**Corporation stock-price forecasting**

**Vuk Vladisavljević**

**1. Motivation**

Stock market trends and share prices are very hard to accurately forecast, considering often fluctuations, unforeseen events and many other financial factors. However, dramatical technology improvements have provided an ability to overcome aforementioned problems, at least to some extent. Such tools and prediction methods can be crucial in maximizing profit and keeping the investors risk as low as possible.

The general hypothesis is that since the factors which determine stock trends in the future are placed in the future, it is impossible to predict those and the effect they will have on share prices. Stock market prediction programs challenge this idea by using available data and trends from company’s past trading to predict future ones, with accuracy to some extent.

Considering that stock market is one of the branches that has most use of machine learning algorithms I wanted to expose some basic principles on which such programs rely on.

**2. Research questions**

Specific problem I wanted to address with is the use of regression to predict stock price trend of Google, Amazon, Apple and Microsoft companies throughout some time and the compare precision of it with regard to the time period. Data on which I will base forecasting on will be the company’s share prices on the stock market and it’s fluctuations over long time.

Several categories of data to use when predicting price exist. They summarize company’s financial history and combined can provide basic knowledge of its success or failure.

These factors are:

1. Sentiment analysis (not included in this assignment)
2. Sales growth
3. Dividends

When summarized, they can show company’s vital statistics and be manipulated to predict circumstances that will affect future stock price and the company’s respond to it. Because of all aforementioned , stock market prices can be forecasted with relative accuracy by extrapolating the graph.

Description of data set and its columns follows.

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Date** | **Open** | **High** | **Low** | **Close** | **Volume** | **Ex-Dividend** | **Split Ratio** | **Adj. Open** | **Adj. High** | **Adj. Low** | **Adj. Close** | **Adj. Volume** |
| 2012-12-31 | 700.0 | 710.57 | 696.0 | 707.38 | 3990800 | 0.0 | 1.0 | 351.08 | 356.3 | 349.07 | 354.78 | 3990800 |
| 2013-01-02 | 719.2 | 727.00 | 716.5 | 723.25 | 5077500 | 0.0 | 1.0 | 360.82 | 364.6 | 359.38 | 362.74 | 5077500 |
| 2013-01-03 | 724.9 | 731.93 | 720.7 | 723.67 | 4631700 | 0.0 | 1.0 | 363.58 | 367.0 | 361.47 | 362.95 | 4631700 |

The following text will explain the meaning of each of the columns:

1. **Date** – day of the values in question
2. **Open** – Stock price at the time of stock marketing opening hours
3. **High** – Highest stock price for that day
4. **Low** – Lowest stock price for that day
5. **Close** - Stock price at the time of stock marketing closing hours
6. **Volume** – number of stocks sold that day
7. **Ex-Dividend** – stock dividend not including the next dividend
8. **Split-Ratio** – way of splitting shares ( 0.0 if none)
9. **Adj. Open** – Adjusted Open stock price that calculates price with regard to dividends, split-ratios and other factors
10. **Adj. High** - Adjusted High stock price that calculates price with regard to dividends, split-ratios and other factors
11. **Adj. Low** - Adjusted High stock price that calculates price with regard to dividends, split-ratios and other factors
12. **Adj. Close** - Adjusted High stock price that calculates price with regard to dividends, split-ratios and other factors
13. **Adj. Volume** - Adjusted High stock price that calculates price with regard to dividends, split-ratios and other factors

**3. Related work**

A lot of authors are trying to implement the best tool for forecasting the stock prices. Unlike my somewhat basic solution that is using regression, some are trying to solve this problem by using artificial neural network, which have a potential of very accurate prediction. Since neural networks represent highly effective method for collecting, analyzing and predicting data, they have been used for solving many different problems, with share prices prediction one of them.

As for their specific use in recent years, one of them is Elman neural network that is processing sequential data and was created for predicting share prices and their fluctuations. The other proposed methodology is on the basis of artificial neural networks, which is able to predict maximum and minimum share prices of company.

**4. Methodology**

This represents typical regression problem (opposed to a classification problem) since we are predicting continuous values and is solved as such.

Problem solution starts with loading WIKI/company\_codename stock data from quandl.com server. Data loaded contains prices from period between **31.12.2007** and **31.12.2017**.

The table provided in the **Research Questions** chapter represents head of dataframe got from quandl server. As we can see, it contains many redundant columns and data that would not enhance forecasting significantly . Some of the columns don’t even change. One of the things obvious from the start is that having both adjusted and regular columns is redundant. Considering that adjusted are more precise and therefore more valuable for our forecasting, we will use those and drop regular ones. Adjusted columns also take share splitting in account so we can also drop that column, Split-Ratio.

In order to make more usefull data than the provided, we will create two new columns:

1. Average\_price – that represents average price for the day, taking Adj. High and Adj. Low in account
2. Change\_percentage that represents daily volatility, measured by Adj. Close and Adj. Open

By now, we are supposed to finish with data manipulation and we can continue onto regression.

In supervised learning, you have features, which are descriptive attributes supposed to make prediction available, and labels that represent value you are trying to predict.

Since there isn’t test\_data, we must create new one that will contain data to be compared with predicted values. Such column will be created by shifting the **average\_price** column so that the new one contains average\_price value for the predefined time in future.

The features that contribute to the precision the most will be chosen with SelectKBest with **mutual\_info\_regression** function and will yield 5 best columns.

Next step is to make X, which represents features, and y which represents labels in order to train regression method. X will contain all data except the data contained in **predicted\_price** column which will be contained in y. After declaring those two sets, we must normalize values, which I have chosen to do by MinMax normalization, using sklearn MinMaxScaler.

In order to get as good X and y train and test sets, cross\_validation (model\_selection.train\_test\_split) is used, with the 20% size of test set.

After the training and test data is formed, PCA will be used to reduce dimensionality of the data.

Finally, it is time to define and use method to predict prices. After thorough testing, the best results came with using Lasso regression with tolerance 0.01 and max iterations set to 10 000.

**5. Discussion**

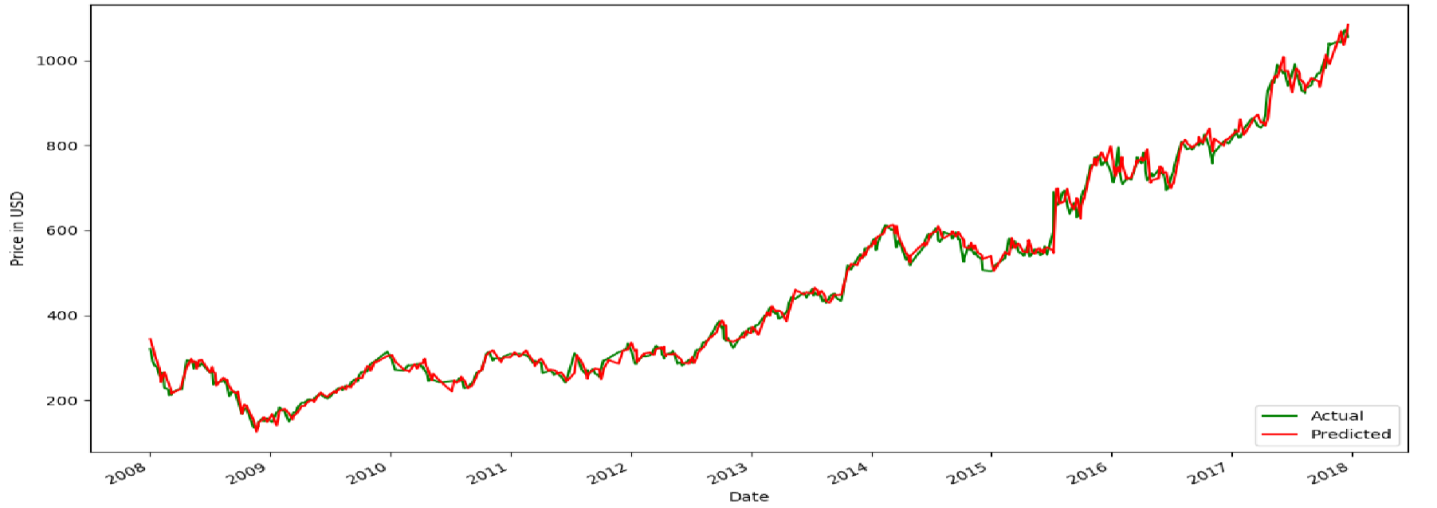
Forecasting performance will be measured with **Root Mean Squared Error (RMSE)** by which it will be possible to calculate forecasting precision. It represents the error between the actual and predicted price. Many different regressors were tested, with each of them giving similar results. Testing values in the following table resulted from 7-days forecast. 

Figure 1 7-days forecast graph

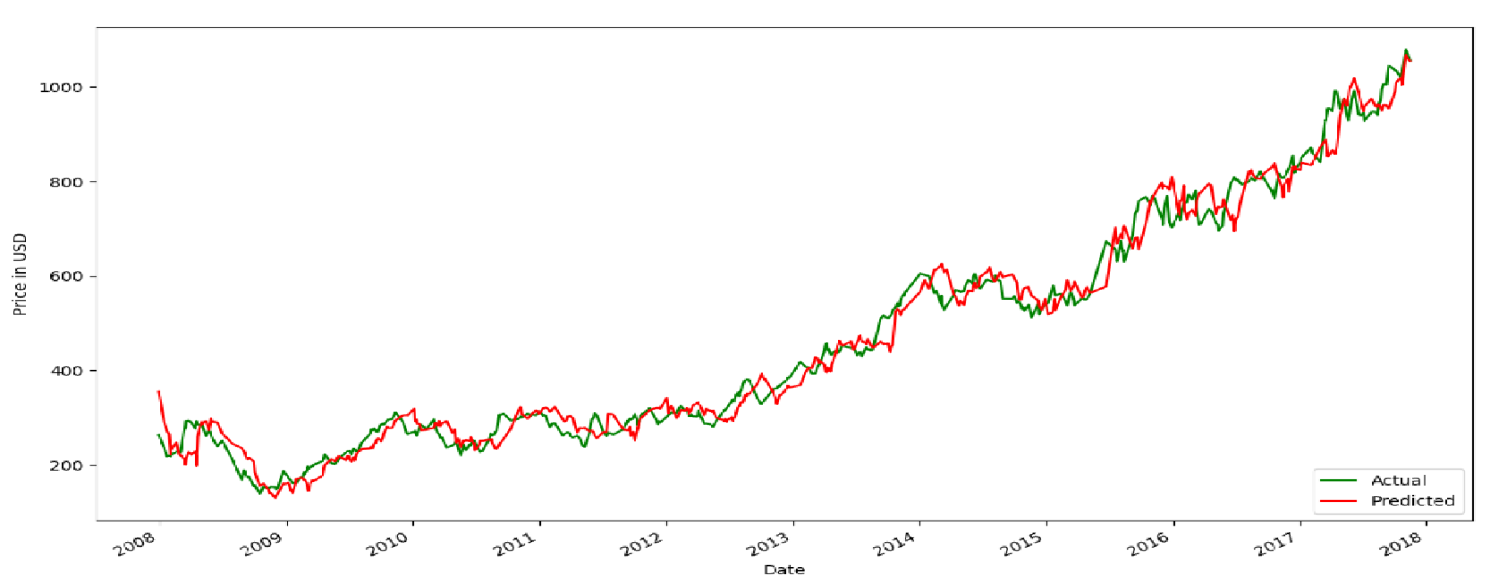
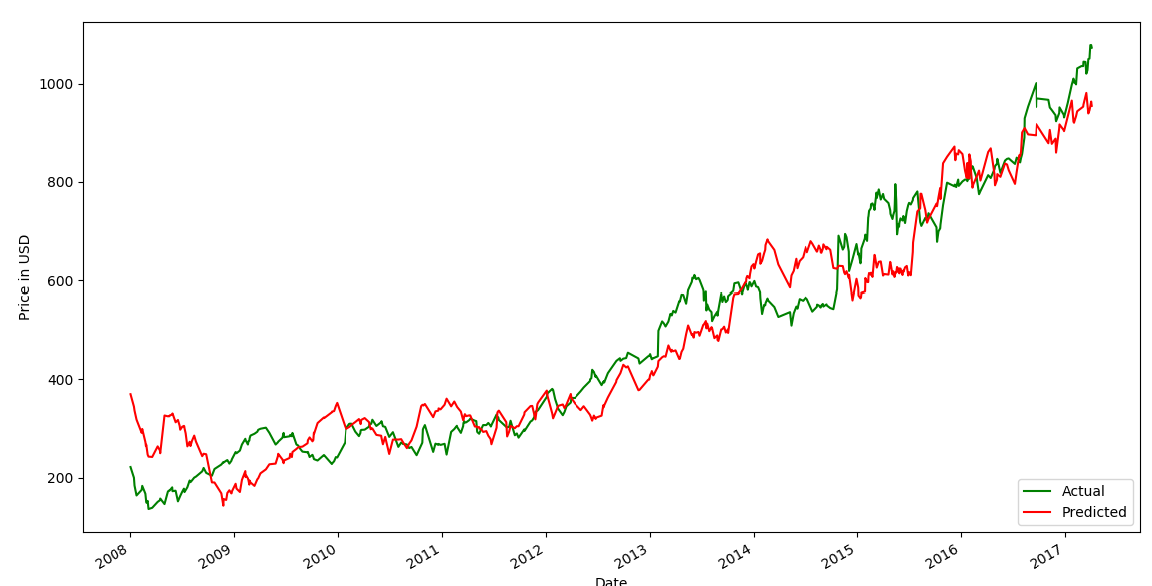
When the longer period is tried, results differ drastically. Following table and graph represent state when period being forecasted is 30 days. 

Figure 2 30-days forecast

Considering that all the methods gave somewhat same results, we can see that for this problem, data provided is of the bigger importance, and only by improving it, we would improve the testing accuracy as well.



**6. References**

<https://medium.com/mlreview/a-simple-deep-learning-model-for-stock-price-prediction-using-tensorflow-30505541d877>

<https://towardsdatascience.com/just-another-ai-trying-to-predict-the-stock-market-part-1-88605f9d8e45>

<https://www.quora.com/Can-machine-learning-algorithms-models-predict-the-stock-prices-If-yes-which-are-the-best-machine-learning-algorithm-models-to-predict-the-stock-prices>