**EECS6895 Milestone 3 Report: A-share Stock Auto Trader**

**Topic:** B9: Investment Strategy - AI Trader (CN/HK/TW/JP)

**Team:** Yiwen Fang (yf2560) | Guoshiwen Han (gh2567)

**Contents**

1. Model Training Refinement

1.1 Data Process

1.2 Model Training

2. Experiments and Evaluations

2.1 SVM Prediction Result

2.2 Accuracy and Comparison with LSTM

3. Web Application

4. Challenges

4.1 Solved in Milestone 3

4.2 Solved in Final Project

4. Final Project Plan

5. Initial Plan and Achievement

6. Final Project Plan

7. References

**1. Model Training Refinement**

**1.1 Data Process**

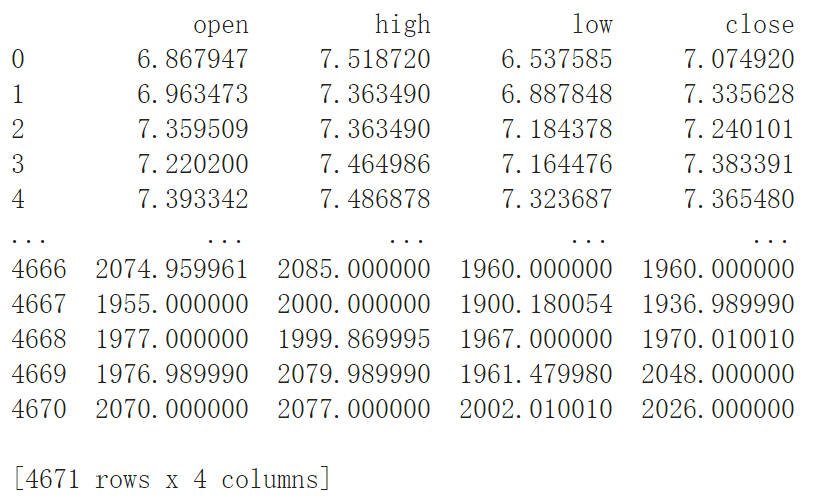
In Milestone 2, we used LSTM **[1, 2]** model to predict the prices of a stock based on the absolute values of prices. This time we change all the data; now, the model is not based on the absolute values of prices but on the relation between the prices. The transformation code is as follows. (**Code 1**)

value = pd.Series(Data['close'].shift(-n) - Data['close'],  
 index=Data.index) # after n days, rise or fall  
Data['High-Low'] = Data['high'] - Data['low'] # Difference between High and Low  
Data['Open-NClose'] = Data['open'] - Data['close'].shift(  
 n) # today's open - close of n days before  
Data['Close-NClose'] = Data['close'] - Data['close'].shift(  
 n) # Today is rise or fall comparing with n days before  
Data['Close-Open'] = Data['close'] - Data['open'] # today's Close – Open  
Data['High-Close'] = Data['high'] - Data['close'] # today's High – Close  
Data['Close-Low'] = Data['close'] - Data['low'] # today's Close – Low  
value[value > 0] = 1 # 1 means rise  
value[value == 0] = 0 # 0 means it doesn't rise or fall  
value[value < 0] = -1 # -1 means fall  
Data = Data.dropna(how='any')  
del (Data['open'])  
del (Data['close'])  
del (Data['high'])  
del (Data['low'])  
print(Data)  
# print(type(Data))  
Data['Value'] = value

**Code 1.** Converting the absolute values of prices to the relation between the prices.

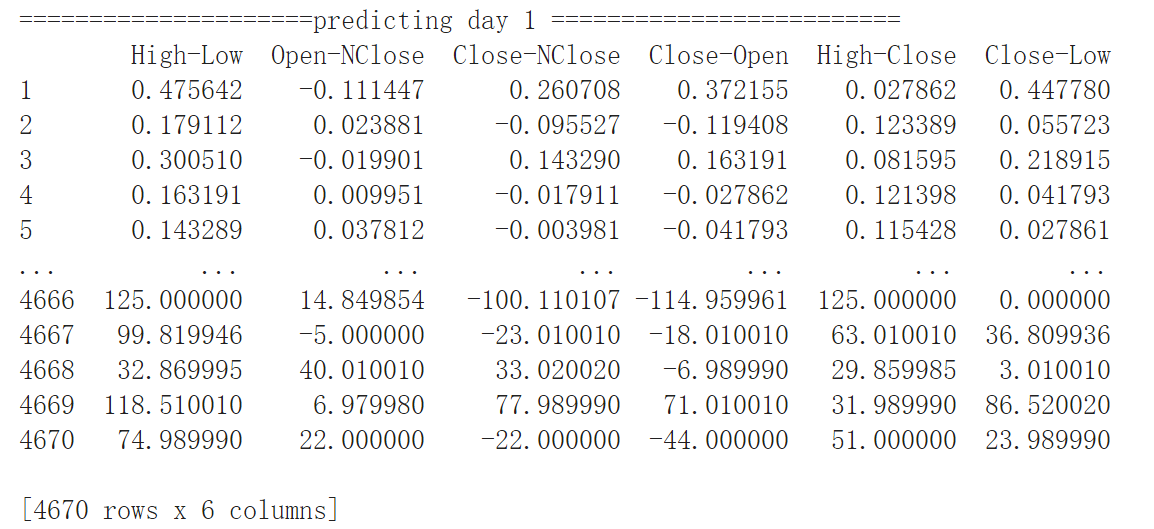
As you can see, now the data has 6 columns based on “open”, “close”, “high”, and “low”, which are absolute price values. Column “value”, which we will predict, is generated according to the difference between today’s closing price and the closing price of n days ago. If it is larger than 0, it means that today's price rises compared to n days ago; If it equals 0, it means that today’s price doesn't rise or fall compared to n days ago; If it is smaller than 0, it means that today’s price falls comparing to n days ago.

Take Kweichow Moutai (600519) as an example. The original data is as follows. (**Figure 1**)



**Figure 1.** Original data of Kweichow Moutai (600519).

After procession, the data is as follows. (**Figure 2**)



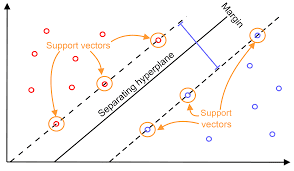
**Figure 2.** Processed data of Kweichow Moutai (600519).

When predicting day 1, N equals 1, so “Close - NClose” equals today’s closing price minus yesterday’s closing price. For example, in the first line of the second table, Close - Nclose = 0.260708, it comes from 7.335628 - 7.074920. It is larger than 0, which means this day the stock rises compared with the day before this day, so “value” of this day equals 1. Our SVM (Support Vector Machine) model is developed to forecast the “value”.

Now our goal converts from predicting the price value of someday in the future to predicting the stock will rise or fall that day. Correspondingly, our prediction result only has 3 values: 1 means the stock will rise; -1 means the stock will fall; 0 means the price doesn’t change.

**1.2 Model Training**

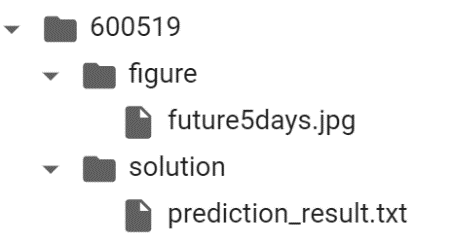
SVM **[3]** can divide the target result to be predicted into “up” and “down” according to several feature sample data, and use the SVM algorithm to train the relationship between these feature values and the “up” and “down” of the stock, that is, through the feature, the value divides the boundary between the “up” and “down” of the specified stock, so that once other stock characteristic data is input, the corresponding rise and fall can be predicted. (**Figure 3**)



**Figure 3.** SVM algorithm. (Source: https://static.packt-cdn.com/products/9781789345070/graphics/6a831600-9a0d-429f-9d34-d957c45b9517.png)

Here we set kernel equals poly and degree equals 20. Then according to the prediction result, we calculate the correct rate to see the accuracy.

Still, take Kweichow Moutai (600519) as an example. The training result folder is as follows. (**Figure 4**)



**Figure 4.** Result file structure of Kweichow Moutai (600519).

Future5days.jpg is a graph showing the prediction result; prediction\_result.txt is a text file storing the result for our trading part.

There will be 3 prediction results: 1, 0, and -1. 1 means this stock will rise; -1 means this stock will fall; 0 means this stock won’t change. Then we will give out operations according to the prediction. (**Table 1**)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Day 1 | Day 2 | Day 3 | Day 4 | Day 5 |
| buy | hold | hold | sell | hold |

**Table 1.** Operations for future 5 days.

**2. Experiments and Evaluations**

**2.1 SVM Prediction Result**

We selected the top 5 stocks by market capitalization to test our model. **(Table 2)**



**Table 2.** Top 5 stocks by market capitalization.

We used SVM model to predict the stock and give the operations. (**Table 3**)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Stock** | **Day 1** | **Day 2** | **Day 3** | **Day 4** | **Day 5** |
| 600519 | buy | hold | hold | sell | hold |
| 601288 | hold | buy | hold | sell | hold |
| 601398 | hold | hold | hold | hold | hold |
| 601318 | buy | hold | sell | hold | hold |
| 600036 | buy | hold | hold | hold | sell |

**Table 3.** Operations for future 5 days.

**2.2 Accuracy and Comparison with LSTM**

Operations of LSTM model are illustrated as follows. (**Table 4**)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Stock** | **Day 1** | **Day 2** | **Day 3** | **Day 4** | **Day 5** |
| 600519 | 2105.601 | 2145.876 | 2245.104 | 2162.372 | 2237.245 |
| buy | hold | sell | buy | sell |
| 601288 | 3.075491 | 3.022646 | 3.562933 | 3.529859 | 3.536449 |
| hold | buy | sell | buy | sell |
| 601398 | 5.378158 | 5.417605 | 5.439079 | 5.530206 | 5.373069 |
| buy | hold | hold | sell | hold |
| 601318 | 86.36325 | 83.28569 | 85.2497 | 87.23275 | 86.70916 |
| hold | buy | hold | sell | hold |
| 600036 | 53.55167 | 54.04567 | 54.48933 | 55.10054 | 55.49297 |
| buy | hold | hold | hold | sell |

**Table 4.** Operations for future 5 days.

As you can see, LSTM result contains the prediction prices, which are absolute values. We pick up the top 2 stocks to compare the results of LSTM and SVM with the real prices to see the model accuracy. (**Figure 5, Figure 6**)

**buy**

**buy**

**buy**

**sell**

**hold**

**hold**

**sell**

**hold**

**sell**

**hold**

**Figure 5.** Comparison example of Kweichow Moutai (600519).

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **LSTM** |  |  |  |  |  | **SVM** |  |  |  |  |
| **Day 1** | **Day 2** | **Day 3** | **Day 4** | **Day 5** |  | **Day 1** | **Day 2** | **Day 3** | **Day 4** | **Day 5** |
| 2105.6 | 2145.88 | 2245.1 | 2162.37 | 2237.25 |  | buy | hold | hold | sell | hold |
| buy | hold | sell | buy | sell |  |  |  |  |  |  |

Correct rate: 2/5 Correct rate: 5/5

**sell**

**hold**

**buy**

**hold**

**hold**

**hold**

**hold**

**hold**

**hold**

**hold**

**Figure 6.** Comparison example of ICBC (601398).

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **LSTM** |  |  |  |  |  | **SVM** |  |  |  |  |
| **Day 1** | **Day 2** | **Day 3** | **Day 4** | **Day 5** |  | **Day 1** | **Day 2** | **Day 3** | **Day 4** | **Day 5** |
| 5.378158 | 5.378158 | 5.439079 | 5.530206 | 5.373069 |  | hold | hold | hold | hold | hold |
| buy | hold | hold | sell | hold |  |  |  |  |  |  |

Correct rate: 3/5 Correct rate: 3/5

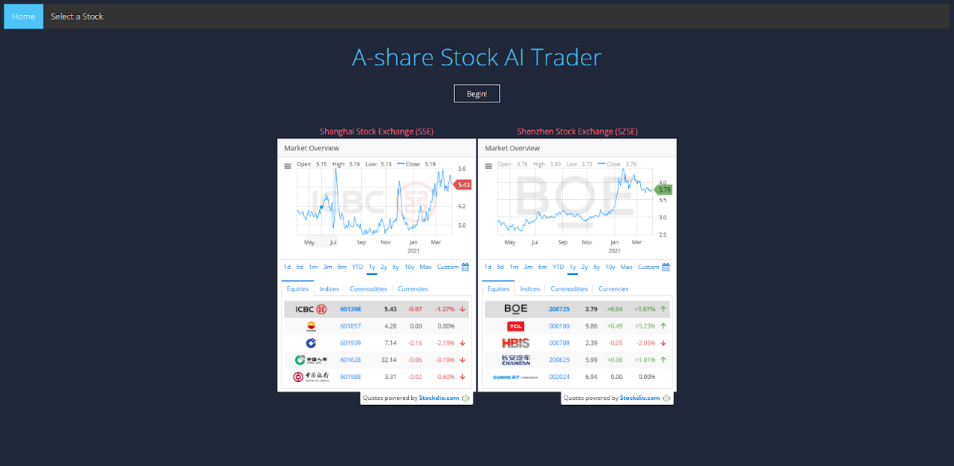
Based on the previous experience, it is kind of impossible in itself to predict the quasi-stock price, especially for A-shares. The factors affecting stock prices are very complex and opaque, as well as the reflexive theory of the financial investment field, making this prediction more experimental. Generally speaking, the correct rates of LSTM and SVM are slightly more than 50%. There are too many factors affecting the rise and fall of stocks, so I think our model has a lot of room for improvement in terms of features. Maybe adding more attributions to our data will make our model more precise. We will do further researches and experiments.

**3. Web Application**

Django web framework **[4]** is developed for users to interact with pre-trained models and get trading strategies. The recorded demo video can be accessed from the link of <https://youtu.be/mKN3sfGr4Yw>. The main pages are as follows.

1) Home page

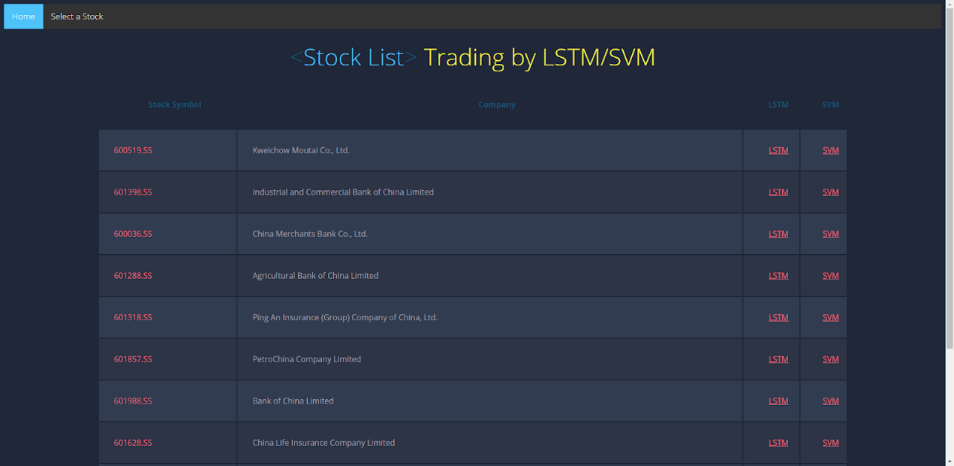
It illustrates the entry point for the web application and providing real-time stock market information **[5]** from two A-share stock exchange markets. (**Figure 7**)



**Figure 7.** Home page.

2) Stock list with LSTM/SVM

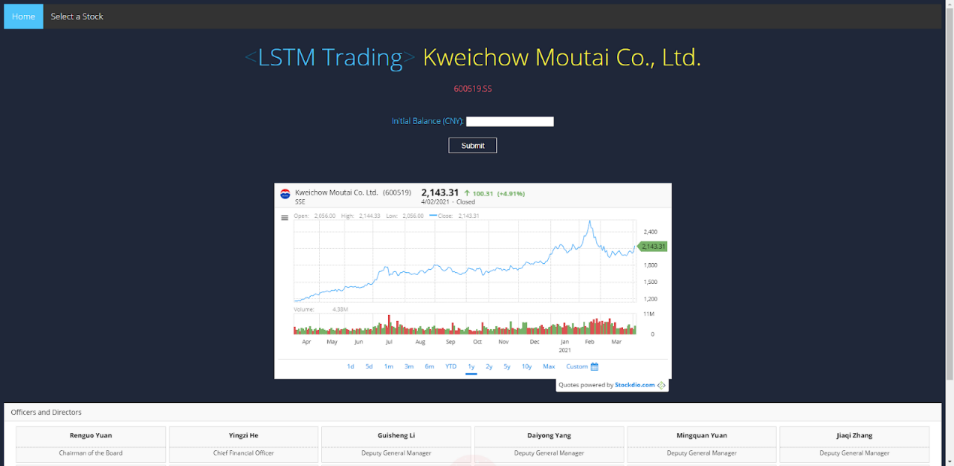
It lists all A-share stocks in a table, and providing two models for trading, i.e., LSTM and SVM. (**Figure 8**)



**Figure 8.** Stock list with LSTM/SVM.

3) Start LSTM trading with real-time data

It shows the LSTM trading parameters, initial balance, which can be filled by users before running LSTM trading. It also provides a detailed stock profile for a specific chosen company. (**Figure 9**)



**Figure 9.** Start LSTM trading with real-time data.

4) Start SVM trading with real-time data

It shows the SVM trading page. Similar to the LSTM page, it also provides a detailed stock profile for a specific chosen company. (**Figure 10**)



**Figure 10.** Start SVM trading with real-time data.

5) Suggested LSTM and SVM trading strategy

It shows the pages of LSTM (left) and SVM (right) trading strategies. Each strategy gives both the suggested operations (buy, sell, hold) for the following trading days. The graphs for the trend of the predicted stock price in the future five days are also illustrated.

Additionally, LSTM page provides the calculated estimated profit and balance after the trading operations. The graphs of predicted stock price vs actual stock price for five days are also provided. (**Figure 11**)

|  |  |
| --- | --- |
| https://lh4.googleusercontent.com/DPCtZM9xHGqCecRYV9gzMazXzmT-W22icm42U-SMVRclqU0gQTr_mwQcSA2Fi3F4JGXoXVbZf20Rg8Dd6a5YyElFSZrw0d0WlrSXCvIN_RiG_C6kOJe0J2sBjAg5l53uLYvQARNYHPw | https://lh3.googleusercontent.com/nzyOmR_7fDEVpqRTQahc4dIHbFeYwisnQ68Pq-b0LN8zuiL3E18F3r8LadoCgXOr1cxBGAM4QocDglJM-a-qOPqKlElgMovlIlMOdmAsvCfDTcMLIu4ThFWqaNVUJm0E3Dp05iSlVr8 |

**Figure 11.** Suggested LSTM and SVM trading strategy.

**4. Challenges**

**4.1 Solved in Milestone 3**

The training of SVM shows that SVM may fail to be trained if all the values are 1 or -1; SVM needs y to be at least 2 values. If this situation occurs, we can increase the training sample size to solve it.

It needs additional effort to integrate the Django web framework with machine learning Python units, usually written in the Jupyter Notebook. Also, a user can take actions of data updating, training, predictions, visualizing data via the web application. In our system, a user interacts with RESTful API to communicate with the view controller. The view controller handles all the tasks, including data processing, machine learning, and visualizations. The machine learning modules are packed as functions for the view controller to invoke.

Single thread cause blocks in the time-consuming training task if many users operate at the same time. Multi-thread processes are created in the view controller; thus, supporting multi-user operations to avoid blocks.

**4.2 Solved in Final Project**

Since stock trading data is updated every day, the data and the pre-trained models need to be updated frequently. Since we have thousands of stocks while each stock is associated with more than one model, an automatic updating script is used to do batch downloading and training tasks during a non-trading time, e.g., closing hour to opening hour.

Currently, either LSTM or SVM makes predictions alone, and they do not incorporate to benefit from each other. A new model ARIMA (autoregressive integrated moving average) **[6, 7]**, will be implemented; therefore, a final predicted result is determined together by LSTM, SVM, and ARIMA.

In Milestone 3, we use default plots from Matplotlib. However, this scientific style is not attractive enough for an AI trading frontend. Therefore, a more powerful, interactive charting and visualization library called Echarts **[8]** is used to replace Matplotlib to make a more attractive website.

**5. Initial Plan and Achievement**

The following table (**Table 5**) shows the milestone tasks we have completed. As planned, we have investigated and implemented more models, i.e., SVM, ARIMA for price predictions, as suggested by the professor. We are also developing a nicely looked UI to illustrate all the results, and you could see it in the final project.

|  |  |  |
| --- | --- | --- |
| **Milestone 1** | **Milestone 2** | **Milestone 3** |
| * Literature survey * Methodology investigation * System design initiative | * Data collection and procession for OHLC of 2000+ stocks via Python and yahoo\_fin API **[9]** * Determine the training method of the model: LSTM model for stock price forecasting * LSTM model obtained and prediction made successfully * Codes showed | * Model training evaluation and tuning * Make Django web app assembled of both frontend and backend * Extend the training methods to get more models, and then make operation decisions based on the prediction results of all these models together (LSTM and SVM) |

**Table 5.** Milestone achievements.

**6. Final Project Plan**

1) Update data and model automatically in a set period

Since the stock price is daily updated, the input training data and pre-trained models will be updated every day to ensure the most precise predicted results.

2) Develop a new model

In addition to the existed LSTM and SVM models, a third model ARIMA (autoregressive integrated moving average), is being developed to provide additional price prediction support.

3) Make trading decisions based on the integration of three models, i.e., LSTM, SVM, and ARIMA

The suggested operations (buy, sell, hold) will be voted by the results from these three models instead of only one model.

4) Deploy to the cloud platform

The whole web application, including source data processing and model training, will be deployed to the cloud, such as AWS and Google Cloud Platform. Therefore, everyone over the world can get access to this AI trading service.

5) Focus on the frontend UI, especially the improvement of charts

The professional chart drawing tools such as ECharts will replace the default Matplotlib, thus showing remarkably marvelous and vivid price charts on the frontend and improving the user experience.

**7. References**

[1] Le, X.H., Ho, H.V., Lee, G. and Jung, S., 2019. Application of long short-term memory (LSTM) neural network for flood forecasting. *Water*, *11*(7), p.1387.

[2] https://towardsdatascience.com/predicting-stock-price-with-lstm-13af86a74944

[3] Henrique, B.M., Sobreiro, V.A. and Kimura, H., 2018. Stock price prediction using support vector regression on daily and up to the minute prices. *The Journal of finance and data science*, *4*(3), pp.183-201.

[4] https://docs.djangoproject.com/en/3.2/

[5] https://services.stockdio.com/howtouse

[6] https://blog.csdn.net/qq\_40707407/article/details/81938941

[7] https://programtip.com/zh/art-524

[8] https://echarts.apache.org/en/index.html

[9] https://pypi.org/project/yahoo-fin/