DEGREE PROJECT, IN COMPUTER SCIENCE , FIRST LEVEL
*STOCKHOLM*, *SWEDEN 2015*

# Testing Stock Market Efficiency Using Historical Trading Data and Machine Learning

SAMI PURMONEN & PAUL GRIFFIN

**KTH ROYAL INSTITUTE OF TECHNOLOGY**

**CSC SCHOOL**


KTH Computer Science
and Communication

# Testing Stock Market Efficiency Using Historical Trading Data and Machine Learning

SAMI PURMONEN
PAUL GRIFFIN

Degree Project in Computer Science, DD143X, KTH, CSC
Supervisor: Alex Kozlov
Examiner: Örjan Ekeberg

# Abstract

Stock forecasting is a problem that is important in finance because it aids investors in financial decision making. According to the efficient market hypothesis stock markets are efficient in such a way that it's impossible to gain excess returns over the market by making decisions based on current available information. This paper evaluates the usage of machine learning algorithms and historical trading data for stock price prediction combined with investment strategies in order to test the efficient market hypothesis. The results show that none of the tested machine learning algorithms managed to gain excess returns over the market which confirms the efficient market hypothesis.

# Contents

# Chapter 1

# Introduction

In the stock market shares of companies are traded. It's an important component of the economy because it gives companies access to capital in exchange for a slice of ownership. Investors can speculate in stock prices, buy when they think prices are going up and sell when they think prices are going down. This would be profitable for investors making accurate predictions of stock prices. However, the predictability of stock prices has been questioned. According to the efficient market hypothesis stock prices are unpredictable in such a way that one can not beat the market [4]. In this paper we attempt to predict prices using machine learning algorithms. The predicted prices are then used with an investment strategy and its profits are compared to market returns. If the investment strategy beats the market the efficient market hypothesis is falsified.

## 1.1 Problem Statement

The goal of this paper is to investigate whether the stock market is predictable and if the efficient market hypothesis is true. These questions are both contained in the question: is the efficient market hypothesis true?

## 1.2 Problem Constraints

The main constraint is that only historical trading data is used for predictions. There's a large amount of variables that could be of affect stock prices such as macroeconomic variables or opinions on social media such as Twitter.

# Chapter 2

# Background

## 2.1 Efficient market hypothesis (EMH)

According to the EMH the stock market follows a random walk and is inherently unpredictable [7]. The EMH states that all information available today is reflected in prices of stocks today and tomorrows price changes will only depend on tomorrows news. The reason for this is that the stock market is perceived to be highly competitive and informationally efficient. Since news by definition is unpredictable, stock forecasting is impossible. This would have the consequence that prices are fairly priced and one can not beat the market using current information. Makiel claims that expert investors being unable to beat the market is a strong argument for the market being efficient and unpredictable [7].

The EMH comes in three forms based on the strengths of their assumptions about what information is reflected in stock prices. The weak-form EMH states that historical trading data cannot be used in order to gain excess returns on the stock market which would make technical analysis useless. The semi-strong form EMH states that all publicly available information is already reflected in prices and makes fundamental analysis useless. The strong-form EMH states that all information, even inside information, is already reflected in prices which means that no information can give the investor consistent excess returns on the stock market.

## 2.2 Machine Learning (ML)

ML is a set of techniques that attempt to let a computer recognize patterns. The goal is for the computer to learn how a system behaves so it can fill in missing data, predict data or categorize data. Machine learning can be implemented in many different ways but all implementations have to have some method of training. Training can either be:

- **Supervised** - The algorithm is provided with test cases and the correct answer for those cases.

- **Unsupervised** - The algorithm is not provided with correct answers, this works for example when dividing a data set into groups and not knowing what groups exist.

- **Reinforcement** - The algorithm only gets the signal success of failure. The task could be long and complicated, for example when playing a game of chess the algorithm does not get feedback every move, but only get to know if the whole game was a victory or defeat.

### 2.2.1 Artificial Neural Network (ANN)

An ANN is software network inspired by the human brain. It is capable of learning. Given a set of inputs and outputs it can be trained to recognize relationships and then generalize to unseen inputs. An ANN consists of several layers. First there is the input layer with one node per input value. This layer is connected to a hidden layer with an arbitrary number of nodes. The hidden layer is finally connected to the output layer that produces the result. Depending on the number of hidden layers and hidden nodes, the neural network has different capabilities. The network can be train for example with back propagation [2].

### 2.2.2 Linear Regression

Linear regression in one dimension is the process of fitting a line while minimizing the distance between the points to fit to and the line. In more dimensions a hyperplane is used instead.

### 2.2.3 Random Forest

Random forest uses many decision trees and takes the average output of all the trees as the result. Each tree is trained on a random subset of the input data. This removes the issue of over training and allows for more features for a smaller performance cost [5].

Decision trees are flowcharts that describe outcome probabilities depending on given variables. The problem of creating such trees optimally is NP complete [6] but approximate methods exist. One common method is induction of decision. With this method the input set is divided based on some output value being above a random threshold. Then a input variable is chosen and a threshold set so that as many values are classified correctly as possible. The process continues recursively until all the data is classified correctly or no improvement can be found [10].

### 2.2.4 Nearest Neighbors

Nearest neighbor works by saving all training patterns and target values during training. For classification or prediction the input pattern is compared to all training

patterns and the closest k samples are chosen and maybe weighted. The average of those is the result [3].

## 2.3 Related work

Lots of research has been done on neural networks and stock forecasting.

Suljkic & Molin [11] tested three different neural networks, NARX, AMARX and ANFIS for stock forecasting. They reported ANFIS as the most successful. Suljkic & Molin [11] also suggests that the stock market is predictable. However, they did not calculate investment profits so it's unknown whether they could beat the market.

Meier & Olsson [8] researched the optimal number of training days for a multi layer perceptron using stock forecasting performance as a benchmark. They additionally suggests that they were able to beat the stock market index OMX Stockholm 30 suggesting that their predictions were successful. However, they did not consider fees which limits their findings.

Tao and Lim [12] suggests that looking at the mean average error is not enough to determine the quality of a stock prediction and that profit made using the algorithm, and direction of prediction also should be taken into account.

# Chapter 3

# Method

At a high level the HD-model is used.

1. **Hypothesis** - The EMH is true
2. **Implications** - It's impossible to beat the market
3. **Test** - Different investment strategies are tested. If one that beats the market is found then the EMH is falsified. Otherwise it is confirmed.

Proving that EMH is true is a problem that has been attempted by many researchers but there is no definitive consensus. This paper presents an attempt at disproving EMH by looking at a necessary consequence and attempting to falsify it.

The EMH states that it is not possible to consistently beat the market. This means that if one can find an investment strategy that consistently produces excess returns compared to market indexes, the EMH would not be true.

The method used in this paper is based on price predictions. The first step is predicting prices using historical trading data. If historical trading data is related to future prices it would be hard to find out how they are related manually since it may be complex non-linear relationships. Therefore ML algorithms are used which has been shown to be good at finding such relationships if they exist [9]. The second step is to invest money based on the predictions and evaluate the returns compared with the market. If the market is outperformed, EMH must be falsified.

## 3.1 ML algorithms

Four different ML algorithms are used. Artificial neural network, linear regression, nearest neighbors and random forest. The implementation of each algorithm comes from Mathematica v10.0.1.0. This ensures that the implementations are well tested since they are used by many people. All the ML algorithms used in the paper use supervised learning and train on 90 days data prior to the predicted date. Input to the algorithm after training is the past 5 days data.

## 3.2 Naive algorithm

In order to gain insight into the ML algorithms ability to predict a naive algorithm is included as a benchmark. The naive algorithm simply predicts that the price for a certain day is the same as the price of the previous day. If the ML algorithms have any ability at all to predict prices it's reasonable to believe that they would be able to beat the naive algorithm.

## 3.3 Test Data

The data set comes from Standard & Poor's 500 (S&P 500) which is a stock market index tracking the 500 companies with the largest market capitalization in the USA and The Financial Times Stock Exchange (FTSE 100) which is a stock market index tracking the 100 companies with the largest market capitalization on the London Stock exchange. The data set consists of closing, opening, volume, low and high prices of each day. This index was chosen because it is commonly used in other scientific reports to make the results as comparable as possible.

## 3.4 Investment Strategy

The investment strategy is to invest all money in in the index when the price is predicted to rise and sell all shares when the price is predicted to fall. Details of the algorithm are displayed in pseudo code below.

```
money = 1
for t in 2...predictedPrices.length
    if predictedPrices[t] >= realPrices[t-1]
        money = money * realPrices[t] / realPrices[t-1]
```

In the real world one must also take into account the fees for buying and selling. The investment strategy is evaluated with and without fees. The transaction fees used is 0.12% at Avanza [1]. Details of the algorithm are displayed in pseudo code below.

```
money = 1
hasInvested = true
for t=2:length(predictedPrices)
    transactionCost = money * transactionFee
    if hasInvested
        if predictedPrices(t) < realPrices(t-1) -
           transactionCost
            money = money - transactionCost
            hasInvest = false
        else
            money = money * realPrices(t)/ realPrices(t-1)
```

```
else
    if predictedPrices(t) - transactionCost >=
       realPrices(t-1)
        money = (money-transactionCost) * realPrices(t)
            / realPrices(t-1)
        hasInvested = true
```

The benchmark strategy is to invest all money on the first day and sell all shares on the last day of the prediction period. This is called the buy-and-hold strategy. The money made using this method is:

```
money = realPrices[end] / realPrices[1]
```

## 3.5 Performance measures

In order to measure the performance of the predictions the following error measures are used.

### 3.5.1 Absolute Percentage Error (MAPE)

$\frac{1}{n}\sum_{i=1}^{n}\left|\frac{predictedPrices[i]}{realPrices[i]} - 1\right|$

This measurement can intuitively be seen as the difference between the real values and predicted values but changed to remove influence from number of samples and the stock value. The advantage of this performance measure is that it is invariant to the number of predictions made and the stock value. This makes it easy to compare with other reports. It does not, however, represent the profit one would make if applying the methods with real money.

### 3.5.2 Movement accuracy

This measurement is the percentage of predicted movements that are correct. If nothing is known about the future stock prices one would expect this value to be 50%. This means that values below 50% indicate that the algorithm measured is performing poorly.

### 3.5.3 Investment profit

This value represents using the output of one of the predictors to buy stocks. So if the predictor suggests that the stock price will increase all money is invested into that stock. If the opposite happens all money is withdrawn. This gives an estimate of the return on investment using the predictors. It does not take into account differences between buy prices and sell prices, commission and fees. This means the value is not directly useful for buying stocks but provides a value that can compare predictors.

### 3.5.4 Investment profit without fees

This value is the same investment profit except it also considers transaction fees. This is the most realistic value performance measure since it models the real fees when investing in stocks.

# Chapter 4

# Results

## 4.1 S&P 500

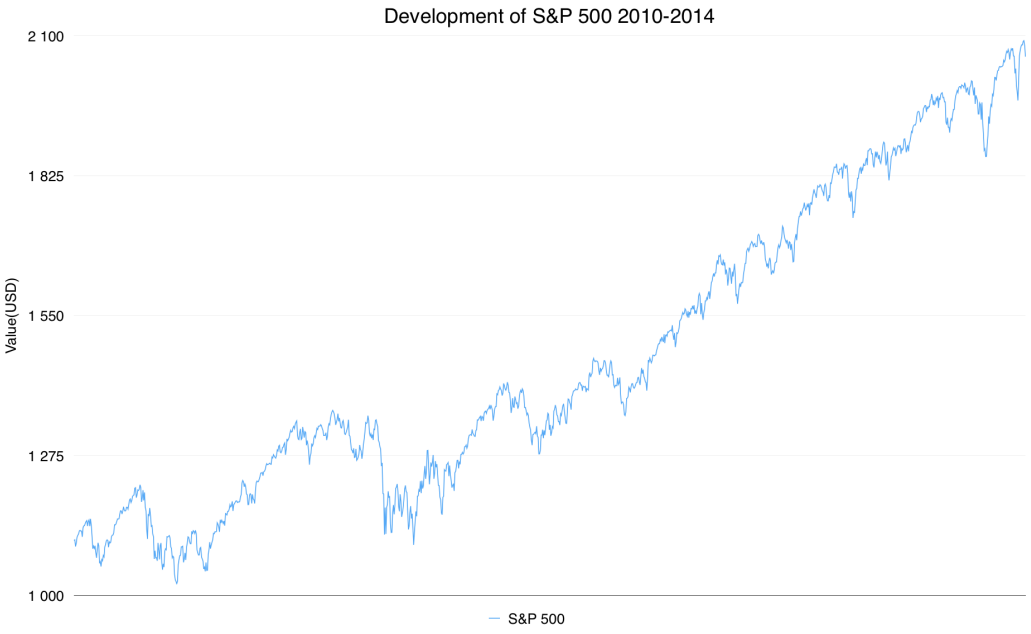


**Figure 4.1.** S&P 500


**Figure 4.2.** S&P 500 and predicted prices zoomed in

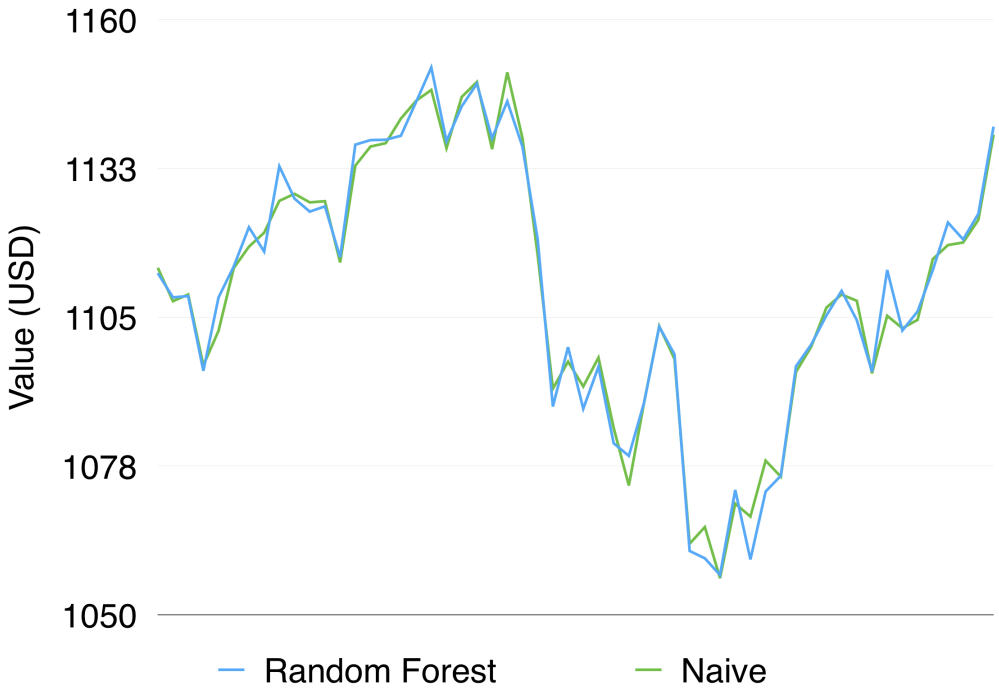


**Figure 4.3.** S&P 500 and predicted prices zoomed in

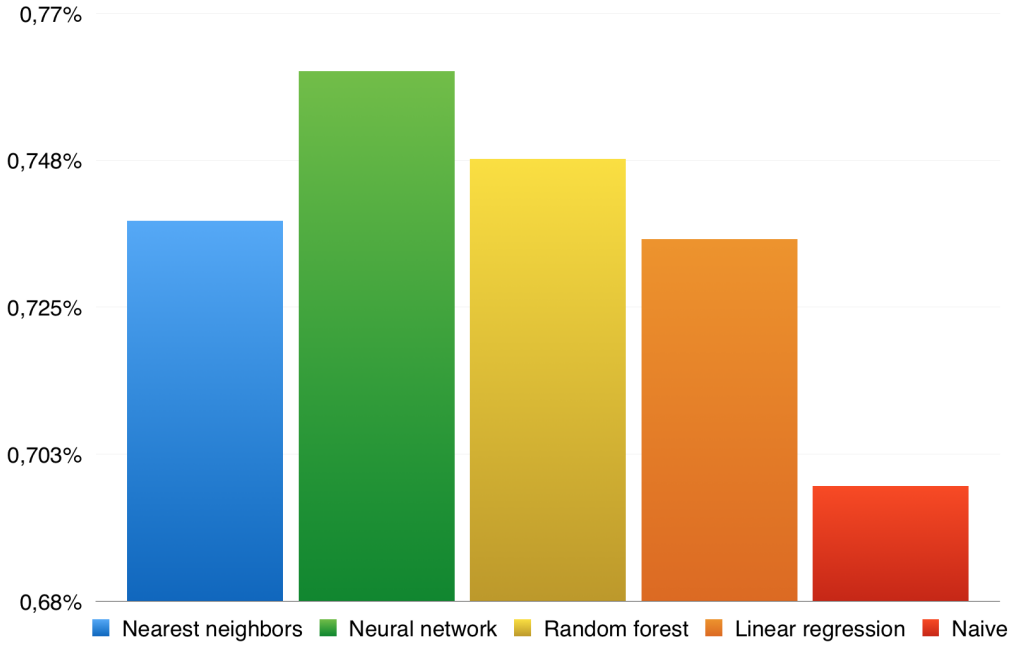


**Figure 4.4.** S&P 500 MAPE for each algorithm

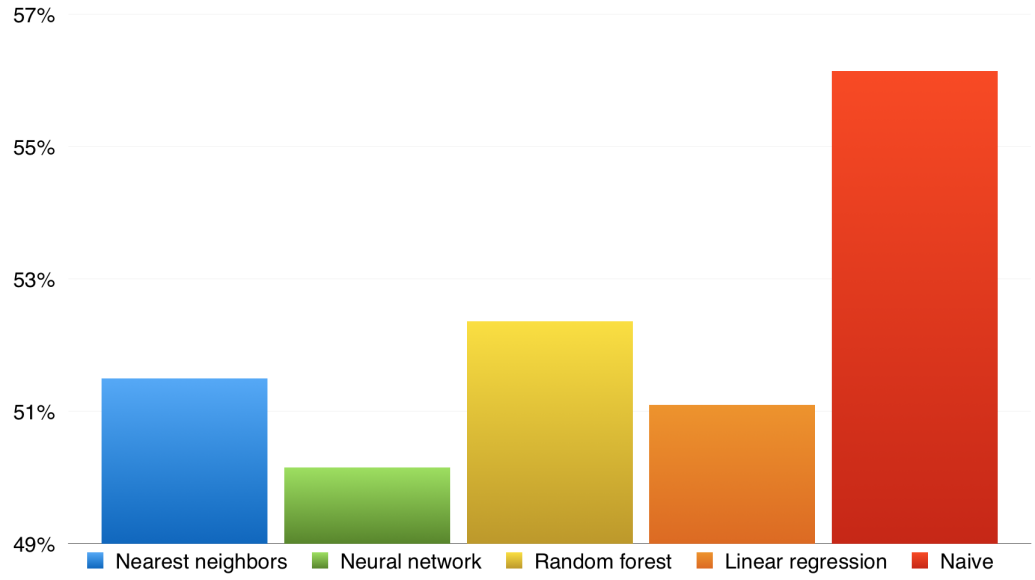


**Figure 4.5.** S&P 500 Movement accuracy for each algorithm

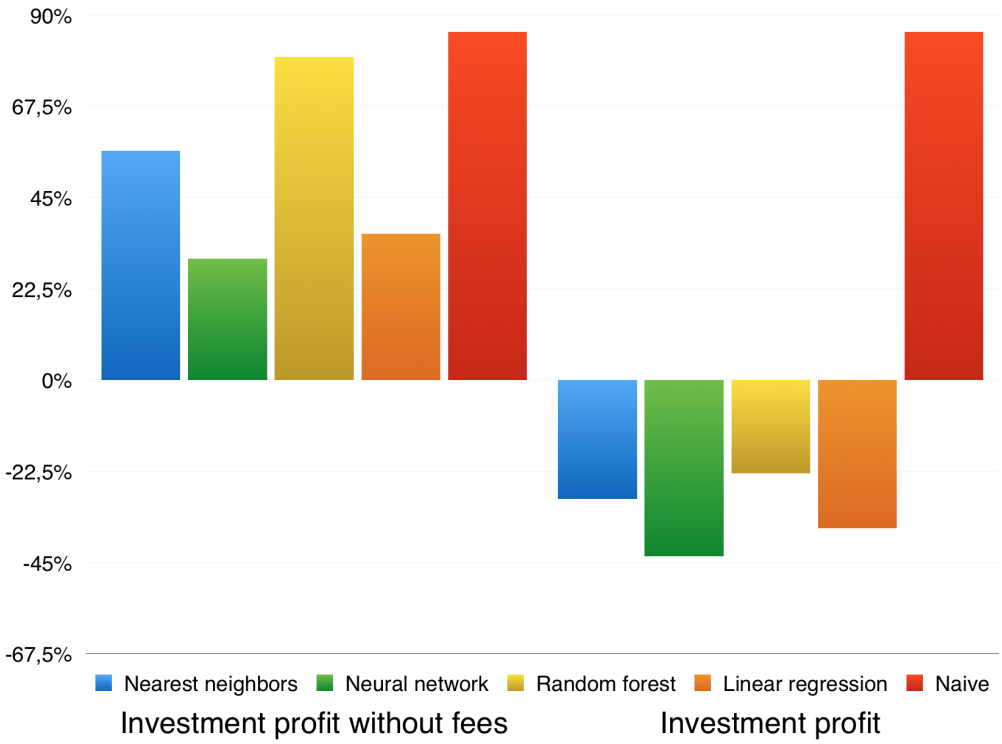


**Figure 4.6.** S&P 500 Profits with and without fees for each algorithm

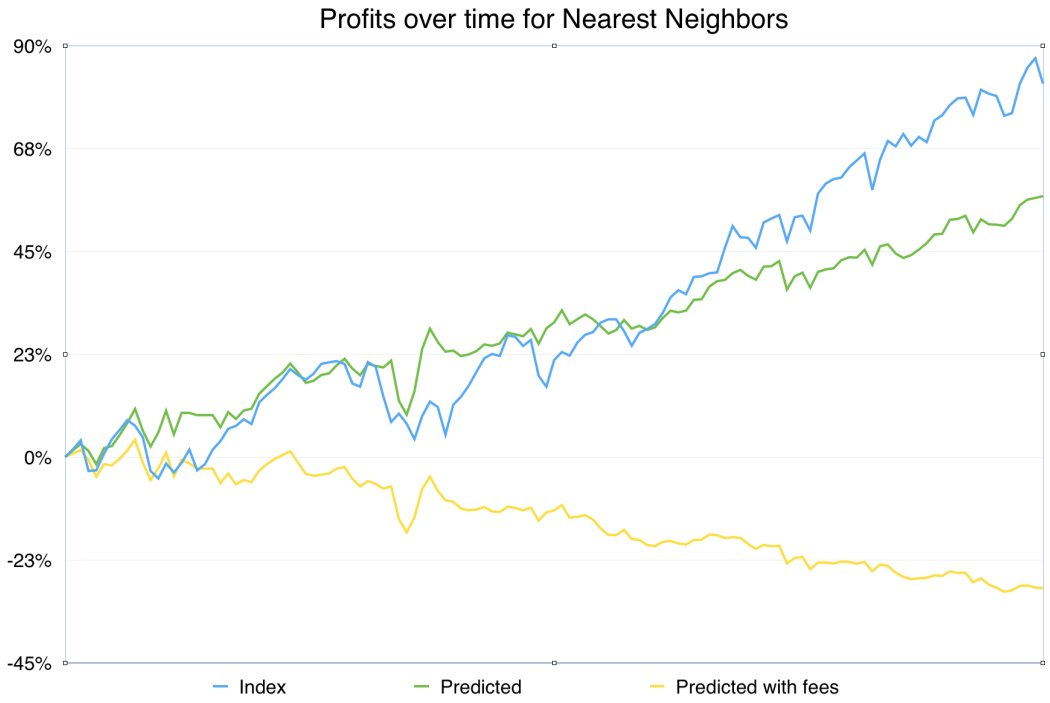


**Figure 4.7.** S&P 500 Profits for nearest neighbor

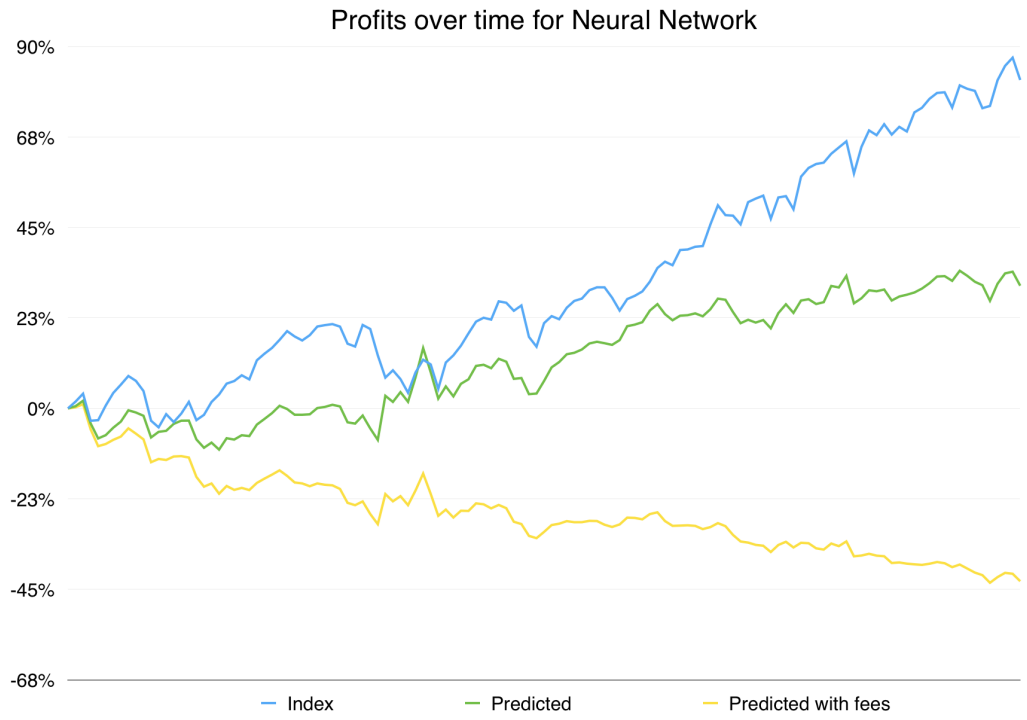


**Figure 4.8.** S&P 500 Profits for neural network

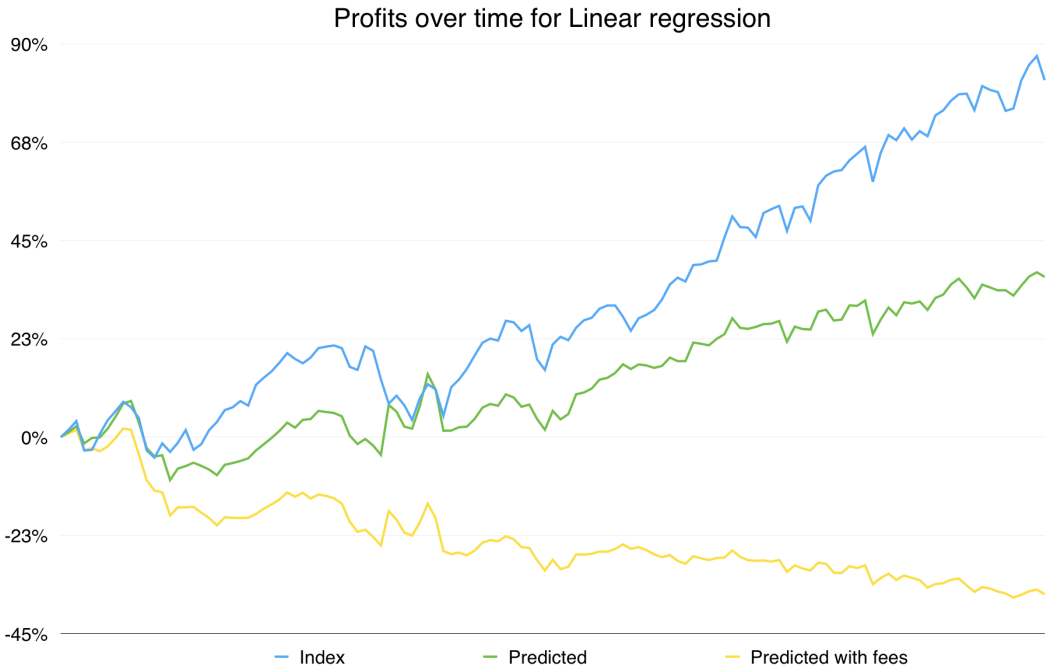


**Figure 4.9.** S&P 500 Profits for linear regression

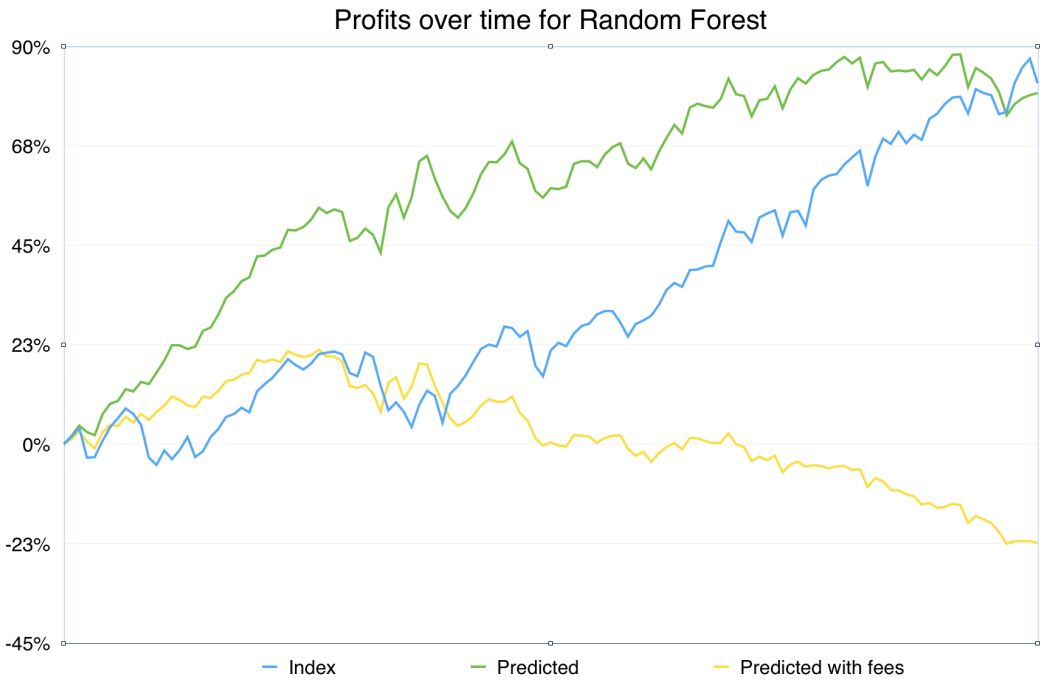


**Figure 4.10.** S&P 500 Profits for random forest

## 4.2 FTSE

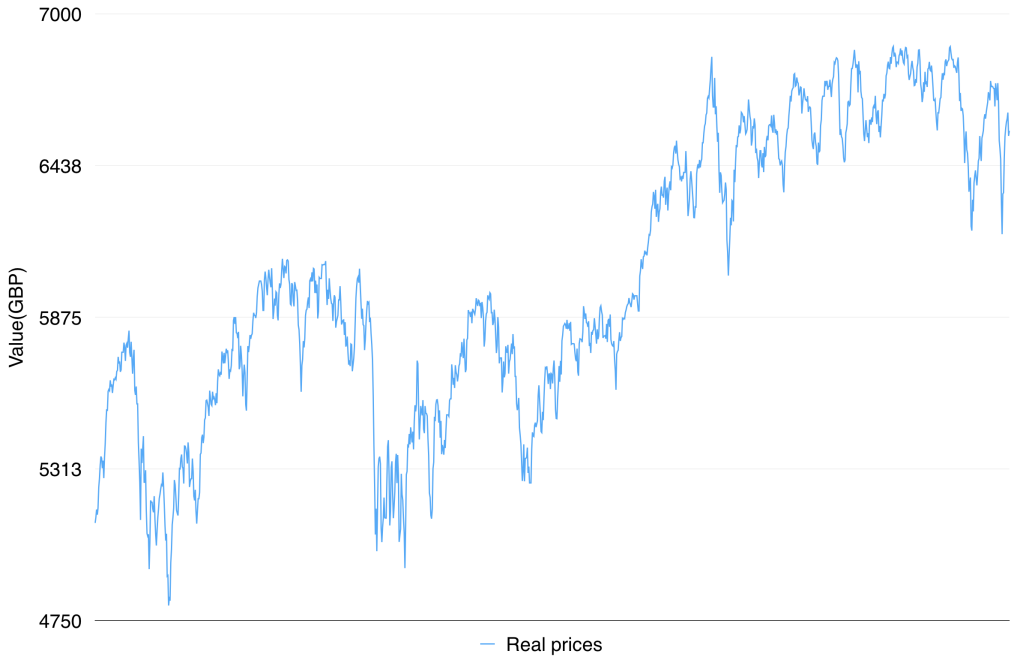


**Figure 4.11.** FTSE 500


**Figure 4.12.** FTSE and predicted prices zoomed in

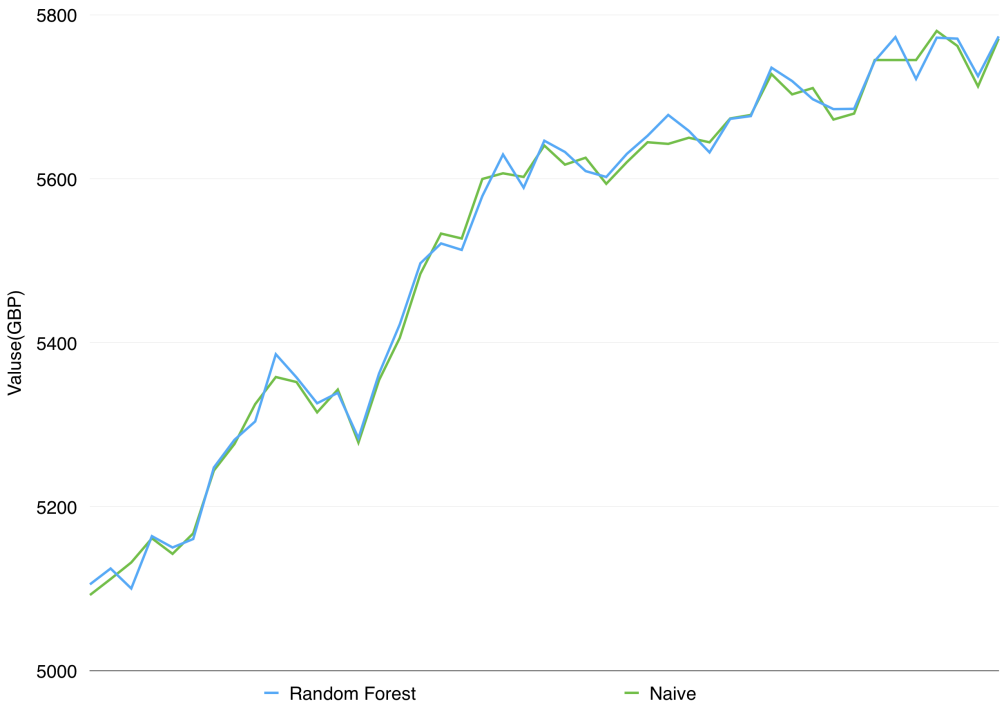


**Figure 4.13.** FTSE and predicted prices zoomed in

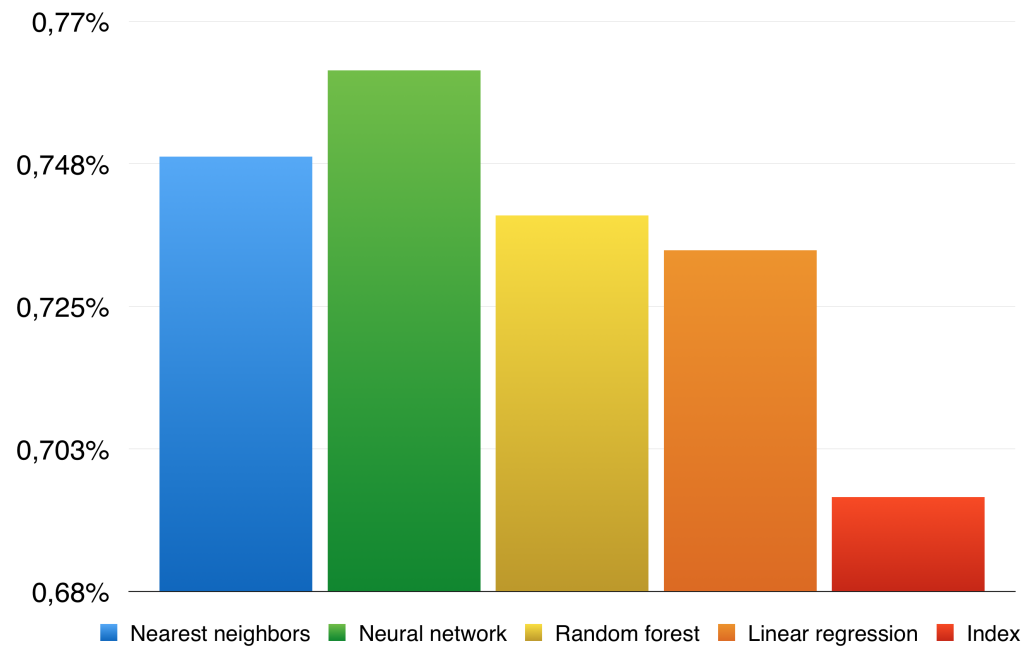


**Figure 4.14.** FTSE MAPE for each algorithm

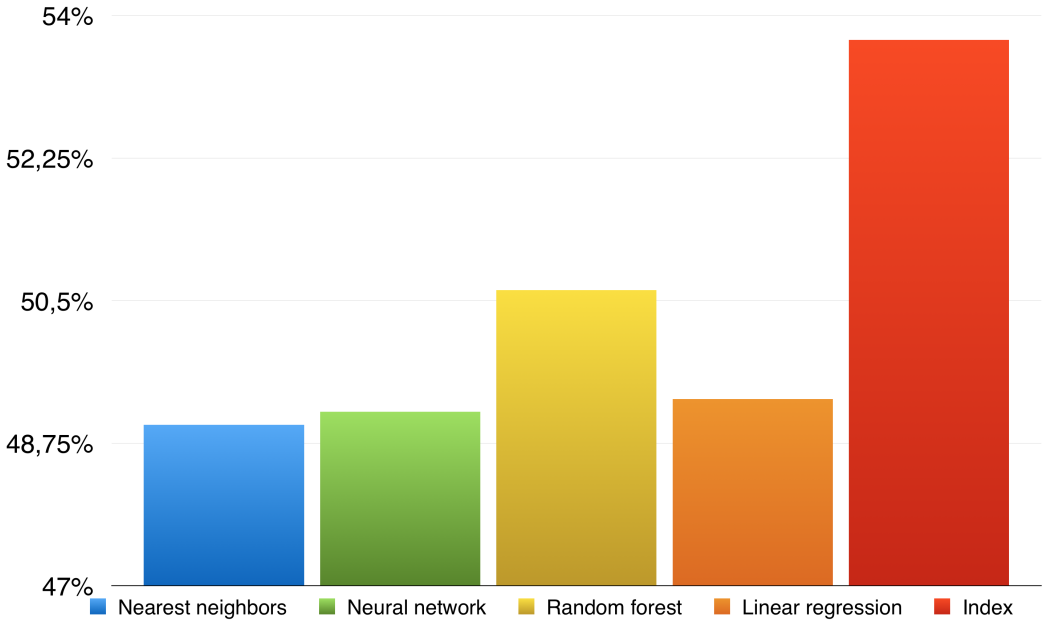


**Figure 4.15.** FTSE Movement accuracy for each algorithm


**Figure 4.16.** FTSE Profits with and without fees for each algorithm

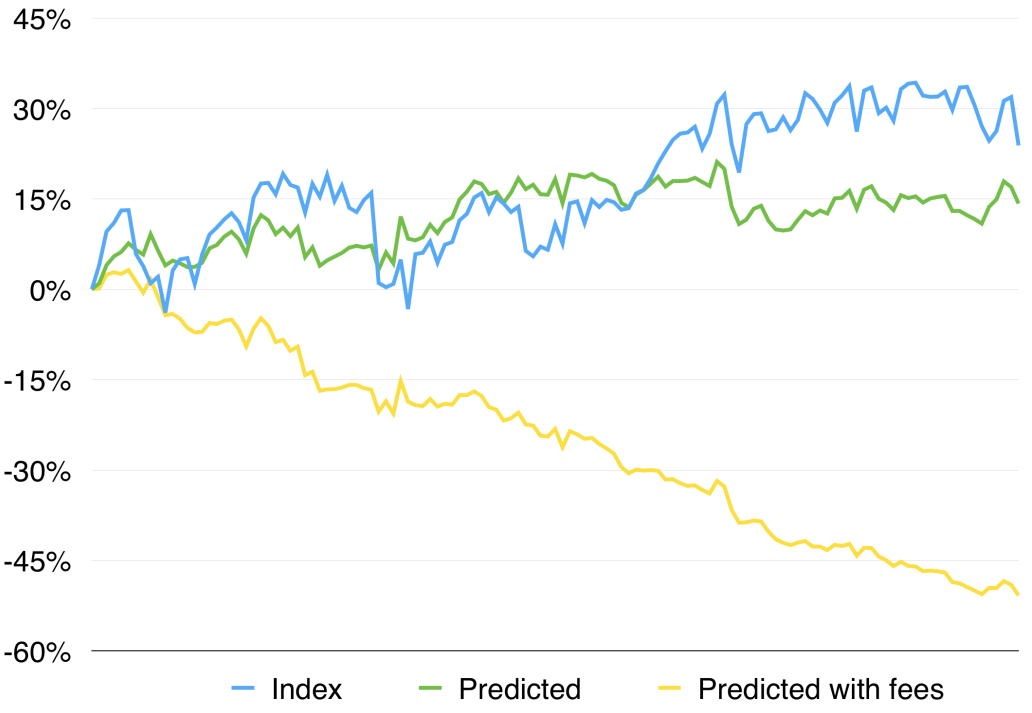


**Figure 4.17.** FTSE Profits for nearest neighbor

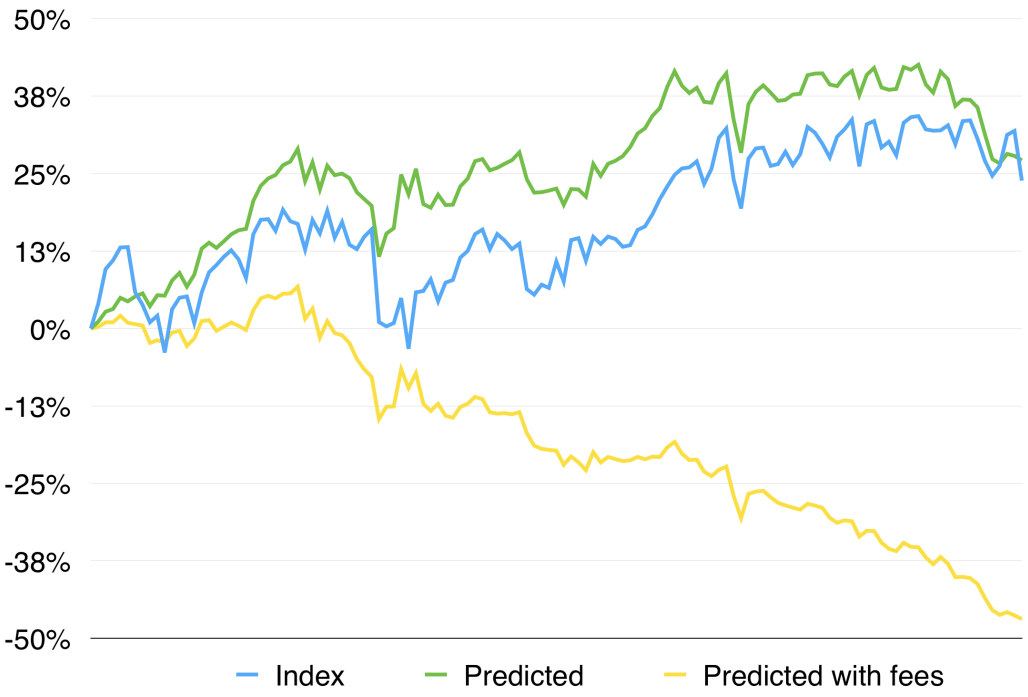


**Figure 4.18.** FTSE Profits for neural network

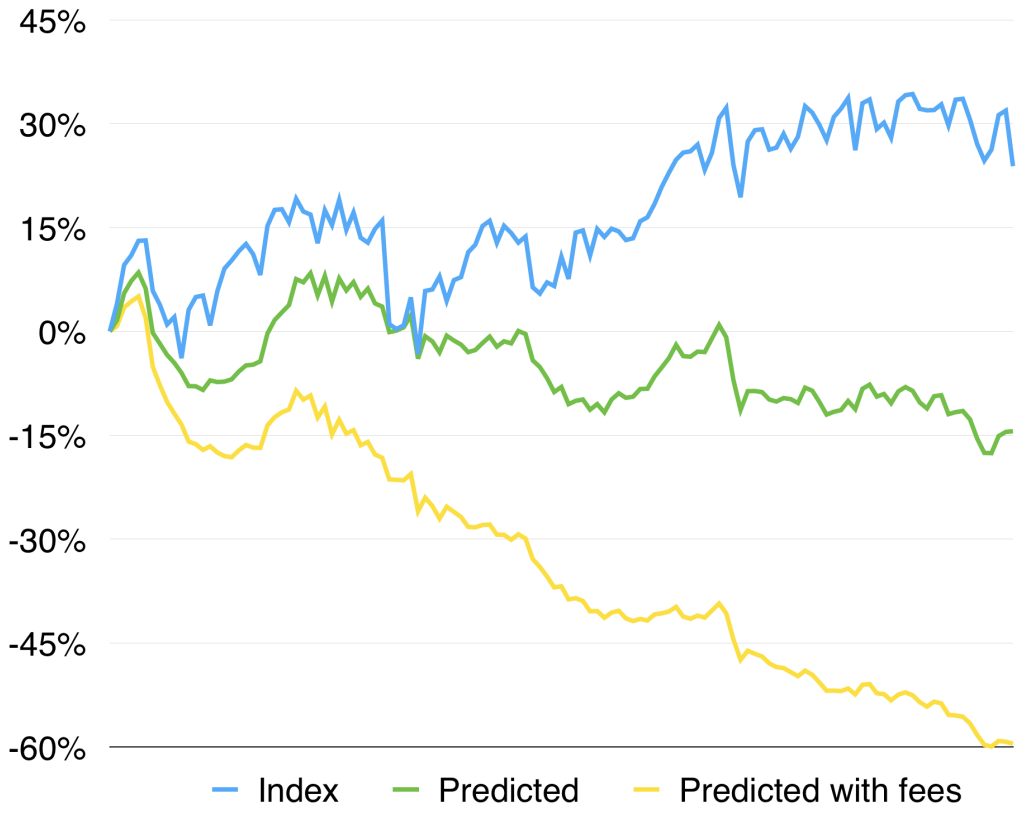


**Figure 4.19.** FTSE Profits for linear regression

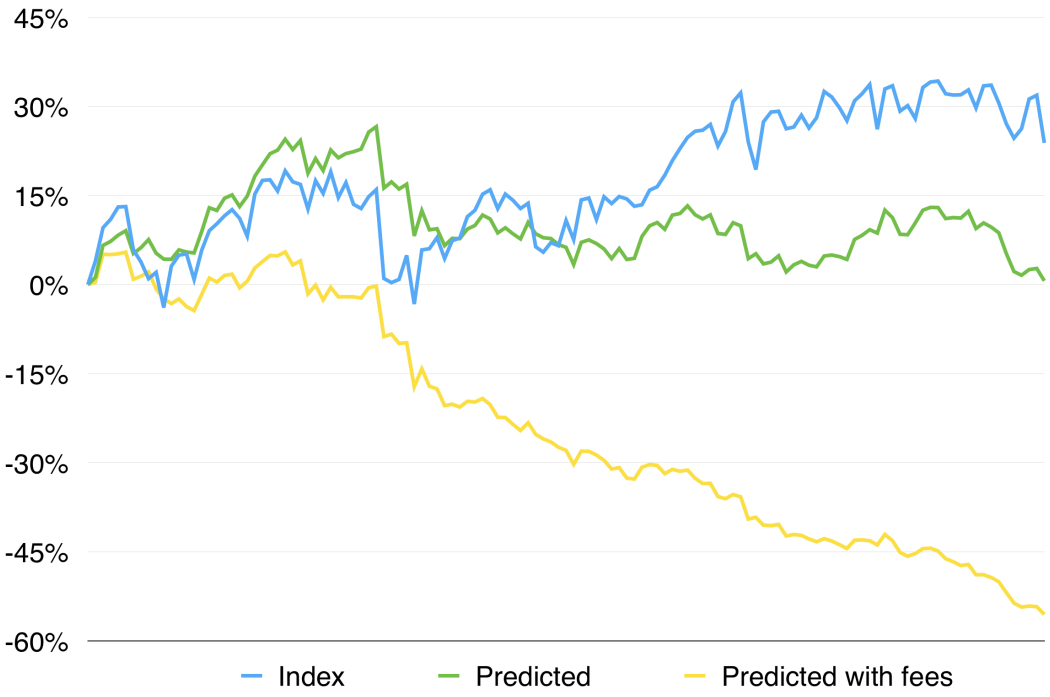


**Figure 4.20.** FTSE Profits for random forest

# Chapter 5

# Discussion

Several interesting things were discovered when looking at the results. A general look at the indices is available in figure 4.1 and 4.11 and more details are discussed in this section.

## 5.1 Investment Profits

As seen in figure 4.6 all algorithms produce positive returns on S&P 500 and in figure 4.16 we see that all methods except linear regressions produce positive returns on FTSE 100. However, the naive algorithm which is following the index performed better than any ML algorithm on both indices. This means that the results support the EMH. Further, when adding transaction fees all methods produced negative returns on both indices. It might be possible to get smaller fees than Avanza, but since the ML algorithms underperformed the market without fees there's no point in further investing that.

In the graphs of profit over time the large impact of fees can be seen. For all ML algorithms on the S&P 500 except random forest which was the best performing algorithm, seen in figure 4.10, the return with fees is almost always negative. In figure 4.10 the return with fees actually outperform the market during a short periods of time. However, during most part it's negative as well and in the end it severely under performed the index. This could be taken as a sign that EMH is false, but since the trades do not take place under the same circumstances as every one else trades this would be a false conclusion. It does however suggest that trying to minimize transaction fees leads to better results, and finding a way to get lower fees than other market participants could lead to beating the market. As seen in figure 4.7, 4.8 and 4.9 the neural network, nearest neighbors and linear regression all produced similar results with the neural network being the worst.

When looking at the results on FTSE 100 we see that the neural network in figure 4.18 is the best performing algorithm. This is interesting because it means that the algorithm that performed best on one index is not the same performing best on another. This makes it hard to make conclusions whether any of the algorithms

actually is better than the other algorithms.

As seen in figure 4.17, 4.20 and 4.19 the random forest, nearest neighbors and linear regression all produced similar results with the linear regression being the worst.

## 5.2 MAPE

As seen in figure 4.4 and 4.14 all algorithms produce errors that are less than 1%. Without looking at other measurements this may seem low and suggest that predictions are accurate. However, the naive algorithm outperforms all ML algorithms suggesting that the ML algorithms ability to predict prices is very low.

## 5.3 Local Predictor Behavior

As seen in figure 4.2 and 4.12 the predicted value seems to lag one day behind the real value. This indicates that the Random Tree algorithm did not find a solution that differs much from the naive solution. In figure 4.3 and 4.13 it can be seen that Random Forest is much closer to the Naive solution than to the actual values. There are however some differences between the naive curve and the Random Forest curve, since this increases the error as displayed in figure 4.4 and 4.14 the changes are probably not beneficial.

## 5.4 Movement Accuracy

As can be seen in figure 4.5, all ML algorithms had slightly better movement accuracy than 50% with random forest being the best at 52%. This may suggest some predictability of stocks since accuracy was better than 50%. However, the naive approach of just guessing that the movement is going to be positive would give 56% accuracy which means that the stock index in general is going up. It's in general true that the market over time is going up so the most naive approach would outperform all ML algorithms in this aspect which means that movement accuracy greater than 50% is not enough to exploit the market.

Looking at FTSE in figure 4.15 three of the four prediction methods have a movement accuracy below 50%. The variance in the numbers between indices suggests that the methods chosen can not cope with general data from different indices and reinforces the EMH.

It is also worth noting an algorithm with a low movement accuracy could still be good since movement accuracy does not take into account how wrong the prediction is. An algorithm mispredicts small changes but gets the large movements correct could be wrong often, but right when it counts.

## 5.5 Conclusion

None of the ML algorithms managed to beat the market on either index which means that the results support the EMH.

# Bibliography


[1] Courtage via internet eller handelsapplikation. `https://www.avanza.se/pro/prislista/handel.html`. Accessed: 2015-05-11.

[2] ACC Coolen. A beginner's guide to the mathematics of neural networks. In *Concepts for Neural Networks*, pages 13–70. Springer, 1998.

[3] Thomas Cover and Peter Hart. Nearest neighbor pattern classification. *Information Theory, IEEE Transactions on*, 13(1):21–27, 1967.

[4] Eugene F. Fama. Efficient capital markets: A review of theory and empirical work*. *The Journal of Finance*, 25(2):383–417, 1970.

[5] Tin Kam Ho. Random decision forests. In *Document Analysis and Recognition, 1995., Proceedings of the Third International Conference on*, volume 1, pages 278–282. IEEE, 1995.

[6] Laurent Hyafil and Ronald L Rivest. Constructing optimal binary decision trees is np-complete. *Information Processing Letters*, 5(1):15–17, 1976.

[7] Burton G. Malkiel. The efficient market hypothesis and its critics. *Journal of Economic Perspectives*, 17(1):59–82, 2003.

[8] A. Meier and P. Olsson. The optimal training interval for a multilayer perceptron on a day to day estimation of the swedish omxs30 index. 2014.

[9] Donald Michie, David J Spiegelhalter, and Charles C Taylor. *Machine learning, neural and statistical classification*. Citeseer, 1994.

[10] J. Ross Quinlan. Induction of decision trees. *Machine learning*, 1(1):81–106, 1986.

[11] J. Suljkic and E. Molin. Testing the predictability of stock markets on real data. 2014.

[12] Jingtao Yao and Chew Lim Tan. A study on training criteria for financial time series forecasting. In *Proceedings of International Conference on Neural Information Processing*. Citeseer, 2001.