Finance Bull Meeting Note

02/04:
Briefly discuss the content of the lab after the class, and everyone researches the assignment individually.

02/05:
Due to the difficulty of dividing the workload for this lab assignment, we decided that each member would individually choose and implement one algorithm.

Chris: Moving Average.
Ashley: ARIMA.
Johnson: LSTM.

02/06:

Ashley Arima:

I started with the Simple Moving Average (SMA) Strategy for initial implementation. Then I tried an ARIMA model for forecasting, using a walk-forward out-of-sample approach, where the model was trained on past data and used to predict the next day's price. To leverage both method's advantage, I combined SMA and ARIMA into a hybrid strategy, generating buy/sell signals only when both methods aligned. This approach provided the most stable and effective trading signals, reducing false positives and improving overall profitability. I allocated $10,000 per stock instead of $10,000 per portfolio because I wanted to make it easier to compare performance across different stocks using the same strategy, ensuring that each stock had an equal starting point for evaluation. This helped isolate the effectiveness of the trading approach without bias from portfolio allocation differences.

I tried using auto_arima to find the optimal ARIMA parameters. But it took far too long to run, as it iteratively searches for the best (p, d, q) values by evaluating multiple models. Despite this extensive computation, the improvement in return (<1%) and Sharpe ratio (<0.05) was negligible, making the process not worth the added computational cost. Instead, we proceed with the default parameters (5,1,0).

In addition, I incorporated risk management techniques into the trading simulation. Specifically, I implemented stop-loss and take-profit thresholds to manage downside risk and lock in profits. If the stock price dropped 5% below the entry price, the position was exited to prevent further losses. Conversely, if the price rose 10% above the entry price, the system automatically took profits.

02/07:

Chris:

Our simple strategy involves deciding whether to buy or sell a stock based on moving averages. I utilize three types of moving averages: the Simple Moving Average (SMA), the Exponential Moving Average (EMA), and the Weighted Moving Average (WMA). Users can choose a strategy that relies on only one moving average or a combination of two or three. The results from simulated trading can vary significantly depending on which moving averages are used.

Additionally, I take into account profit-taking, stop-loss measures, and commission rates to make these small trading simulations more reflective of real-world scenarios. The strategy is straightforward: it uses the comparison between short-term and long-term moving averages to determine when to buy or sell.

In this program, users start with a customizable initial fund of $10000. Based on the selected portfolio, the program automatically retrieves the stock data from the database and begins making buy or sell decisions from the first date in the portfolio until the end date. After the simulation, it generates evaluation metrics for each stock as well as the overall portfolio.

02/08:

We also implement a stock trading strategy using a Long Short-Term Memory (LSTM) model trained on historical stock price data. The strategy consists of data processing, model training, and performance evaluation through mock trading.

Data Processing

We extract historical stock prices from a MySQL database, retrieving date, open, high, low, close prices, and volume data, sorted chronologically. To enhance predictive power, we compute technical indicators such as Simple Moving Averages (SMA), Exponential Moving Averages (EMA), Relative Strength Index (RSI), Bollinger Bands (BB), and MACD. The target variable (label) is set to 1 if the stock price increases after 5 days, otherwise 0. Data is normalized using MinMaxScaler, and 60-day lookback windows are created for time-series learning.

Model Training

The LSTM model consists of two LSTM layers (50 hidden units each), followed by a fully connected layer with sigmoid activation for binary classification. It is trained using the Adam optimizer with Binary Cross-Entropy Loss (BCE), using a batch size of 32 and running for 20 epochs.

Trading Strategy & Risk Management

Predictions are used to simulate a mock trading strategy:

- Buy if LSTM predicts an increase (1), investing 20% of available capital.

- Sell if LSTM predicts a decrease (0), exiting all holdings.
- Stop-loss at 5% to minimize losses.
- Take-profit at 10% to lock in gains.
- Minimum holding period (3 days) to prevent excessive trading.

Results & Visualization

The strategy’s performance is evaluated using key metrics like Sharpe Ratio, Cumulative Return, Max Drawdown, and Win Rate. A portfolio performance graph is plotted, comparing portfolio value vs. stock price movements over time.