

Stock Market Analysis

Internship Project Report

1. Introduction

This project focuses on analyzing stock market data of major companies to understand price trends, volatility, trading volume, and correlations using data analytics techniques.

2. Objective

- Analyze historical stock prices
- Identify trends using moving averages
- Measure volatility and correlations among stocks

3. Dataset Description

Dataset file name: **stocks.csv**. It contains historical stock data including Date, Company, Open, High, Low, Close, and Volume.

4. Tools Used

Python, Pandas, NumPy, Matplotlib, Seaborn, VS Code, GitHub

5. Data Preprocessing

The data was cleaned by converting date columns to datetime format, sorting records, and checking for missing values.

6. Exploratory Data Analysis

EDA was performed to analyze closing price distribution, volume trends, and company-wise comparisons.

7. Trend & Volatility Analysis

Moving averages were calculated to identify trends, and daily returns were used to measure volatility.

8. Correlation Analysis

Correlation analysis was performed to understand the relationship between different company stocks.

9. Results & Insights

The analysis provided insights into price movements, volatility differences, and strong or weak correlations among stocks.

10. Conclusion

This project demonstrates how stock market data can be analyzed using Python to support data-driven investment insights.

11. Code Appendix

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

df = pd.read_csv("stocks.csv")

df['Date'] = pd.to_datetime(df['Date'])
df = df.sort_values(by='Date')

companies = df['Company'].unique()

plt.figure(figsize=(10,6))
for company in companies:
    company_data = df[df['Company'] == company]
    plt.plot(company_data['Date'], company_data['Close'], label=company)
plt.legend()
plt.show()

df['MA_20'] = df.groupby('Company')['Close'].transform(
    lambda x: x.rolling(window=20).mean()
)

df['Daily_Return'] = df.groupby('Company')['Close'].pct_change()

pivot_df = df.pivot(index='Date', columns='Company', values='Close')
corr_matrix = pivot_df.corr()

sns.heatmap(corr_matrix, annot=True)
plt.show()
```