DSCI-560: Lab 4 Report

Course: Data Science Practicum
Team Members: Jaival Upadhyay, Pratham Solanki, Mayank Patil
Team Name : Guardians of the algorithm(Group 6)

1) Algorithm Development

Methodology

The trading algorithm is designed to make smart buy and sell decisions by combining technical analysis and predictive modeling. It uses three main strategies: Moving Average Crossover, Relative Strength Index (RSI), and Time Series Forecasting (ARIMA and LSTM). The Moving Average Crossover looks at two trends—the short-term 50-day moving average and the long-term 200-day moving average—and generates buy signals when the short-term trend moves above the long-term trend, and sell signals when it falls below. The RSI indicator helps determine if a stock is overbought or oversold, signaling potential price reversals. To make longer-term predictions, the algorithm uses ARIMA (statistical modeling) and LSTM (deep learning), which analyze historical stock price patterns to forecast future trends. By combining these techniques, the algorithm aims to capture both short-term trading opportunities and long-term market trends.

trading algorithms implemented –

- Moving Average Crossover, RSI, ARIMA, and LSTM-based predictions.

How were buy and sell signals generated –

- Buy signals occur when the short-term SMA crosses above the long-term SMA or when RSI < 30 (oversold).
- Sell signals occur when the short-term SMA drops below the long-term SMA or when RSI > 70 (overbought).

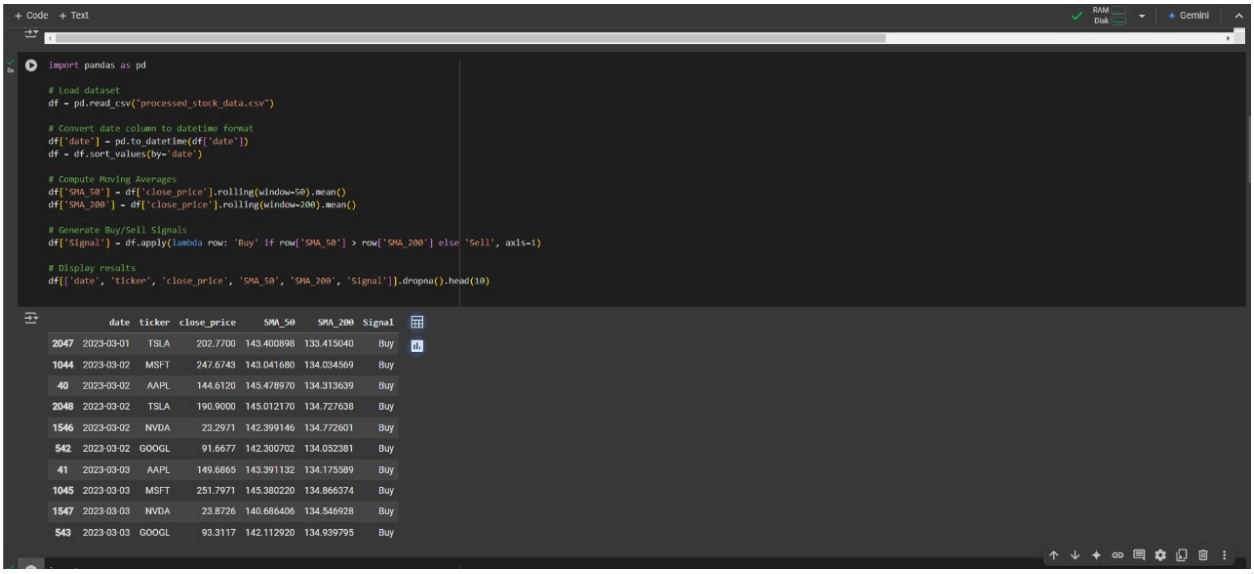
Model Evaluation –

- Total Return, Annualized Return, and Sharpe Ratio were used for evaluating strategy profitability.MAE and RMSE were calculated to measure forecasting accuracy for ARIMA and LSTM.

Code Explanation

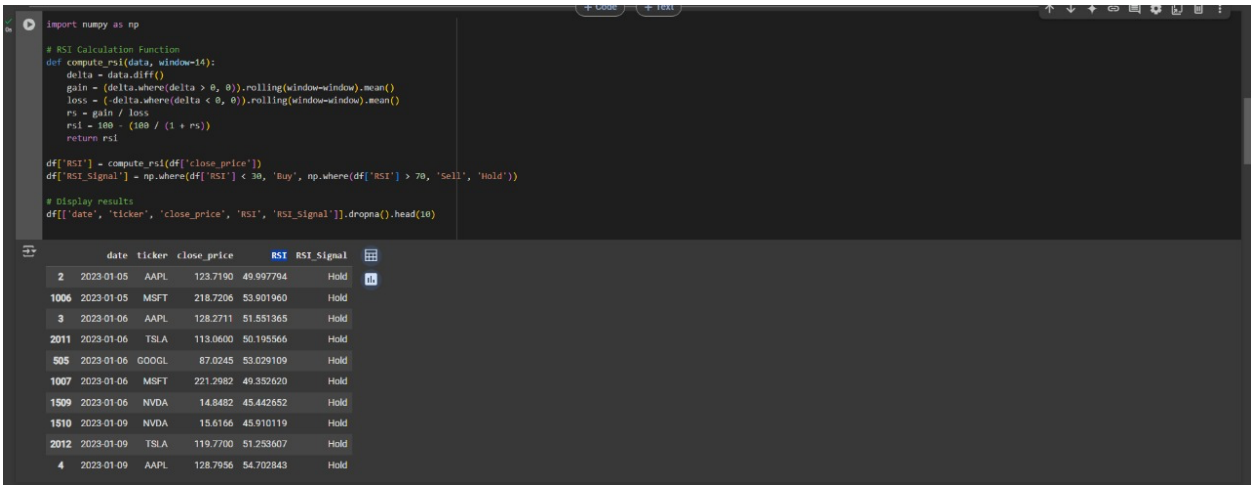
The code starts by loading stock price data from processed_stock_data.csv and ensuring it has the necessary information like date, ticker, and closing price. It then calculates 50-day and 200-day moving averages and assigns buy/sell signals based on when these lines cross. The RSI is computed for 14-day periods, marking stocks as overbought or oversold. For price forecasting, the ARIMA model is trained on 80% of historical stock prices and tested on the remaining 20% to evaluate its accuracy. The LSTM model is built using sequences of past 50-day stock prices to predict future values. The output includes a log of trade decisions, predicted stock prices, and performance metrics, which are visualized through graphs and tables.

Output



```
import pandas as pd

# Load dataset
df = pd.read_csv("processed_stock_data.csv")

# Convert date column to datetime format
df['date'] = pd.to_datetime(df['date'])
df = df.sort_values(by='date')

# Compute Moving Averages
df['SMA_50'] = df['close_price'].rolling(window=50).mean()
df['SMA_200'] = df['close_price'].rolling(window=200).mean()

# Generate Buy/Sell Signals
df['Signal'] = df.apply(lambda row: 'Buy' if row['SMA_50'] > row['SMA_200'] else 'Sell', axis=1)

# Display results
df[['date', 'ticker', 'close_price', 'SMA_50', 'SMA_200', 'Signal']].dropna().head(10)
```

	date	ticker	close_price	SMA_50	SMA_200	Signal
2047	2023-03-01	TSLA	202.7700	143.400898	133.415040	Buy
1044	2023-03-02	MSFT	247.6743	143.041680	134.034569	Buy
40	2023-03-02	AAPL	144.6120	145.478970	134.313639	Buy
2048	2023-03-02	TSLA	190.9000	145.012170	134.727638	Buy
1546	2023-03-02	NVDA	23.2971	142.399146	134.772601	Buy
542	2023-03-02	GOOGL	91.6677	142.300702	134.052381	Buy
41	2023-03-03	AAPL	149.6865	143.391132	134.175589	Buy
1045	2023-03-03	MSFT	251.7971	145.380220	134.866374	Buy
1547	2023-03-03	NVDA	23.8726	140.686406	134.546928	Buy
543	2023-03-03	GOOGL	93.3117	142.112920	134.939795	Buy



```
import numpy as np

# RSI Calculation Function
def compute_rsi(data, window=14):
    delta = data.diff()
    gain = (delta.where(delta > 0, 0)).rolling(window=window).mean()
    loss = (-delta.where(delta < 0, 0)).rolling(window=window).mean()
    rs = gain / loss
    rsi = 100 - (100 / (1 + rs))
    return rsi

df['RSI'] = compute_rsi(df['close_price'])
df['RSI_Signal'] = np.where(df['RSI'] < 30, 'Buy', np.where(df['RSI'] > 70, 'Sell', 'Hold'))

# Display results
df[['date', 'ticker', 'close_price', 'RSI', 'RSI_Signal']].dropna().head(10)
```

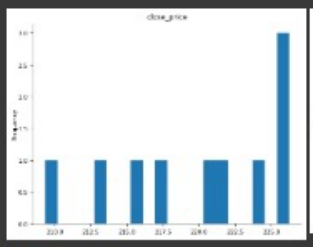
	date	ticker	close_price	RSI	RSI_Signal
2	2023-01-05	AAPL	123.7190	49.997794	Hold
1006	2023-01-05	MSFT	218.7206	53.901960	Hold
3	2023-01-06	AAPL	128.2711	51.551365	Hold
2011	2023-01-06	TSLA	113.0600	50.195566	Hold
505	2023-01-06	GOOGL	87.0245	53.029109	Hold
1007	2023-01-06	MSFT	221.2982	49.352620	Hold
1509	2023-01-06	NVDA	14.8482	45.442652	Hold
1510	2023-01-09	NVDA	15.6166	45.910119	Hold
2012	2023-01-09	TSLA	119.7700	51.253607	Hold
4	2023-01-09	AAPL	128.7956	54.702843	Hold

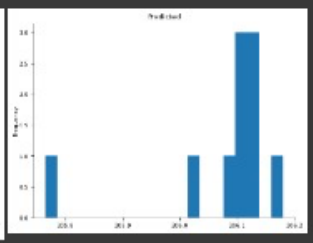
+ Code + Text

0s

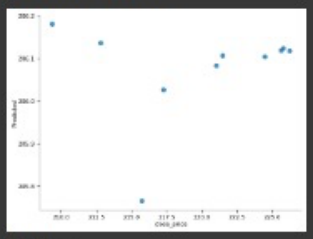
date	close_price	Predicted
2024-08-07	209.3471	206.180370
2024-08-08	212.8292	206.136702
2024-08-09	215.7526	205.765273
2024-08-12	217.2909	206.026395
2024-08-13	221.0268	206.082757
2024-08-14	221.4763	206.106554
2024-08-15	224.4730	206.103836
2024-08-16	225.8016	206.124025
2024-08-19	225.6418	206.118394
2024-08-20	226.2611	206.117345

Distributions

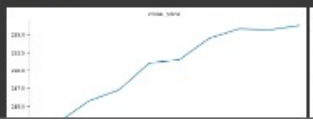


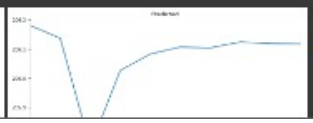


2-d distributions



Values

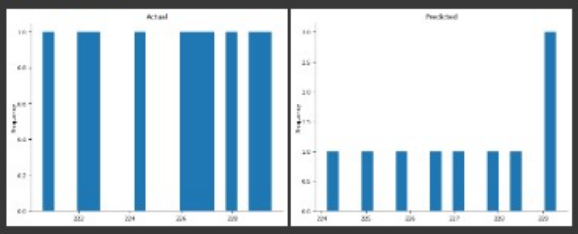




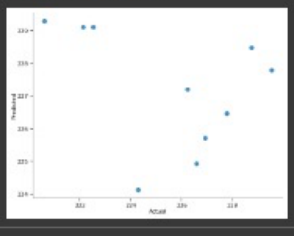
LSTM MAE: 4.10827112197876, RMSE: 5.1092826064969

	Actual	Predicted
0	224.283203	224.125793
1	226.590698	224.920715
2	226.930298	225.711807
3	227.779404	226.461563
4	226.241104	227.199936
5	229.537506	227.785828
6	228.748291	228.477020
7	222.525208	229.115860
8	220.607300	229.292770
9	222.135605	229.111176

Distributions



2-d distributions



2) Mock Trading Environment

The mock trading environment simulates stock trading using historical price data from processed_stock_data.csv. Users start with a fixed initial capital, evenly distributed across selected stocks.

Trading Strategy

The trading strategy is based on a simple moving average (SMA) crossover method to decide when to buy and sell stocks. It tracks two averages: a 5-day SMA (short-term trend) and a 20-day

SMA (long-term trend). When the 5-day SMA moves above the 20-day SMA, it signals that prices are rising, so the system buys the stock. When the 5-day SMA drops below the 20-day SMA, it suggests prices may fall, so the system sells. The goal is to buy when an uptrend starts and sell before a downtrend, aiming to maximize gains while minimizing losses.

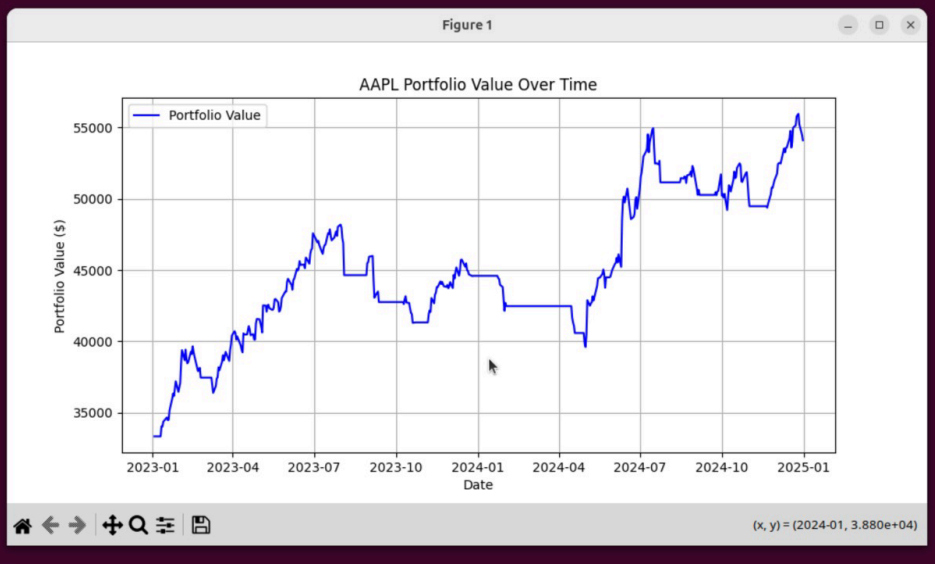
Code explanation

The code simulates a mock trading system using a moving average crossover strategy to analyze stock trends and execute trades. It loads historical stock price data from processed_stock_data.csv, initializes an investment amount, and distributes capital across selected stocks. The strategy calculates 5-day and 20-day SMAs, generating buy signals when the short-term SMA crosses above the long-term SMA and sell signals when it drops below. The system executes trades, updates cash balance, portfolio value, and holdings, and logs transactions. To evaluate performance, it computes total return, annualized return, and Sharpe ratio, along with a portfolio value visualization. The results help assess the effectiveness of the trading strategy.

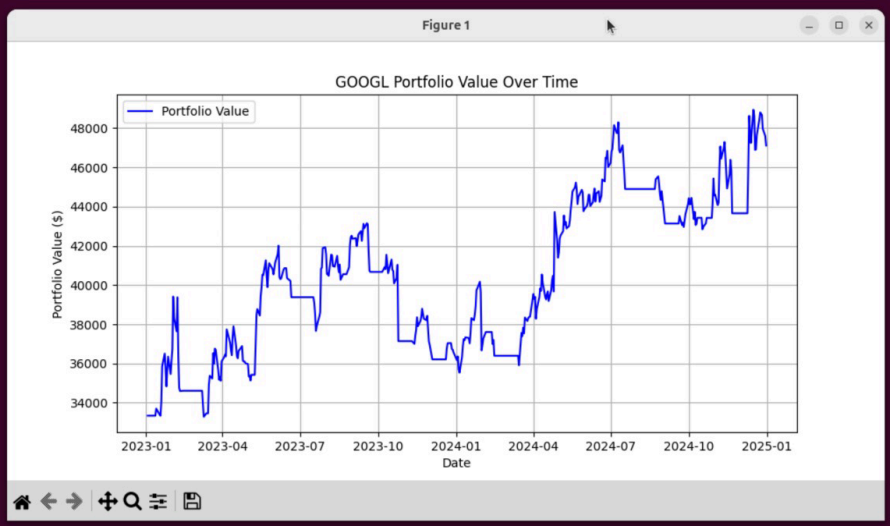
Output:

```
(myenv) jaival@jaival:~/Downloads/dsci-560-main(1)/dsci-560-main/Lab-4$ python3 mock_trading.py
Initial total allocated fund:100000
```

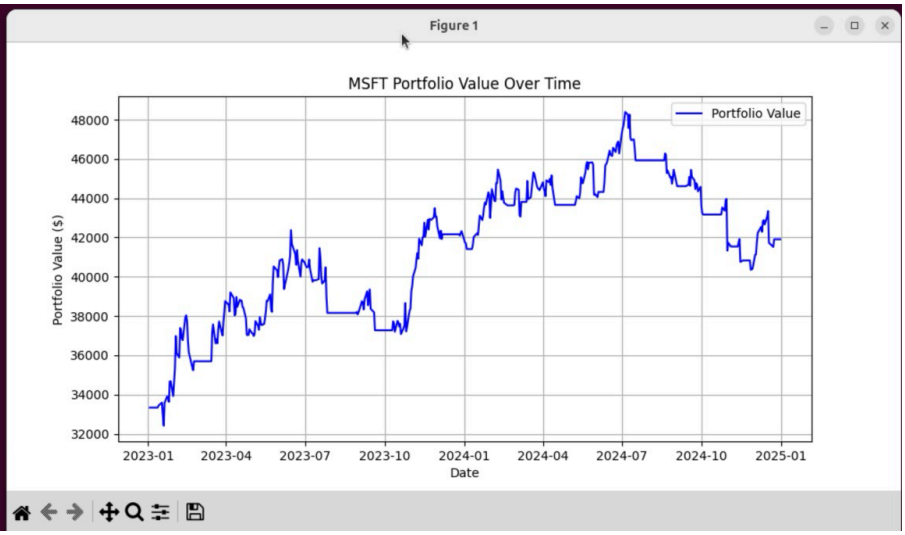
```
--- Simulating for ticker: AAPL ---
Ticker: AAPL
Final Portfolio Value: $54107.94
Total Return: 62.32%
Annualized Return: 27.49%
Sharpe Ratio: 1.81
```



```
--- Simulating for ticker: GOOGL ---
Ticker: GOOGL
Final Portfolio Value: $47112.66
Total Return: 41.34%
Annualized Return: 18.94%
Sharpe Ratio: 0.99
```




```
Annualized Return: 18.94%
Sharpe Ratio: 0.99

--- Simulating for ticker: MSFT ---
Ticker: MSFT
Final Portfolio Value: $41906.21
Total Return: 25.72%
Annualized Return: 12.16%
Sharpe Ratio: 0.88
```




```
--- OVERALL PORTFOLIO SUMMARY ---
Total Portfolio Value: $143126.82
Performance by Ticker:
AAPL: Final Value = $54107.94, Annualized Return = 27.49%, Sharpe Ratio = 1.81
GOOGL: Final Value = $47112.66, Annualized Return = 18.94%, Sharpe Ratio = 0.99
MSFT: Final Value = $41906.21, Annualized Return = 12.16%, Sharpe Ratio = 0.88

Transaction log saved to 'transaction_log.csv'.
(myenv) jaival@jaival:~/Downloads/dsci-560-main(1)/dsci-560-main/Lab-4$
```

- Transaction Logs – “transaction_log.csv”:

	A	B	C	D	E
1	ticker	date	action	shares	price
2	AAPL	2023-01-10	BUY	154	129.3696
3	AAPL	2023-02-24	SELL	154	145.4049
4	AAPL	2023-03-08	BUY	148	151.5101
5	AAPL	2023-08-04	SELL	148	180.6206
6	AAPL	2023-08-29	BUY	146	182.9813
7	AAPL	2023-09-12	SELL	146	175.2097
8	AAPL	2023-10-09	BUY	144	177.8831
9	AAPL	2023-10-23	SELL	144	171.9301
10	AAPL	2023-11-06	BUY	139	178.1216
11	AAPL	2023-12-27	SELL	139	192.2084
12	AAPL	2024-01-23	BUY	137	194.2285
13	AAPL	2024-02-02	SELL	137	184.944
14	AAPL	2024-04-15	BUY	148	172.0674
15	AAPL	2024-04-19	SELL	148	164.4051
16	AAPL	2024-04-29	BUY	140	172.8745
17	AAPL	2024-07-24	SELL	140	218.0474
18	AAPL	2024-08-15	BUY	136	224.473
19	AAPL	2024-09-06	SELL	136	220.5773
20	AAPL	2024-09-23	BUY	133	226.2211
21	AAPL	2024-11-01	SELL	133	222.665
22	AAPL	2024-11-20	BUY	129	229
23	GOOG	2023-01-12	BUY	220	90.8008
24	GOOG	2023-02-13	SELL	220	94.2683
25	GOOG	2023-03-08	BUY	221	93.9096
26	GOOG	2023-03-13	SELL	221	90.7809
27	GOOG	2023-03-15	BUY	209	95.7628
28	GOOG	2023-04-26	SELL	209	103.3354
29	GOOG	2023-04-27	BUY	201	107.2014
30	GOOG	2023-05-05	SELL	201	105.1887
31	GOOG	2023-05-09	BUY	198	106.9623
32	GOOG	2023-06-13	SELL	198	123.3827
33	GOOG	2023-06-15	BUY	196	124.6382
34	GOOG	2023-06-21	SELL	196	120.1146
35	GOOG	2023-07-17	BUY	190	124.1998
36	GOOG	2023-08-21	SELL	190	127.9063
37	GOOG	2023-08-25	BUY	187	129.4109
38	GOOG	2023-09-22	SELL	187	129.7795

3) Team Discussions

To ensure smooth collaboration, the team met daily to discuss the assignment progress and next steps. The Minutes of Meeting (MoM) document is attached separately, detailing our discussions from Wednesday to Saturday.

4) Details

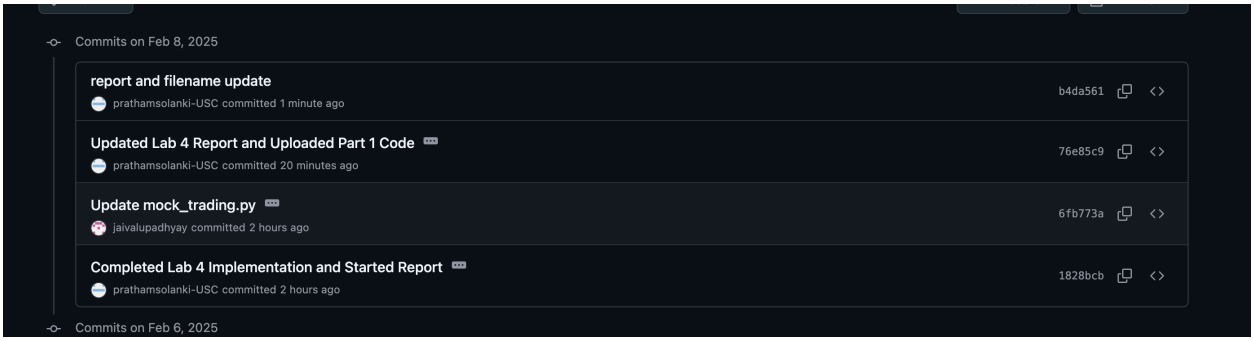
The following files have been submitted as part of Lab 4:

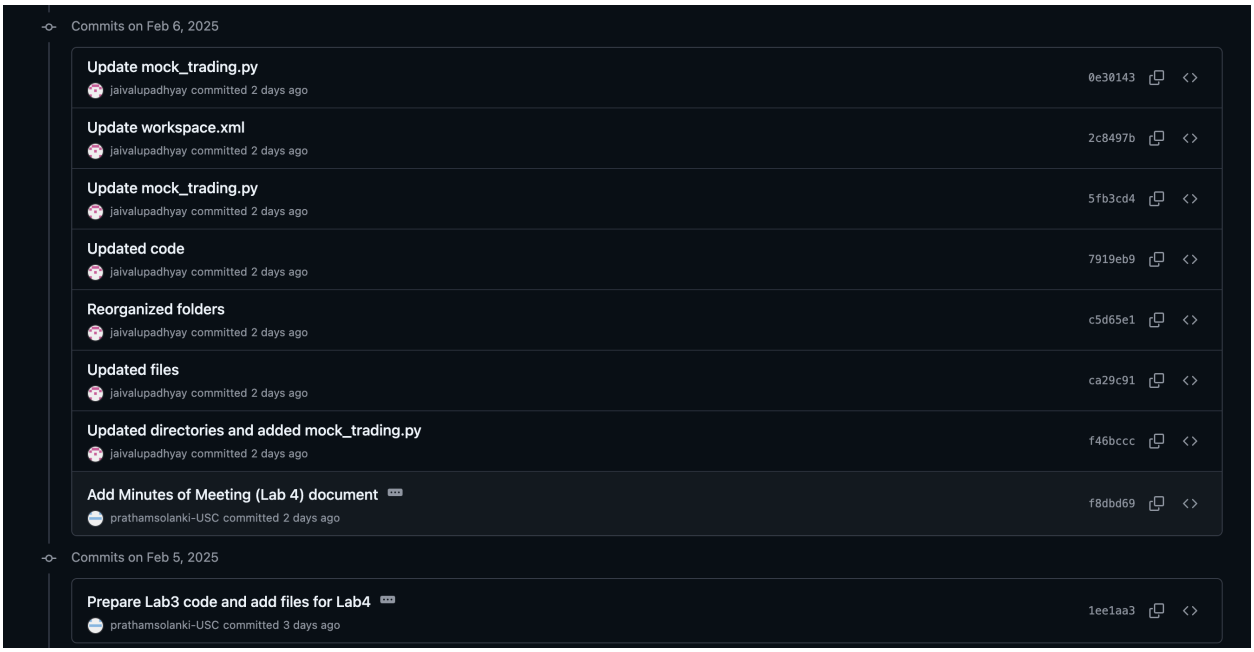
- All code files for the working solution.
- Report_readme documen explaining how to run the code.
- Minutes of Meeting documenting team discussions.
- GitHub history details for version tracking.

5) Conclusion

This lab provided hands-on experience in real-time stock price analysis and algorithmic trading. We successfully implemented and tested a trading algorithm in a simulated environment, evaluated its performance, and participated in a team competition.

6) Github History




Commits on Feb 6, 2025
Update mock_trading.py
jaivalupadhyay committed 2 days ago
0e30143
Update workspace.xml
jaivalupadhyay committed 2 days ago
2c8497b
Update mock_trading.py
jaivalupadhyay committed 2 days ago
5fb3cd4
Updated code
jaivalupadhyay committed 2 days ago
7919eb9
Reorganized folders
jaivalupadhyay committed 2 days ago
c5d65e1
Updated files
jaivalupadhyay committed 2 days ago
ca29c91
Updated directories and added mock_trading.py
jaivalupadhyay committed 2 days ago
f46bccc
Add Minutes of Meeting (Lab 4) document
prathamsolanki-USC committed 2 days ago
f8dbd69
Commits on Feb 5, 2025
Prepare Lab3 code and add files for Lab4
prathamsolanki-USC committed 3 days ago
1ee1aa3