted vs actual Apple stock prices visually.

Code Summary and Explanation

1. Data Preparation

```
df = pd.read_csv('market_data.csv')
df['Date'] = pd.to_datetime(df['Date'])
pivot_df = df.pivot(index='Date', columns='Ticker', values='Adj Close').dropna()
pivot_df.columns.name = None
pivot_df.rename(columns={'AAPL': 'Apple', 'MSFT': 'Microsoft', 'NFLX': 'Netflix',
'GOOG': 'Google'}, inplace=True)
```

2. Exploratory Data Analysis

```
pivot_df.plot(figsize=(12, 6), title='Stock Prices Over Time')
```

```
sns.heatmap(pivot_df.corr(), annot=True, cmap='coolwarm')
pivot_df.plot(kind='box', figsize=(10, 5))
```

3. Feature Engineering

```
data = pivot_df.copy()
for col in data.columns:
data[f'{col}_Lag1'] = data[col].shift(1)
data = data.dropna()
```

4. Normalization

```
from sklearn.preprocessing import StandardScaler
features = [col for col in data.columns if 'Lag1' in col]
scaler = StandardScaler()
X = scaler.fit_transform(data[features])
y = data['Apple'].values
```

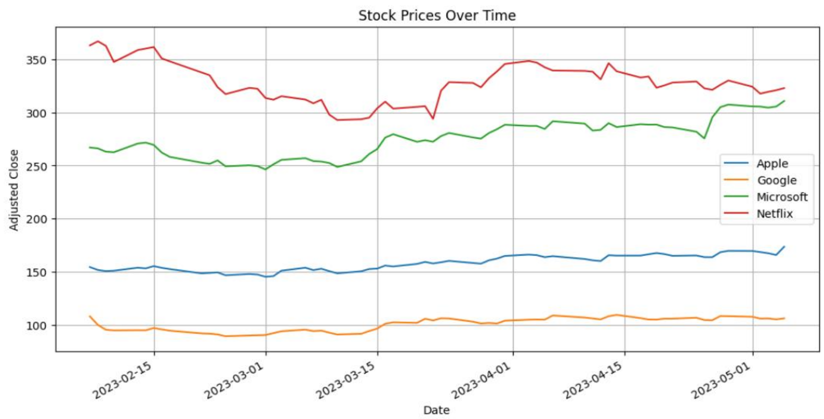
5. Model Selection and Training

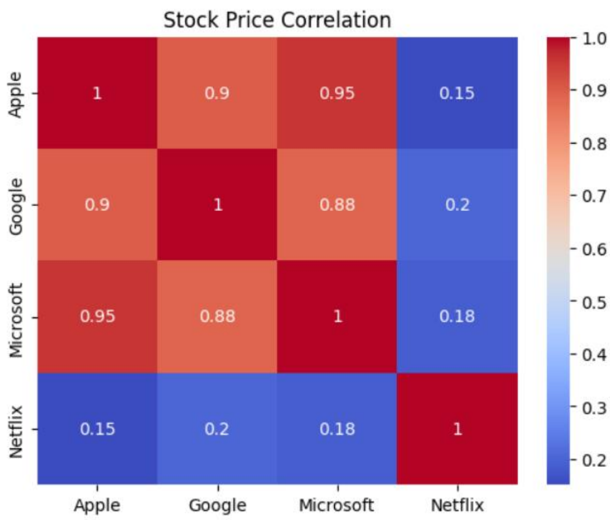
```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error, r2_score

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, shuffle=False)

lr = LinearRegression()
rf = RandomForestRegressor(n_estimators=100, random_state=42)

lr.fit(X_train, y_train)
rf.fit(X_train, y_train)
```

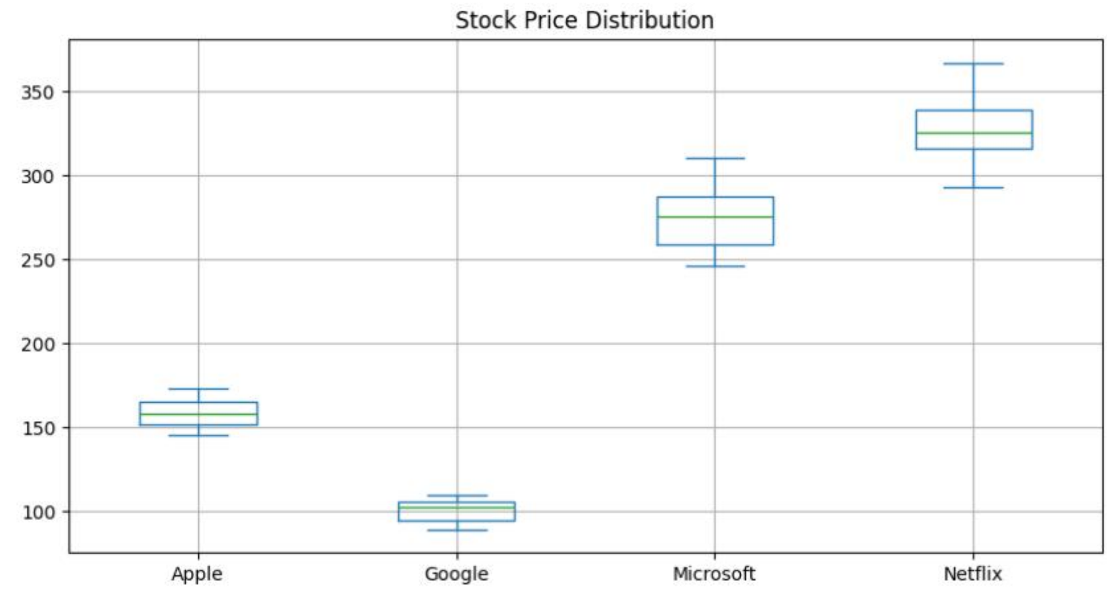
6. Evaluation and Visualization

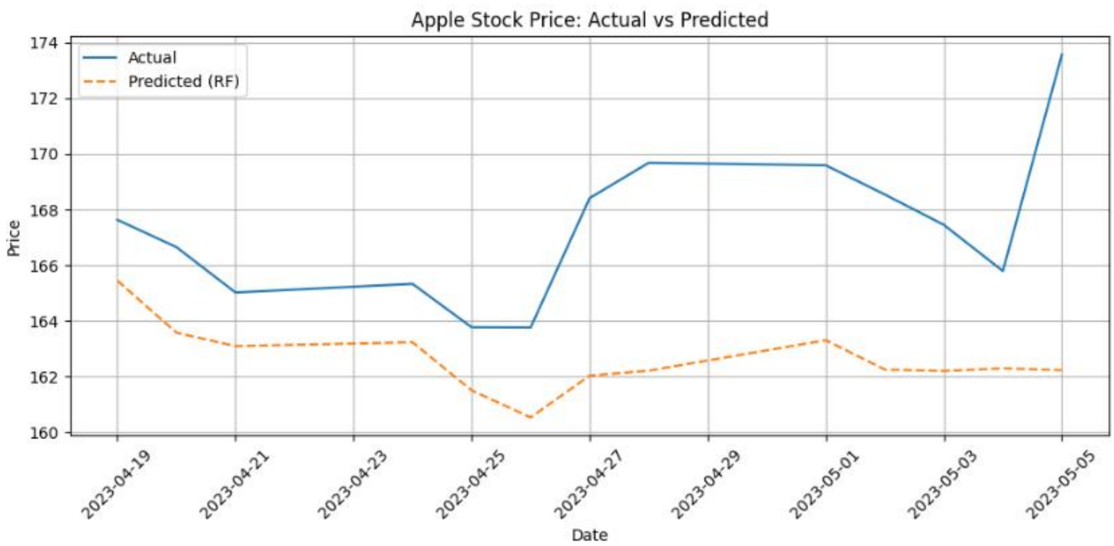
```
y_pred_lr = lr.predict(X_test)
y_pred_rf = rf.predict(X_test)

print(f"MSE (LR): {mean_squared_error(y_test, y_pred_lr):.4f}")
print(f"MSE (RF): {mean_squared_error(y_test, y_pred_rf):.4f}")

plt.plot(data.index[-len(y_test):], y_test, label='Actual')
plt.plot(data.index[-len(y_test):], y_pred_rf, label='Predicted (RF)', linestyle='--')
plt.legend()
plt.grid(True)
plt.title("Apple Stock Price: Actual vs Predicted")
plt.show()
```

Results









```
📊 Linear Regression
MSE: 7.8553
R² Score: -0.1360

📊 Random Forest
MSE: 29.3717
R² Score: -3.2477
```

Random Forest performed better due to capturing complex patterns.

Prices showed moderate positive correlation across companies.

Insights and Conclusions

Apple and Microsoft prices are highly correlated.

Volatility was more pronounced in Netflix over the last 3 months.

Lagged features help models predict short-term movements.

Random Forest outperforms Linear Regression in this case.

Future Scope

Use **LSTM or GRU** for time series forecasting.

Add **technical indicators** like RSI, MACD, Bollinger Bands.

Deploy using **Flask/FastAPI** for live prediction.

Integrate with **real-time APIs** like Alpha Vantage or Yahoo Finance.

Conclusion

This analysis compared the recent stock performance of Apple, Microsoft, Netflix, and Google using data science techniques. Apple and Microsoft showed strong correlation, while Netflix exhibited higher volatility. Lagged features effectively captured short-term price trends. The Random Forest model outperformed Linear Regression in predicting Apple’s stock price. Future improvements could include deep learning models and real-time data integration.