FINAL REPORT

Stock investment Algorithmic Trading

Team members:

Selene Xiang - shijiex@usc.edu
ID: 2284690632

Chengjun Liu - cliu4428@usc.edu
ID:7116099552

Mahesh Pandit - maheshpa@usc.edu
ID: 1845691455

Marko Uskokovic - uskokovi@usc.edu
ID: 5295349658

DSO 530: Applied Modern Statistical Learning Methods
Professor: Xin Tong

Problem Definition:

Our job was to build a trading algorithm that will maximize $1,000,000 we received in our investment account while taking into consideration investors' risk-averse psychology.

This Project Presumes:

1. We have 765 days of the daily stock data of 50 companies, which are constituents of the SSE 50 index.
2. On the evening of day 504, we received $1,000,000 in our investment account which does not generate interest. From Day 505 to Day 756, we trade the 50 stocks.
3. By the end of Day 756, we are required to sell off all our stocks on the last trading day, or in other words to have cash only in our investment account.

Project Background:

Algorithmic trading is widely adopted and many traders are using it from investment banks to pension funds, mutual funds, and hedge funds. This type of trading was developed to make use of the speed and data processing advantages that computers have over human traders.

It's easy to justify the relevancy of this project when we know that in the US the percentage of algorithmic trading is about 70% while in emerging economies is roughly 40%. A centerpiece of algorithmic trading is prediction models based on past data and statistics and machine learning techniques. In this project, we will be conducting algorithmic trading using supervised learning approaches.

This type of trading is valuable because the execution of a larger order can be too fast for human traders to react to. Moreover, algorithmic trading, in theory, can generate profits at the speed and frequency that is impossible for a human trader.

One of the most obvious benefits is ruling out the impact of human emotion or errors on trading activities. The systematic approach is appealing for investors since decisions to buy or sell are based on clearly defined sets of instructions that consider the timing, price, quantity, and many other predefined parameters.

Some of the benefits of algorithmic trading that we tried to take advantage of are:

- Trades are made at the best possible price and time
- Trades are instant and accurate preventing missing out on the great opportunity due to significant price changes in the short time period
- Reduced transaction cost

- Reduced risk of manual errors based on emotional, psychological, or any other factors applied to human traders

Data Available:

As mentioned earlier, we have 765 days of the daily stock data of 50 companies, which are constituents of the SSE 50 index. In our data, we can clearly distinguish common features and features engineered from the raw data.

Common features:

Open Price, High Price, Low Price, Volume, Adjusted Close Price

Engineered features:

5-day moving average, 15-day moving average, 5-day return, 15-day return, Relative strength index, Stochastic oscillator, Chaikin AD line, Price rate of change, On balance volume

Except for the engineered features mentioned above, we also create following features:

- MACD: a trend-following momentum indicator that shows the relationship between two moving averages of a security's price.
- 3-Day Return & 3-Day Moving Average
- Lagged open, high, low, close price for up to 5 days (totally 20 features)

Since we were using momentum trading strategy, the core of which is to correctly detect the right trend and therefore means we need to predict and classify each possible transaction day into an upward trend or downward trend, we created a binary output variable to denote whether a particular day is on an upward trend or not.

- Y Label: 1 if a day is on upward trend. To be labeled 1, following criteria should be met:

  Firstly, in the following 5 days for a record, there must be at least three days showing a positive return rate. Secondly, the price must increase more than 5% after 5 days from the labeling day.

We only labeled training and testing dataset, which start from day 1 to day 504. Since we need to use data from the future 5 days, the last day with Y label is day 499.

Now we are able to find certain patterns between existing information from past data and the future price trend.

Oversampling and Undersampling:

After labeling trends, we checked whether the 2 labels are balanced and found that only 15% of the records were labeled 1. We used SMOTE algorithm to try both oversampling and undersampling methods for each stock in the portfolio. Since undersampling only gave us a dataset with around 150 records for training and testing for each stock, together with the number of features we have, model performance were unsatisfactory for almost every stock in our portfolio; therefore, we chose oversampling as our final method to make balanced datasets.

Trading Strategy

Portfolio selection

Before building models, we did portfolio creation using the closing price from day 1 to day 504 of all 50 stocks based on **Markowitz's theory**. According to Markowitz's theory, all asset combinations lying on the efficient frontier are valid and best combinations that can achieve optimal returns given different levels of risk. To build efficient frontier, we used mean historical returns to estimate future returns and covariance shrinkage techniques, which involve combining the sample covariance matrix with a structured estimator to reduce the effect of erroneous weights, to estimate risk. Although mean historical returns and covariance shrinkage have several disadvantages, yet they are easily interpretable and very intuitive, and therefore reasonable to perform as parameters in the portfolio selection.
From the efficient frontier, we chose an investment portfolio based on the **objective function** to maximize Sharpe Ratio that can help to find the optimal return per unit risk.
Our final investment portfolio consist of the following stocks and their corresponding investment ratio:

SH601390: 0.5%	SH600050: 3.6%	SH600030: 4.6%
SH601088: 7.7%	SH600036: 9.5%	SH601318: 11%
SH600585: 14.3%	SH600406: 21.7%	SH600519: 27.1%

Based on the portfolio selected, we are expected to achieve a return rate of about 30.4% and Sharpe Ratio of 1.52.

Trading strategy

We follow the **Momentum investing method**. The goal is to work with volatility by finding the buying opportunities in short-term uptrends and then sell when the securities start to lose momentum. We did weekly evaluations and transactions. Starting from day 505, if the model output is 1, which means there is an upward trend for the following 5 days, we will buy in this particular stock with all the capital assigned to it. Then, after 5 days (effective transaction days for a week), we would evaluate this stock asset again to determine whether the next 5 days are still on an upward trend; if so, we would keep this asset for another 5 days; if not, we would sell this asset for cash. If the model output is 0 on day 505, we would not buy the specific asset, and

then evaluate this stock every day until we find the first buy-in transaction day indicated by the model. Then we would follow the same procedure as above.

Review of the Statistical Learning Approaches:

To build the trading algorithm we built classification models using the following:

1. Logistic Regression
2. Boosted Trees
3. Random Forest
4. Neural Networks

We took the following steps while creating each classification model:

- All the records from before day 505 were randomly divided into training data (70%) and testing data (30%)
- All the variables in the training and testing dataset were normalized using the mean and standard deviation of the variables in the training dataset
- Minority classes were oversampled using SMOTE
- The buying and selling actions were performed using the data of the records after day 505
- Based on portfolio optimization theory and using data before day 505, we selected 9 stocks with their corresponding investment ratio.
- All 4 classification models were applied to each of the 9 selected stocks.
- The performance of each model was evaluated by calculating the percentage of all label 1 found in the top 25% minus the proportion of label 0 in the top 25% of the data sorted by probabilities.

Overview of the models used:

Logistic Regression

Logistic regression was used to determine the probability of each record belonging to class 1. Since logistic regression is a linear classification model that finds linear relationships between variables, we used the performance of the logistic regression classification model as the baseline performance for the remaining classification models. This model achieved an average classification score of around 13% for all 9 stocks.

Random Forest

Random forest is an ensemble classification method, where each record is assigned a label that is the average of all the labels assigned to it by an ensemble of randomly generated trees. We evaluated the performance of the model by changing various hyperparameters such as:

- n_estimators (The number of trees used in the classification model): 100, 200
- min_samples_leaf (The minimum number of samples in each leaf of a tree): 3, 5, 10,

20

- max_features (The maximum number of features considered for a split in each tree): 3, 5, 10, 15

The best performance for this model was achieved when

- n_estimators = 200
- min_samples_leaf = 5
- max_features = 10

The random forest model achieved the best performance for the stocks SH600088 and SH600390. However, since these two stocks did not have high-momentum days after day 505, no buy/sell actions were made on these stocks.

Boosted Trees

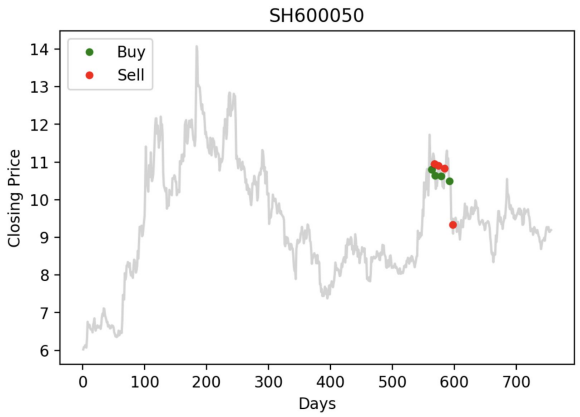
Boosted trees is also an ensemble of trees, where a series of weak trees combine to form a strong classifier. The trees are built in a sequence, with each tree improving upon the classification error of the previous tree. We used the XGBoost algorithm in order to implement the boosted trees. We tried the following hyperparameters:

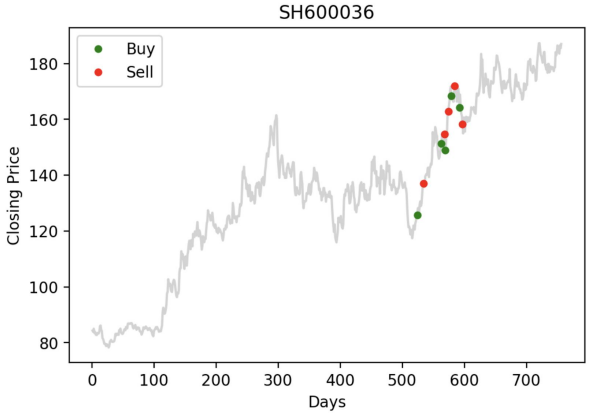
- number of trees: 200, 250, 300, 350, 400
- maximum depth (The number of nodes in each tree): 2, 4, 6, 8
- learning rate (The percentage of improvement achieved by each tree): 0.001, 0.01, 0.1, 0.3
- reg_alpha (L1 regularization term): 1, 5, 10, 15

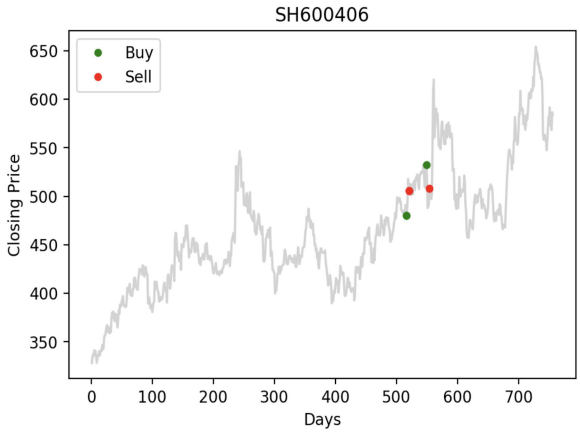
The best performance was achieved with the following hyperparameters:

- number of trees = 250
- maximum depth = 5
- learning rate = 0.1
- reg_alpha = 10

This model achieved the best classification for the stocks SH600036, SH600050, SH600406. The buy and sell decisions that were made are shown below:

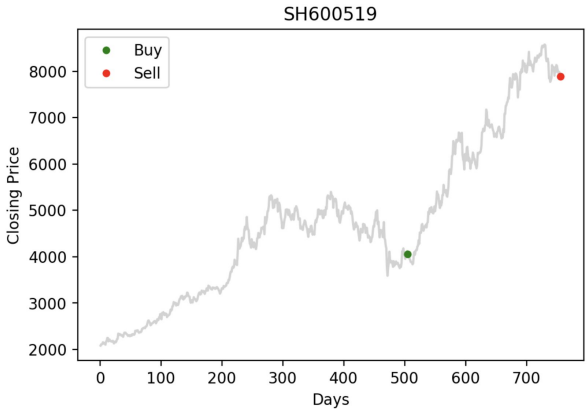


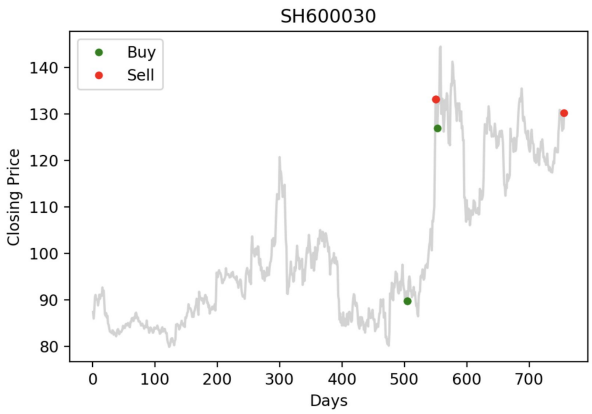


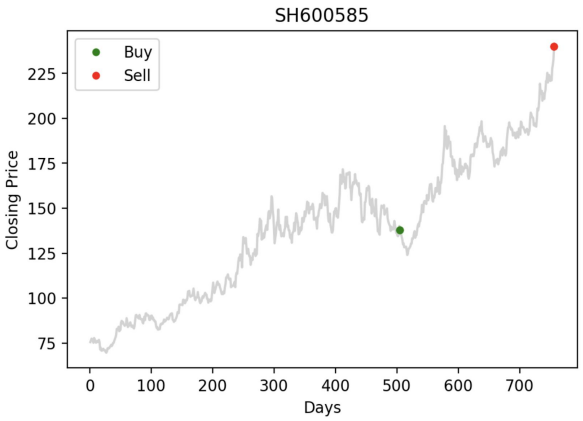


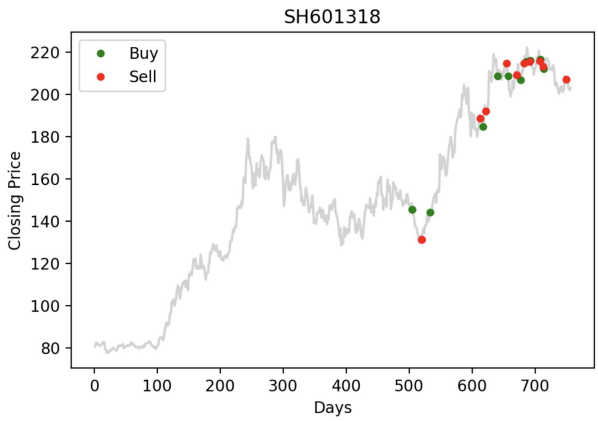
Neural Networks

Neural networks are a set of algorithms, modeled loosely after the human brain, that are designed to recognize patterns. It has an input layer, one or more hidden layers and an output layer. Neural networks can recognize an unknown function f(x) = y, for the input x and output y. In order to prevent overfitting the data, we implemented a neural network with a single hidden layer, and the number of nodes as 4, 6, 8 and 10. The best performance was achieved when the number of nodes in the hidden layer was 8. This model has the best performance for the stocks SH600030, SH600519, SH600585 and SH601318. The buy and sell decisions made are shown below:









Summary of the results:

- Total Earnings = about **$400,000** in profits

By the end of Day 756, we were required to sell off all our stocks on the last trading day, or in other words to have cash only in our investment account.

Our **return** on $1,000,000 investment is **40.6%**

- Sharpe ratio = **2.22**

It measures the performance of an investment compared to a risk-free asset, after adjusting for its risk. It is defined as the difference between the returns of the investment and the risk-free return, divided by the standard deviation of the investment.

Investors consider sharpe ratio greater than 1 as good, greater than 2 very good, greater than 3 excellent, while ratio lower than 1 is considered suboptimal.

- The number of days our trading strategy incurs a loss = **121 days**

Stock	Return	Sharpe Ratio	Earning Days	Holding Days
SH600519	0.930	2.352	129	250
SH600585	0.728	2.182	138	250
SH600030	0.492	1.300	125	247
SH600036	0.261	1.119	101	201
SH601318	0.097	0.533	95	195
SH601088	0	0	0	0
SH601390	0	0	0	0
SH600406	-0.021	-0.188	6	10
SH600050	-0,106	-0.846	8	20
Overall	0.406	2.22	130	

Improvements:

Although we verified our strategies with 40% profits in around 200 days, we still notice that some stocks selected for momentum strategies are not used or making profits, thus we believe there is still space for improvements like promoting the prediction accuracy. Based on our analysis and recommendations from Professor Xin Tong and our classmates, some of the next steps in our project are as follow:

1. Introduce the 3rd trend - **Float**

We currently recognize 2 trends in our data: Upward and Downward trend. This is a common sense towards the trends of the stock market, but based on the fact that the prediction accuracy could be further improved, we believe that other states of stock trends can be discovered. We will introduce the 3rd trend - Float because it is possible that in a period of time, the stock price jumps up and down frequently. We are aware of the fact that the phenomenon is very often. By adding the third classification, we can describe the stock trends more accurately and improve our prediction model by simply including multi-class classification.

2. Update historical data

New stock price records are more correlated with predictions in the near future, so updating strategies with new records will improve accuracy. Usually stock price in one day is mostly explained by 20-50 days before. Because our process includes stock selection and prediction, new data can be applied to both the processes. On the one hand, we update classification models with new records, rolling the training data window for every 10 or 20 days, to make better predictions for the next window. On the other hand, we update the correlation of stocks by extending the data used to calculate correlation. Based on minimum risk theory, the real correlation between stocks are approached by as long records as possible. With renewed correlations, we re-adjust portfolio allocation to reach minimum risk for a rolling window.