NUS INVESTMENT SOCIETY

Academic Research Review

Tuning machine learning models for binary prediction of individual stock price

EXECUTIVE SUMMARY

In this report, we document the use of random forest and support vector machine classifiers for feature selection and individual stock price prediction. While the focus is on predicting if the day's stock price outperforms or underperforms a market index, the process can be easily modified to predict other similar variables.

INTRODUCTION AND MODEL

The future movements of stock prices have been always in the center of attention for investors and therefore, being able to reliably predict whether a price will rise or fall is an important research area. As the fluctuation of stock market depends on many financial indicators and a full analysis requires managing a lot of data, there is an increasing interest in predicting future market behavior using machine learning techniques.

However, it is difficult to come up with a unified model that is equally useful for all types of stocks, a conclusion supported by current literature in this research area. In the interests of accuracy, it is more useful to create a model where the underlying algorithm is the same but the exact predictive variables chosen (financial indicators) are not.

The aim of the following project is to analyse historical stock price movement using financial indicators and develop a model that can predict the fluctuations in price of a single stock. In particular, our models in this paper predict if a stock will outperform or underperform the overall S&P500 index for the day. Thus, we are dealing with a typical binary classification problem.

DATA

To conduct our research, we collected IBM daily stock data and its technical trading indicators, which can be obtained from Bloomberg. The data set includes: trading volume, open, highest, lowest, and close prices of the trading day and other financial indicators such as Aroon Oscillator and 50-day Moving Average, for a total of 44 features. The list of all indicators is presented in the Appendix. We used a data interval from 10/17/2011 to 09/09/2016.

For evaluations, the data set was randomly split using a ratio of 70% to 30% into a training and test sample.

METHODOLOGY

First, due to large dimension of our data set, we concentrated on variable selection. Among the most popular machine learning techniques for variable selection are random forests, due to their relatively good accuracy, robustness and fair simplicity. We briefly describe the notion of random forest that consists of a number of decision trees. Every node in the decision trees is a condition on a single feature, designed to split the dataset into two so that similar response values end up in the same set. The measure based on which the (locally) optimal condition is chosen is called impurity. Thus, when training a tree, it can be computed how much each feature decreases the weighted impurity in a tree. For a forest, the impurity decrease from each feature can be averaged and the features are ranked according to this measure and thus, importance can be assigned to each feature.

Next, we concentrated on the model building. We tested 2 possible models: a random forest classifier using all features and a Support Vector Machine (SVM) classifier using only features with importance higher than 0.02 (heuristically chosen). Depending on the This was done using the Python scikit-learn package, using svm.SVC for C-Support Vector Classification and ensemble.RandomForestClassifier for the random forest classification.

RESULTS

The small sample size of the dataset means that the set of training data sampled is also small (less than 1000 data points), which necessitates the use of k-fold cross-validation of our models. We performed 5-fold cross validation to assess the accuracy of both random forest and SVM in stock price prediction.

Random forest produced an accuracy of 80.1%. However, given that random forest was used to filter out the noise features and identify the most relevant predictors of stock price movements, the moderately high accuracy rate is not a major concern in this study.

On the other hand, the SVM model build using the features selected using the random forest approach gave an accuracy rate of 94.17% when using the parameters C = 2048 and gamma = 6561. The high accuracy rate yielded by the SVM model corroborates the importance of the features selected using the random forest approach.

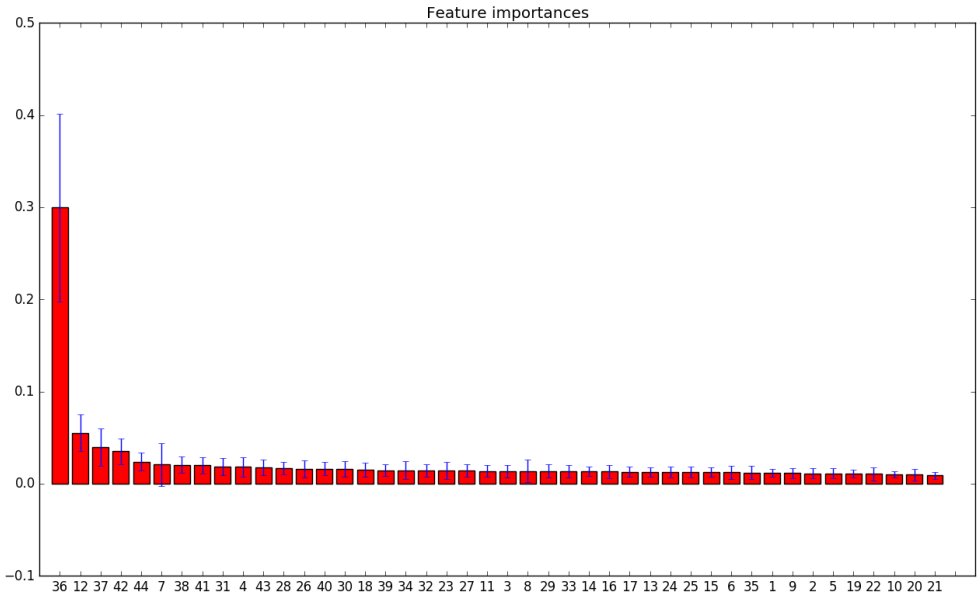

Figure 1. Feature Importances

We can also chart out the feature importances for IBM stock prices, which is an additional tool for investors to identify the most important financial indicators for a certain stock. From Figure 1, only feature

35, which is the Accumulation/Distribution Oscillator, was particularly important for prediction of IBM stock prices as compared to the others.

CONCLUSION

An SVM model performed better in predicting if the stock will outperform the overall S&P500 index that day as compared to a random forest classifier. However, SVM is also more prone to overfitting issues, which necessitates additional work in tuning hyper-parameters as well as pre-selection of a subset of features using random forest.

Improvements could include additional tuning of the subset of features entered into the SVM classifier, testing of these models on a wider variety of stocks, and a greater number of categories which would be more informative to the user (e.g. 4 categories as follows: greatly outperforms index, mildly outperforms index, mildly underperforms index, and greatly underperforms index).

APPENDIX: INDICATOR LIST

1	Open	23	Moving Average Envelope Upper Bound (3d)
2	High	24	Moving Average Envelope Average (15d)
3	Low	25	Moving Average Envelope Lower Bound (3d)
4	Close	26	Parabolic Systems
5	50-day Moving Average	27	Resistance 2 (Daily)
6	100-day Moving Average	28	Resistance 1 (Daily)
7	200-day Moving Average	29	Pivot Point (Daily)
8	Bollinger Band Upper Limit	30	Support 1 (Daily)
9	Bollinger Band Moving Average	31	Support 2 (Daily)
10	Bollinger Band Lower Limit	32	Max (20d)
11	Bollinger Band Width	33	Min (20d)
12	%B Indicator	34	Max-Min Retracement
13	Ichimoku Conversion	35	Accumulation
14	Ichimoku Base	36	Accumulation Distribution Oscillator
15	Ichimoku Lagging Span	37	Accumulation Distribution Oscillator Signal
16	Ichimoku Leading Span A	38	Aroon Oscillator
17	Ichimoku Leading Span B	39	Aroon Up
18	Kaufman's Adaptive Moving Average	40	Aroon Down
19	Keltner Channels Upper Bound	41	Average True Range
20	Keltner Channels Moving Average	42	Commodity Channel Index
21	Keltner Channels Lower Bound	43	Coppock Curve
22	Moving (Simple, 14, 0)	44	Chaikin Money Flow

Tuning machine learning models for binary prediction of individual stock price

Research Analysts: Karolina Stanczak, Lei Siqi, Quek Ying, Yang Xiya

---
