Predicting the direction of Stock movement by using a variety of Machine Learning Algorithms

Seminar: Introduction to Machine Learning

—

Final Project

by

Jacob Christensen, 17-743-204

Nicolas Loosli, 14-736-060

Amira Sakr, 13-208-913

Maximilian Tornow, 11-948-916

17th of April 2018

Abstract

The goal of this project is to predict the stock movement (up/down) for companies in the Dow Jones Industrial Average Index for 1, 3, 6 and 12 months into the future and to determine for which of these periods the best forecast could be made. To predict these trends, the following machine learning algorithms were used: Logistic Regression, K-Nearest Neighbor, Linear Discriminant Analysis, Quadratic Discriminant Analysis, Decision Trees, Random Forest and Support Vector Machines. To train the models two different data-sets (both from Jan 2006 - Dec 2015) were used. One was including only the financial ratios of the companies (derived from the Wharton database) and the other included the financial ratios, the 5y US Treasury Bonds, the 1y Bonds, the 90d Bills and the US Inflation (also derived from the Wharton database), previous returns (calculated from historic prices), as well as the Dow Jones Industrial Average Index (derived from Yahoo Finance). A last question that was analyzed was whether or not it is possible to use a reduced feature set without losing much of the information.

It turns out that increasing the forecast period enhances the performance immensely. The performance of the initial (f_1-score of 0.894) and the extended data-set (f_1-score of 0.897) for the period of 12 month are similar. By decreasing the the numbers of unimportant features, we can show that the performance (f_1-score of 0.872) doesn't suffer a lot. However, by doing so we can reduce the needed computing power.

1 Introduction

In this project multiple algorithms were used in an attempt to predict the stock movement (up/down) 1, 3, 6, and 12 months into the future for the companies of the Down Jones Industrial Average Index by using historic ratio, previous returns and additional features from Jan. 2006 to Dec 2015. After running the different algorithms, we compare the accuracy of their predictions, whether the highest accuracy was achieved by predicting the stock movement 1, 3, 6, or 12 months into the future, and which of the features had the most explanatory power.

2 Data Management and Exploratory Data Analysis

An outer join of the data-sets containing the share prices and the financial ratios was performed, by using the common names 'permno' and 'date' and 'public_date' as keys. From here on, we refer to this data-set as the 'initial data-set'. In order to answer the question "Does adding other features (beside financial ratios) improve the prediction?", we used an extended version of the data-set containing the the 5y US Treasury Bonds, the 1y Bonds, the 90d Bills and the US Inflation from the Wharton database. We then made the exact same join as for the initial data-set and added previous returns and the Dow Jones Industrial Average Index, derived from Yahoo Finance. We refer to this data-set as the 'extended data-set'.

Since most machine learning algorithms can't deal with missing values, which Python labels as NaN (Not A Number), this has to be considered before applying the algorithms. A closer look at the data showed that certain columns are missing the majority of it's observations, i.e. 'PEG_trailing' and 'sale_nwc' are respectively missing 21% and 32% of it's observations. We decided to delete all columns having more than 5% of its values missing and to remove all rows containing a single missing value. This left us with 2930 rows, which means we preserved 82 pct. of the initial data-set. We then created a 'response'-feature, which is a binary variable indicating whether the return for a given period is either strictly positive (1) or negative/zero (0)[1]. When doing the Exploratory Data Anaylsis (EDA) with respect to the 'response'-feature, we noticed that the distribution for the 1 month returns highly mirrors the random walk hypothesis for financial markets, which assumes up/down movements with equal probability (56.6 pct goes up and 43.4 goes down, for the monthly returns). But as the return-interval increases, the hypothesis seems to be less and less fitting, since 69.7 pct. of the 12 months returns are strictly positive. To the best of our knowledge, we do not need to re-balance the Class using tools like up- or down sampling, since we do not consider its distribution to be 'heavily' skewed.

[1] Our reasoning behind categorizing the zero-return as the down response is that trading costs would be incured when trading these shares

At first the algorithms were performed with all features, since we wanted to obtain a prediction from the models when given all available data. Later we use Principal Component Analysis and RandomForestClassifier to transform the data-sets into a new feature-set of a lower dimension in order to see how this affects the predictive power of our models.

3 Setup and the Algorithms

At first, as explained in the end of section 2, we assign all variables (besides the 'response') to our set of features and assign the 'response'-column as our response-variable. In a second step we split the complete data-set into a test- and a train set by using the train_test_split function from Sci-Kit learn package(sklearn). As encouraged in class, we split the train- and test-sets 70:30 and set stratify equal to the proportion of the distribution of our response-variable, so that the ratio is the same with respect to the stock movements.
In order to to combine the performance of multiple processing steps, the pipeline function from the sklearn package was used. It allowed us to combine multiple algorithms with multiple (and different) hyperparameters. In contrast to using the GridSearchCV independently, this does not 'contaminate' the test-set with information from the train-set. When we standardize the features [2]. Once we set up our pipeline and the parameter_grid (with all the nessesary classifiers and respective hyperparameters) we were able to run the gridSearchCV. The GridSearchCV now allow us to find the best possible training score by combining the different given models and hyperparameters. The hyperparameters are found manually, over multiple times, plugging and running the pipeline, while only substituting a single hyperparamter at the time. As encouraged in class, we use 5-fold cross-validation.
Hereafter,we print the different accuracy score and uncertainty measurements, which we do eight times since we have two different data-sets: initial and extended, and four different returns we wish to predict which are shown in section 4.
Then we found that the RandomForestClassifier does a better job summarize the content of our data-set, compared to the Principal Component Analysis, and in order to see if we can achieve a better results (with this smaller set of features) we, again, run the pipeline with the new, reduced set of features.

4 Results

With the above knowledge in consideration we performed for each time period and data-set one GridsearchCV in order to find the best CV accuracy, test score, precision score, f_1 -score and current classifier which has been used for the calculation. This is shown in the tables below.

[2] All the algorithms that we use, beside Decision Trees and Random Forest, are sensitive to the magnitude of the data, which is why we needed to scale the features.

In the table below the performance each forecast period for the initial data-set is shown:

Forecast period:	CV accuracy	Test score	precision	f_1 score	classifier [3]
1 Month	0.58	0.54	0.5579	0.69	RFC
3 Month	0.73	0.71	0.7582	0.7596	SVC
6 Month	0.79	0.80	0.8232	0.8443	SVC
12 Month	0.85	0.86	0.8797	0.8974	SVC

Table 1: Using the initial data-set

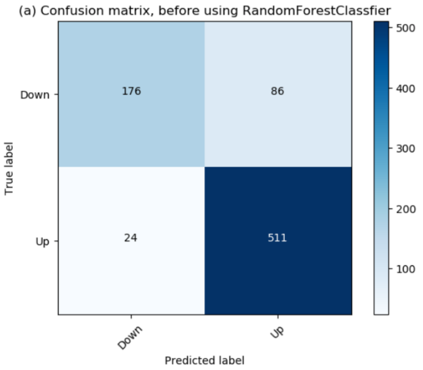
In the table below is the performance for each forecast period, for the extended data-set, shown:

Forecast period:	CV accuracy	Test score	precision	f_1 score	classifier[3]
1 Month	0.65	0.63	0.7038	0.6513	DTC
3 Month	0.70	0.71	0.6969	0.7869	RFC
6 Month	0.77	0.77	0.7519	0.8354	RFC
12 Month	0.84	0.84	0.8328	0.8914	RFC

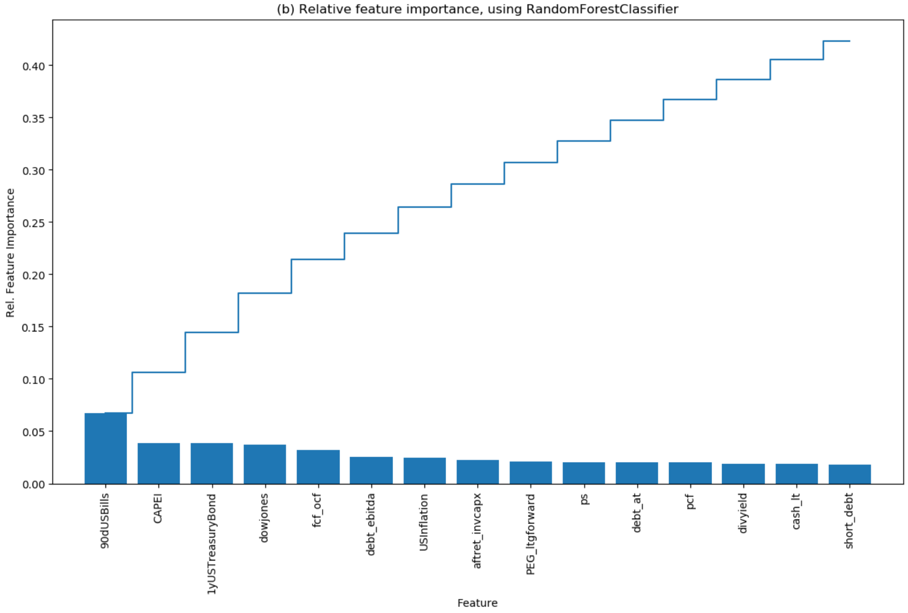
Table 2: Using the extended data-set

Comparing the two tables above there are some similarities between the two of them. By increasing the forecast period the performance enhances rapidly. As the main performance indicator the f_1-score is used. Therefore the focus will be on the 12 month forecast since, in both data-set, these have the best overall performance. Comparing the two data-sets for the forecast period of 12 month we can say that the performance are almost the same. For that reason we will be using the extended data-set for further adjustments. To visualize the performance of our selected data-set and forecast period a confusion matrix is shown in the figure below:

[3] Random Forest Classifier(RFC), Support Vector Classifier(SVC), Decision Tree Classifier(DTC)


(a) Confusion matrix, before using RandomForestClassfier

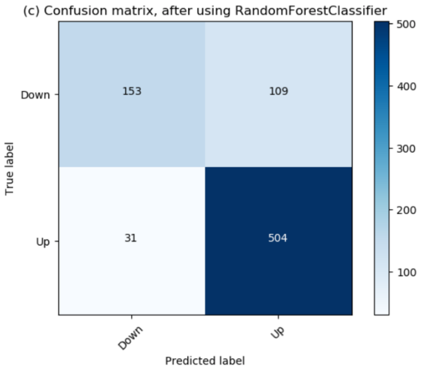
With the help of the RandomForestClassifier we extract the most important features, used in our data-set. In the following figure is the top 15 features which explain most of the variation in our extended data-set:


(b) Relative feature importance, using RandomForestClassifier

In consideration of the importance of the given features we adjusted our data-set further more. In the following table it is shown the performance of the algorithm with the 8 most important features:

Forecast period:	CV accuracy	Test score	precision	f_1 score	classifier[3]
12 Month	0.82	0.82	0.8201	0.8712	RFC

Table 3: Using the extended data-set



(c) Confusion matrix, after using RandomForestClassifier

5 Conclusion

With the help of multiple machine learning algorithms we were able to predict the stock movement (up/down) for the 1, 3, 6 and 12 month forecast period. It can be shown, as seen in the table 1 and 2 above, that the best performance in both data-sets are for a forecast period of 12 month. This makes intuitively sense that it is easier to predict the long-term movements rather than the short-term. For the evaluation of the best data-set we compared the f_1-score of the initial set, which has a score of 0.897 and the extended set, with a score of 0.894. Because the f_1-Score contains the recall and the precision score we took this performance ratio to decide which data-set performed better. Interestingly both of the sets had performed almost similarly. The only difference that crossed our mind was, that the algorithm for the initial set took the SVC and for the extended set the RFC as a classifier, which explains the small difference between the two of them. We decided to go along with the extended data set. By using the RFC we searched for the most important features. We took the eight most important features and performed the algorithm again for the 12 month period. Because further features would not increase the performance, we stayed by those number. The Performance of the reduced data set, which only contain 8 features, is slightly poorer than the one before with a f_1-score of 0.872. Considering the fact that by using a lower amount of features the algorithm needs less computing power and pretty much provide the same result.

Overall we think the classification can be performed quite well with the given data but mostly for a longer forecast period. However over longer periods stock markets tend to go up in general and therefore a passive investing scheme would probably be better than trying to use the predicted stock movements.