

CS 403 PROJECT

Automated Stock Trading Using Machine Learning

Team :

Arpit Singh [120050037]
Naveen Sagar [120050026]
Tapish Raniwal [120050023]
Pintu Lal Meena [120050018]

Proposed Idea:

Today using algorithms for trading decisions has become a prevalent practice in major stock exchanges in the world. The proposed idea was of automated stock trading using machine learning algorithms. It was to take past stock data, learn models from them and predict future stock prices using machine learning algorithms for most profit.

Work Done:

We have used different algorithm like logistic regression and SVM to predict sock data. In these two SVM turned out to be more accurate.

A. Data Source :

- We have taken data from <http://www.bseindia.com> with parameters as date, open price, high price, low price, close price, WAP(Weighed Average Price), no. of shares and many others. We took TATA stocks from here.
- We have also taken data from <http://finance.yahoo.com/q/hp?a=00&b=1&c=1970&d=03&e=28&f=2016&g=d&s=FB%2C+&q1=1> with parameters as date, open price, close price, high price, low price and volume. We took Facebook stocks from here.

B. Data modification: We have written scripts that take raw data and extract it in an organized way according to parameters we need. We created intervals of 'csize' days (the length of the interval can be changed but is currently set at 10). The parameters we extracted are start date, end date, open price, close price, low price, high price, open volume, close volume, low volume, high volume.

C. Features we extracted that will be used to model the data are: 1) percentage change in open price, 2) percentage change in high price, 3) percentage change in low price, 4) percentage change in open volume 5) percentage change in high volume 6) percentage change in low volume, 7)ratio of open price andchange to the most recent lambda number of intervals high-low spread and 8)ratio of open volume change to the most recent lambda number of intervals high-low spread. These can be expressed using the formula below:

$$\begin{aligned} & (P_{open}^t - P_{open}^{t-1})/P_{open}^{t-1} \\ & (P_{high}^{t-1} - P_{high}^{t-2})/P_{high}^{t-2} \\ & (P_{low}^{t-1} - P_{low}^{t-2})/P_{low}^{t-2} \\ & (V_{open}^t - V_{open}^{t-1})/V_{open}^{t-1} \\ & (V_{high}^{t-1} - V_{high}^{t-2})/V_{high}^{t-2} \\ & (V_{low}^{t-1} - V_{low}^{t-2})/V_{low}^{t-2} \end{aligned}$$

$$\begin{aligned} & (P_{open}^t - P_{open}^{t-1})/(PH_{\lambda}^t - PL_{\lambda}^t) \\ & (V_{open}^t - V_{open}^{t-1})/(VH_{\lambda}^t - V_{low}^t) \end{aligned}$$

where:

$$\begin{aligned} PH_{\lambda}^t &= \max(t-\lambda \leq i \leq t-1) P_{high}^i \\ PL_{\lambda}^t &= \min(t-\lambda \leq i \leq t-1) P_{low}^i \end{aligned}$$

$$PL_{\lambda}^t = \min(t - \lambda \leq i \leq t - 1) V_{low}^t$$

$$PL_{\lambda}^t = \min(t - \lambda \leq i \leq t - 1) V_{low}^t$$

Note: Here $y(t)$ is 1 if $priceclose(t) > priceopen(t)$ else it is 0.

D. Data modelling and the corresponding optimization:

- We tried to model the data with different models like linear regression, etc. but we found that SVM was the most effective one. We create the corresponding model using the features mentioned in the above point.
- Also, we checked which combination of the features we should choose to model the data and we chose the combination of the eight features (mentioned in the previous point) because it gave better results.
- To optimize it further, we do a stratified 10-fold cross-validation (preserving the class ratios within each fold).
- Further, we take the best C, kernel and gamma by using Grid Search (which exhaustively considers all the parameter combinations and takes the best one).
- Also, we go through different values of λ and pick the one that gives us the best results.

E. Results: We tested on two stocks (with the models and the optimizations mentioned in the above point) which are as follows:

- FB Stock: We model the FB stock listed from 2012 to 2016 where training data is from 2012 to 2014 and testing data is from 2015 to 2016.

With $\lambda = 5$:

Profit=11.989989
 Accuracy= 0.545454545455
 Precision = 0.545454545455
 Recall= 0.592592592593

With $\lambda = 3$:

Profit= 13.31999
 Accuracy= 0.545454545455
 Precision = 0.545454545455
 Recall= 0.608695652174

We see in this case ($\lambda=3$) that there was an increase in profit and recall without any change in precision and recall.

We choose $\lambda=3$ for better results in our calculations which is also evident in the above case.

- Tata Stock: It is a much more rigorous data than FB stock. We model the data on TATA stock listed from 1995 to 2016 where training data is from 1995 to 2014 and testing data is from 2015 to 2016.

We use $\lambda=3$ in this:

Profit= 29.85
 Accuracy= 0.59375
 Precision = 0.59375
 Recall= 0.5

Future Work:

Predicting stock market is a challenging task due to the trends being influenced by various factors such as noise and volatility. Also, the stock prices reflect all the currently available information and depend on various local-nodes that changes constantly thus it is necessary to capture those changes. We would try to check other stock markets to find location where the algorithm is able to perform better. We can also use other algorithm like reinforcement learning. Feature selection is the most important part of this algorithm so finding better or more descriptive features would be great in making the results even better.

Conclusion:

Factoring in features of high dimensionality after careful selection can be significant in improving the results and our analysis of SVM compared to other regressions was able to show this. We expect that this is the case because of higher-dimensionality which increase the linear separation of the dataset.

Our conclusion is that Machine Learning has great potential in the field of predicting stocks and profit can be increased by finding more features and by other algorithms.

Reference:

<http://cs229.stanford.edu/proj2012/ShahDaiZhong-AutomatedStockTradingUsingMachineLearningAlgorithms.pdf>