AI Based Stock Market Predictions

Yoshihiro Shirai, Xu Zhuang, Yiting Zhang

Abstract

The purpose of this project is to test the capability of different machine learning methods to predict the mid-price closing (COB) $\mathbf{S}_t = (S_t^1, ..., S_t^d)$ of d stocks, given the previous n days of COB prices $\mathbf{S}_{t-n}, ..., \mathbf{S}_{t-1}$ and other variables, including Treasury rates.

The construction of reliable stock market predictions is useful for detecting interdependencies within different assets and ultimately exploiting them for the construction of quantitative trading strategies.

In this project, we choose $d = 3$ and $n = 10$. Our input data are of the form: $\{X_{t-h}\}_{h=1,...,10}$, where $X_{t-h} = \log\left(\frac{S_{t-h}}{S_{t-h-1}}\right)$, $h = 1, ..., 10$.

Source of data

Our data source is the Wharton Research Dataset. We plan to build a Python file that contains various SQL queries to automatically download the data to our Github repo. This is a data set often used in the academic literature and contains reliable stock market quotes, volume traded, and other information.

Project plan and goal

We will focus on the 10 sector ETFs and the SPY fund tracking the S&P500 index. The current plan is to first split the dataset into training, validation, and testing sets. We will then train some basic models from regression analysis, such as Linear Regression and the autoregressive integrated moving average (ARIMA) model, as well as basic recurrent neural network architectures, and also some more advanced ones, such as the long-short-term memory (LSTM) architecture.

Hyperparameter tuning will be performed for each model in the validation set. The root mean square error in the predictions will be estimated in the testing set for each model. The final goal is to understand which model achieves minimal RMSE in the validation test, which will be reported in our

delivery.

Methods descriptions

- **Linear regression** predicts a continuous output by modeling a linear relationship between the dependent and independent variables. It is a simple, effective method that is often used to predict stock prices based on historical data.
- **ARIMA** is a time series model designed to capture both autoregressive (AR) and moving average (MA) patterns in data, while differencing (I) is used to make the series stationary. It is effective for capturing trends and dependencies in financial time series.
- **RNNs** are a type of neural network designed to process sequential data, such as time series or text, maintaining a "memory" of previous inputs through hidden states. They excel at capturing temporal dependencies but can struggle with long sequences due to vanishing or exploding gradients. RNN with Dropout (RNN-DO) introduces dropout, a regularization technique, to prevent overfitting by randomly deactivating neurons during training. This helps the model generalize better, especially when working with limited data, while preserving the temporal structure of the sequence.
- **RNN Dense** combines the sequential processing capabilities of RNNs with fully connected (dense) layers to produce fixed-size outputs, such as classifications or predictions, from sequential data. The RNN captures temporal dependencies, and the dense layer transforms the processed information into the desired output format. This architecture is commonly used in tasks like time series forecasting or text classification, where both sequence analysis and final decision making are required.
- **LSTM** is a specialized RNN designed to address the problem of vanishing gradients by using gates to control the flow of information, allowing it to capture short and long-term dependencies in sequential data. LSTM excels in tasks with complex or long sequences, such as speech recognition or time series forecasting, where it maintains information over extended periods more effectively than standard RNNs.

Conclusion

By comparing MSEs and signals for different tickers(SPY, IBM, AAPL) using different models. We have the following conclusions.

- Linear regression provides reasonable predictions, but performance varies between different stocks. Further refinement of the model or more complex algorithms may improve accuracy.
- ARIMA
  - **AAPL**: Linear regression achieves a lower validation MSE compared to ARIMA, with similar signal values.
  - **IBM:** Both models have comparable validation MSE, but ARIMA shows a slightly higher signal.
  - **SPY:** Linear Regression outperforms ARIMA with a lower validation MSE; Signal values are identical.
- SimpleRNN
  - **AAPL:** Simple RNN achieves Validation MSE similar to LR but with a higher Signal (0.684 vs. 0.604), indicating better correlation with actual prices. It also outperforms ARIMA in both metrics.
  - **IBM and SPY:** Simple RNN performs compared to LR and ARIMA, as it cannot explain the big showing higher Test MSE and negligible signal, suggesting that it did not capture the underlying patterns effectively for these stocks.
- LSTM
  - **AAPL:** LSTM outperforms Linear Regression and ARIMA with lower Test MSE (0.0017) and higher Signal (0.760), indicating improved predictions.
  - **IBM and SPY:** LSTM shows a similar MSE test to linear regression, but with a lower signal, suggesting marginal improvement.
  - **Signal**: Despite its superior performance, LSTM struggles to fully capture the volatility in logarithmic price changes, particularly during significant market events like the COVID-19 pandemic in 2020.

To further improve the model, we may consider Other metrics instead of MSE since MSE does not adequately reflect the model's ability to capture market volatility and dynamics.