Predicting S&P500 ETF Returns Using Non-Price Information

Toavina Andriamanerasoa

12 April 2016

Contents

1 Introduction - Background

Stock markets hold a special place in many countries, acting as a gauge of the health of an economy and investor sentiment, and impacting billions of people's wealth.

Most weekdays speculators, investors and other market participants take positions in the market that reflect their views, their emotions, their willingess to take risks, all with the hope of becoming richer. As those opinions are aggregated and matched, prices fluctuate.

On of the most important market indices, the S&P 500 index, represents 500 of the largest companies in the US equities market which total c.80% of available market capitalization. It is a widely used benchmark for many US equity funds, and with the rise of ETFs and passive investing, it has become easier, cheaper and more popular than ever for retail investors to track.

There are many ways to invest in the S&P 500 index:

1. The S&P500 futures market (whether the E-mini or the full-size contract)
2. Options and other derivatives markets related to the S&P 500
3. Exchange traded products, such as the SPDR S&P 500 ETF (SPY)
4. Investment funds

All those markets have different types of investors, with various degrees of sophistication, risk tolerance, goals and their interactions, alongside changes in the constituent shares of the index, determine the value of the index.

Those investors also leave a trail beyond price, as **those various markets publish information about the activity of their participants and their positioning, which potentially could be used to gauge the future direction and returns of the index.**

1.1 Aim of the project

We would like to investigate whether it is possible to predict the returns of the S&P 500 index based on

1. Previous prices / returns - as traditionally used by technical analysis
2. Non-price information that could hint to the trading activity of influential market participants, namely:
    - Put and call option volume
    - The number of different types of large investors and their positioning in the futures market (long or short)
    - The number of long and short future contracts that investors have

As the non-price data reveals information about investor sentiment and positioning, we might be able to use that information to predict future price movements, **which would be of tremendous use to investors, whether small or very large, to design trading strategies that deliver performance beyond buy and hold**.

We will examine a number of algorithms alongside the above variable to compare their predictive powers for weekly S&P 500 returns using the above data versus very simple models based on average or previous historical returns.

2 The datasets

We will be looking at the following datasets, which are all freely available on Quandl (www.quandl.com):

- The price and trading volume of the SPDR SPY security, a highly liquid ETF which closely tracks the returns of the main index
  - The dependent variable will be the return on closing price adjusted for dividends and splits, as this is the price investors will pay attention to
  - Volume reflects the volume of shares traded at a certain point in time
- The futures only commitment of traders (hereafter "COT"") report for the S&P 500 index, with the following key variables:
  - The number of S&P500 futures contracts, split by long and short positions held by
    - large speculators, often assumed to be "smart money"
    - commercial operators, which include brokers and investment banks
    - other participants
- The put and call ratio for equity indices from the CBOE, which represents the volume of put and call options traded in the US on various equity indices, which is likely to be a good proxy for S&P500 options volume

The idea behind the COT and put-call volume data is to understand how various market participants are positioned at a point in time, and to see whether this has predictive power with respect to changes in the index price.

We have chosen to look at the SPY ETF rather than the futures contract as this would be the most approachable way for retail traders with small accounts to trade the index, albeit with no or limited leverage. However, we would not expect the analysis to differ significantly if using the index itself.

2.1 Note about adjustments to datasets to ensure accurate analysis

All the data loading, analysis, charts and reporting was performed in R using freely available sources and software. Whilst all the raw data was easily available, a number of key steps were performed to clean the data and ensure the analysis performed would reflect real world conditions:

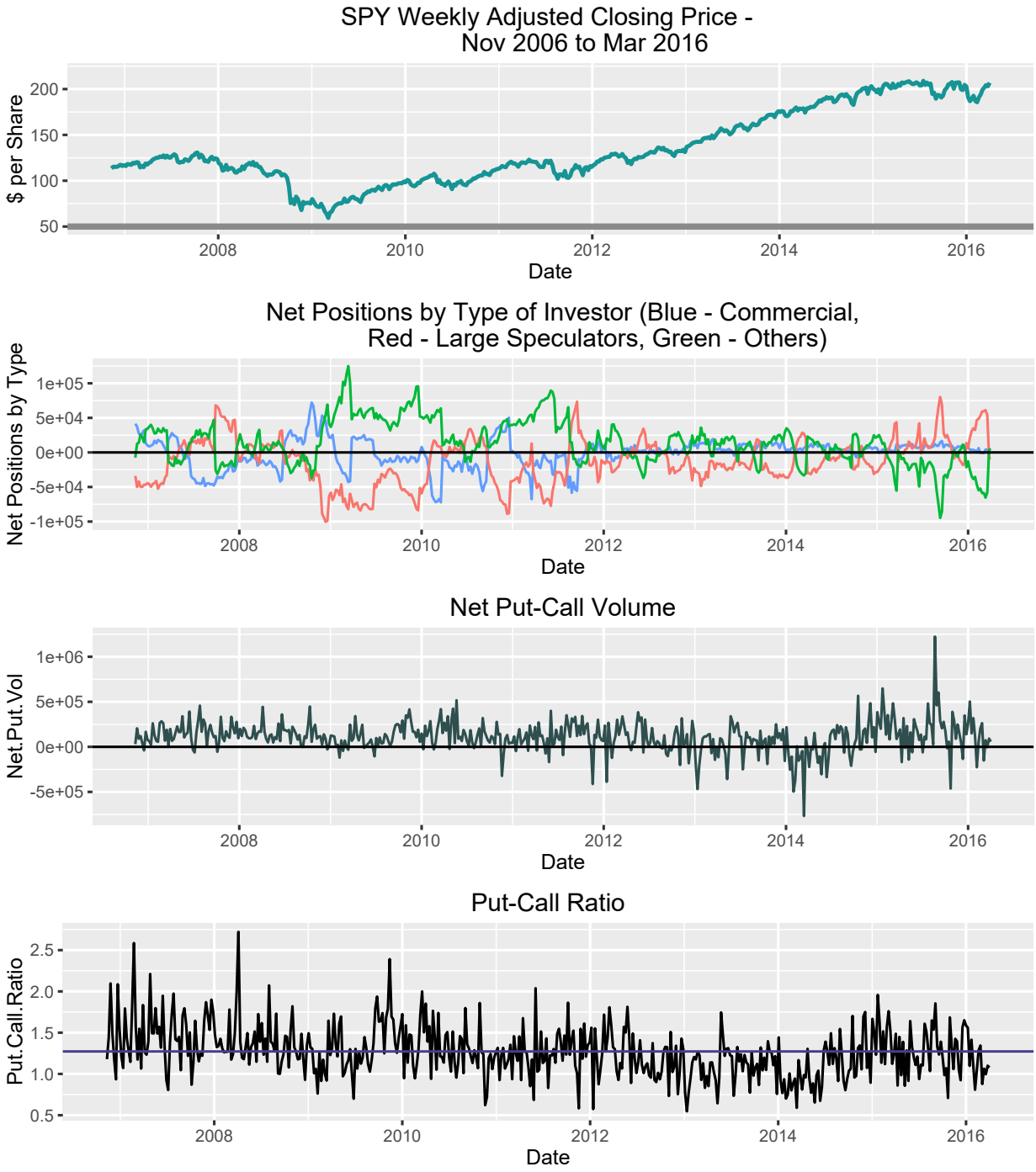
1. **Data frequency harmonisation** : Whilst the price/volume and put-call data were available on a daily frequency (for weekdays, i.e. when the market is open), the COT data is only released weekly on Fridays with respect to investors' positions on Tuesday of the relevant week. As a result, all the datasets were transformed to weekly time series
2. **Adjustments for holidays** : The end of each week for the weekly data represents the last day of that week for which the market was open. i.e. if the market was closed on a Friday, the last data shown for that week relates to a Thursday. Likewise, on holidays, the COT data related to the nearest available day before or after market close (sometimes Mondays, other times Wednesdays). The datasets were adjusted to ensure that no observations were unnecessarily removed due to holidays or mismatched week ends.
3. **Adjustments for look-ahead bias** : To prevent perfect foresight, variables were suitably lagged to ensure that price forecasts were performed only on the basis of information available at the time of the forecast (i.e. only using information from previous observations)

4. **Adjustments to independent variables**: The raw independent variables may have some predictive value, but it may be that returns are responsive to the change in those variables or to various ratios. Variables were transformed by looking at differences and ratios (e.g. looking at both the put/call ratio, and the difference between the volume of call options and put options) which could be more useful than the raw variables.

All the code is available in a separate R file.

3 Exploring relationships between the variables

Let us plot the datasets to see whether we can observe any relationship between the variables:



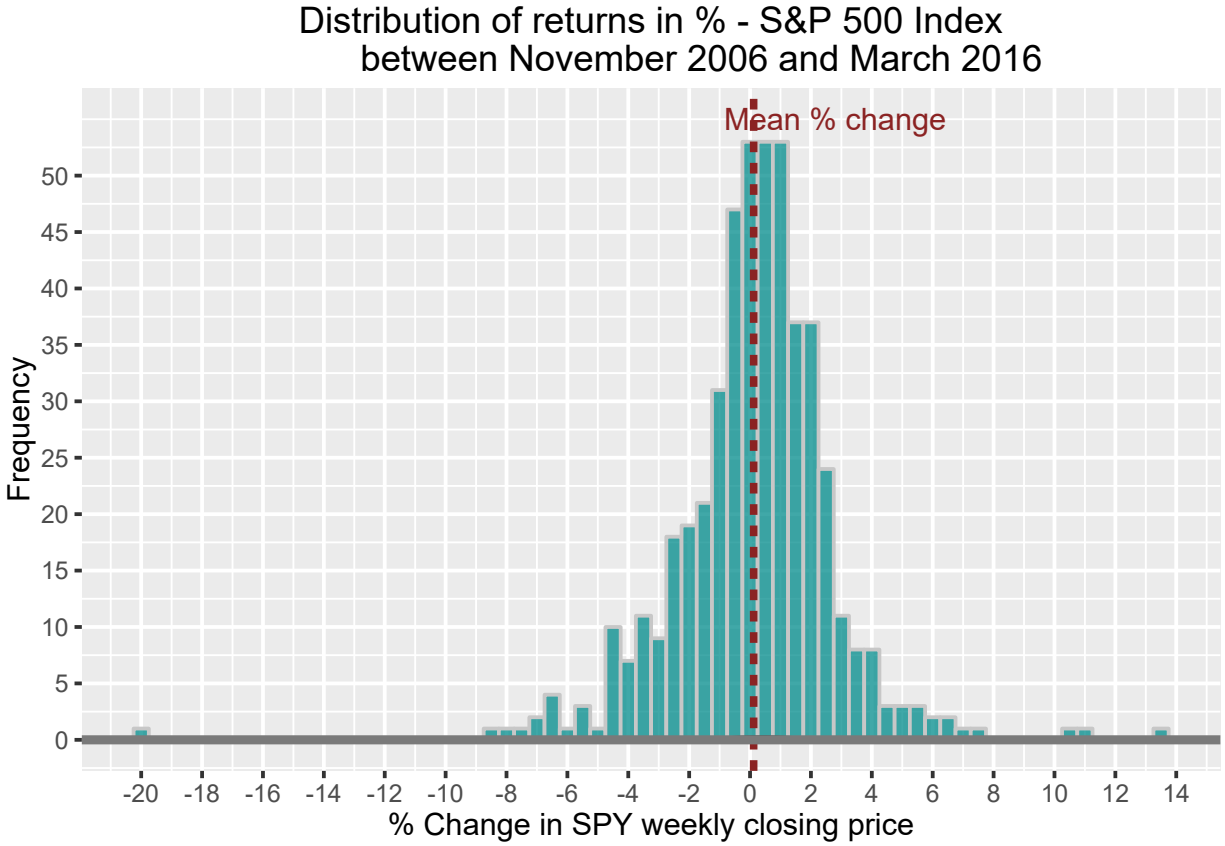
There are a number of interesting observations we can make from the closing price chart:

- The 2008 crisis and subsequent recovery is evident, alongside a long-term uptrend since 2009
- The price series is clearly non-stationary, with mean closing prices changing over time

In order to prevent issues arising from working with non-stationary series (such as but not limited to spurious

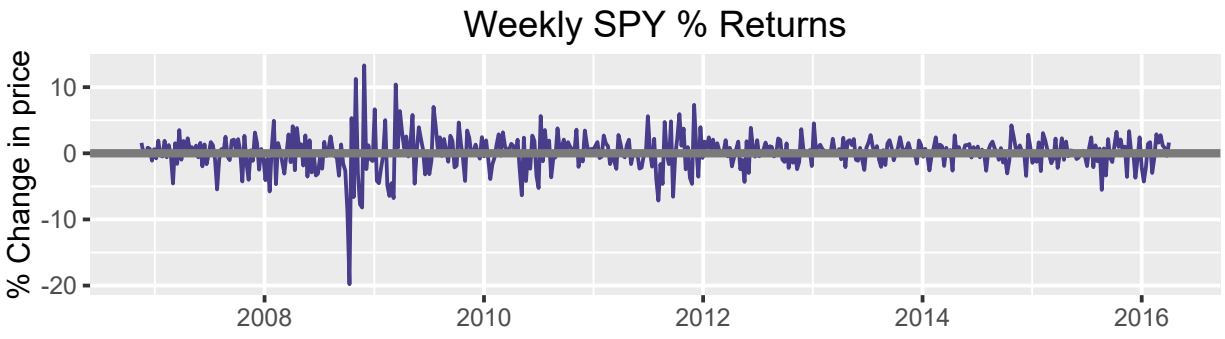
regressions), **going forward we will look at predicting the rate of change (or percentage change in prices)**, from which we can interpolate predicted prices[1].

Let us now turn our attention to the distribution of returns.



We can see that the distribution of returns has fat tails, with a few extreme values (one weekly drop of c.20%, and a few occurences with weekly increases in price of over 10%).

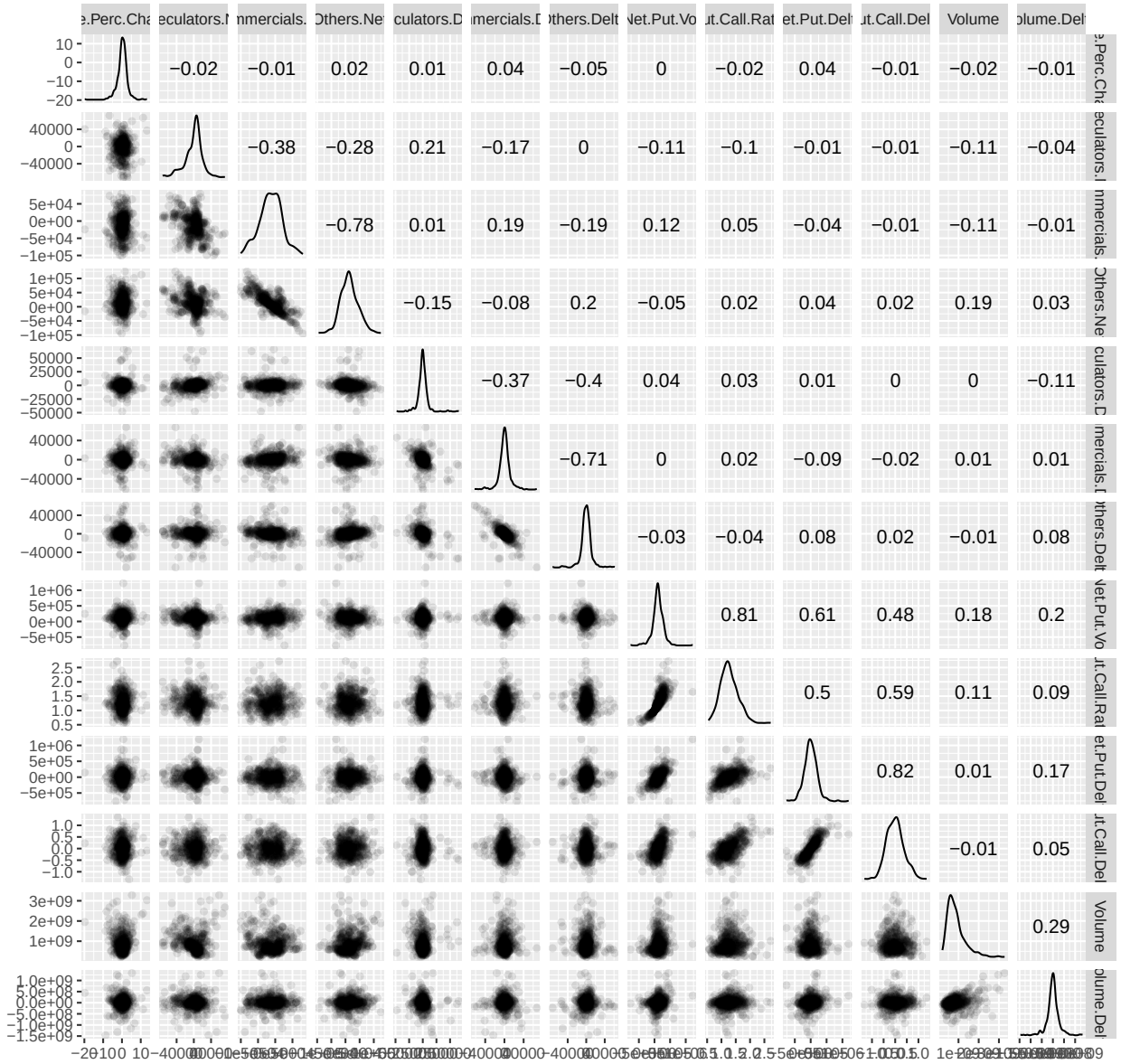
The mean and median percentage change in price are close to zero, but slightly positive.



Plotting returns over time shows that there is significant and differing variance in returns in different time periods.

[1] All price predictions in the paper are calculated from predicted returns.

3.1 Plotting cross-correlations between price and non-price variables



At first sight of the cross-correlation matrix plot, there is no obvious linear correlation between % change in price and non-price variables.

4 Applying various forecasting models

We will look at a number of algorithms to compare them to very simple forecasting algorithms to test their usefulness and accuracy.

The models will be compared as follows:

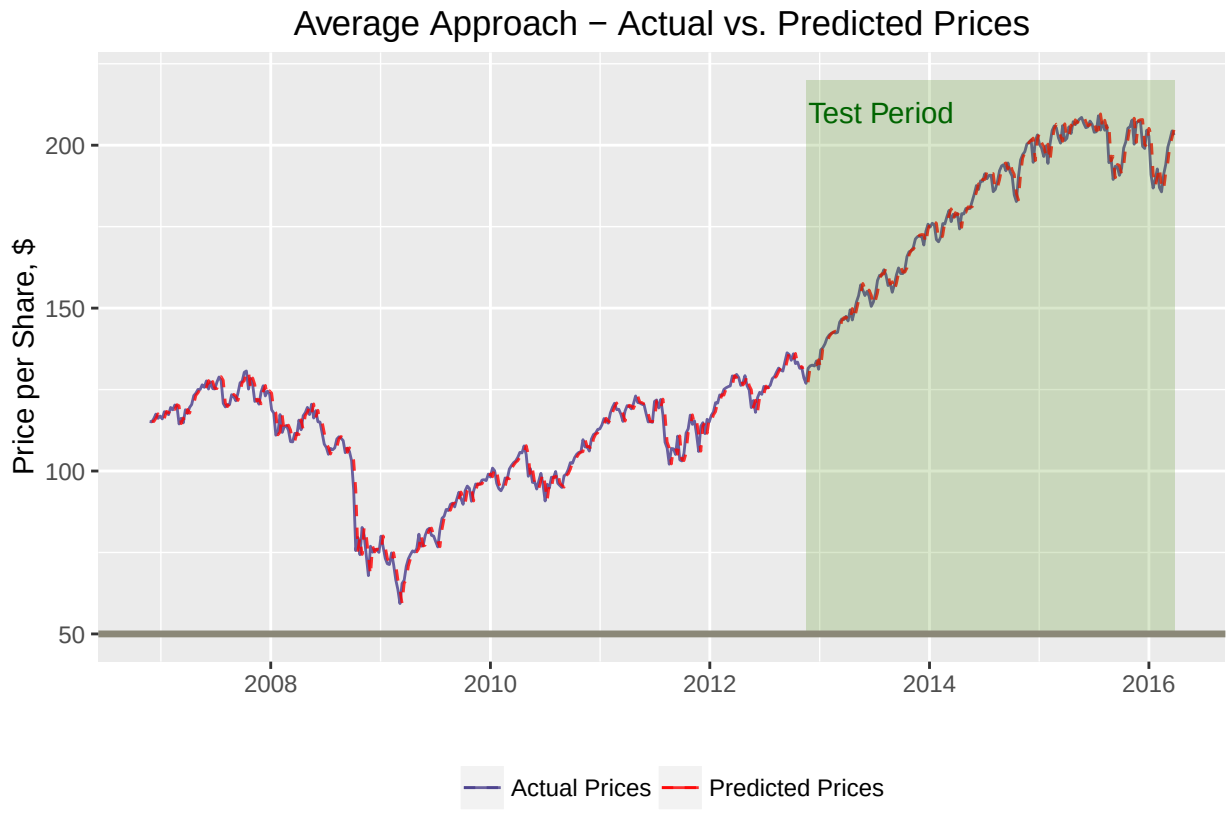
1. A training period (between 9th November 2006 and 9th November 2012) during which the model is fitted
2. An out-of-sample period (from 16 November 2012 to 25th March 2016), where the accuracy of the fitted model is tested

We will look at the square root of the mean squared errors ("RMSE") to compare the models' accuracy.

4.1 Average Approach

Given the distribution of weekly returns, with many observations centered around mean returns, a simple approach would be to **assume returns in the test set are equal to the mean returns of the training set**.

The following chart shows the predicted price (derived from predicted weekly returns) for the average approach:

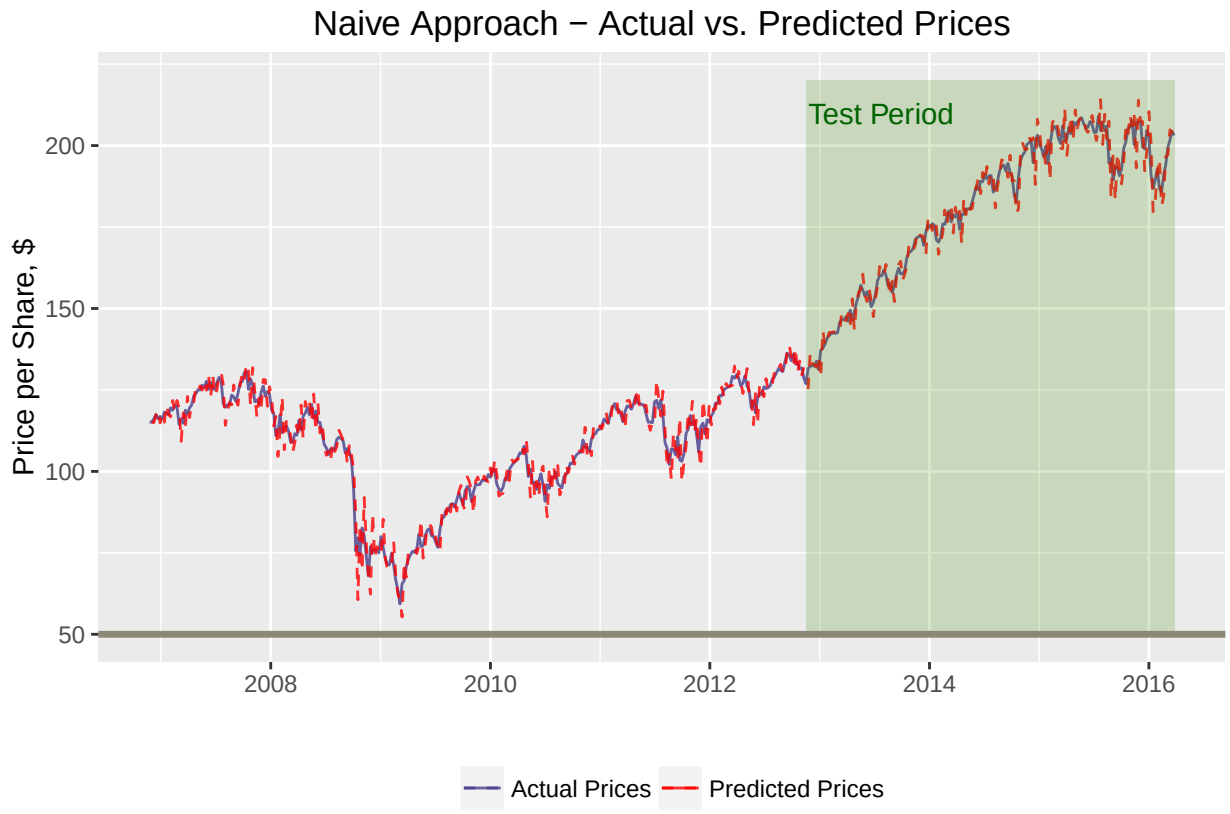


The RMSE for the average approach is `3.0944138` in the training period, and `1.6917216` in the test period.

4.2 Naive Approach

Another simple forecasting model would be to use a naive approach, where **the following week's returns are expected to equal this week's returns**.

The following is a plot of actual versus predicted values for the naive approach:



It seems from the chart that the naive approach is not particularly appropriate for the SPY, and we would expect the RMSE to be greater than for the average approach.

Indeed, in the training period and test period, the RMSE for the naive approach are `4.5593547` and `2.5011942` respectively.

We will now examine whether other algorithms can improve on those metrics.

4.3 Multiple Linear Regression

The first model we will look at is multiple linear regression, where we attempt to forecast wekly returns using a number of variables, including the previous change in weekly prices, the net positions of various types of traders, put-call ratio and volume data.

The summary of the fitted model on the training period is as follows:

```
##
## Call:
## lm(formula = Price.Perc.Change ~ Previous.Price.Perc.Delta +
##     Speculators.Net + Commercials.Net + Others.Net + Speculators.Delta +
##     Commercials.Delta + Others.Delta + Net.Put.Vol + Put.Call.Ratio +
```

```
##     Net.Put.Delta + Put.Call.Delta + Volume + Volume.Delta, data = training)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -20.2964  -1.3990   0.0517   1.6290  13.3218
##
## Coefficients: (2 not defined because of singularities)
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                -6.269e-01  1.733e+00  -0.362   0.7178
## Previous.Price.Perc.Delta -9.532e-02  6.462e-02  -1.475   0.1412
## Speculators.Net           -9.254e-06  8.682e-06  -1.066   0.2874
## Commercials.Net           -5.537e-06  6.077e-06  -0.911   0.3630
## Others.Net                        NA         NA      NA       NA
## Speculators.Delta          1.535e-05  1.863e-05   0.824   0.4107
## Commercials.Delta          2.191e-05  1.742e-05   1.258   0.2094
## Others.Delta                      NA         NA      NA       NA
## Net.Put.Vol               -1.466e-06  3.829e-06  -0.383   0.7020
## Put.Call.Ratio             6.616e-01  1.424e+00   0.465   0.6425
## Net.Put.Delta              5.649e-06  2.822e-06   2.002   0.0462 *
## Put.Call.Delta            -2.122e+00  1.118e+00  -1.898   0.0586 .
## Volume                    -1.212e-10  4.152e-10  -0.292   0.7706
## Volume.Delta              -7.433e-10  6.964e-10  -1.067   0.2867
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.1 on 300 degrees of freedom
## Multiple R-squared:  0.03478,    Adjusted R-squared:  -0.0006102
## F-statistic: 0.9828 on 11 and 300 DF,  p-value: 0.462
```

According to the regression, there are no really significant explanatory variables (with the change in the difference between put and call volume and the change in the put-call ratio being the most important variables, following by previous weekly return).

It is interesting to note that the coefficient for previous weekly return is negative, which would support that returns are not persistent on a weekly basis.

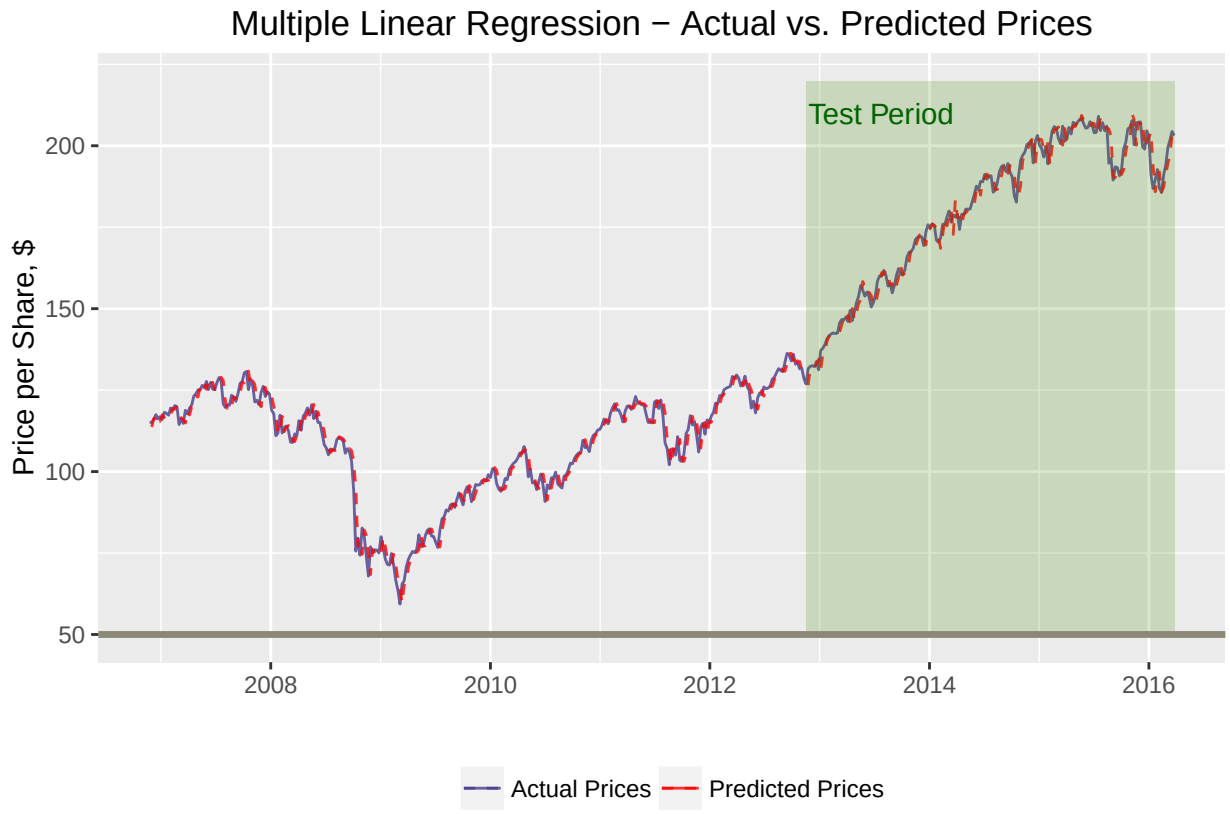
Given the lack of really significant variables and to prevent overfitting, let us simplify the linear regression to only take into account the previous week's returns, change in net put volume and the change in the put call ratio.

The summary of that regression model is as follows.

```
##
## Call:
## lm(formula = Price.Perc.Change ~ Previous.Price.Perc.Delta +
##     Net.Put.Delta + Put.Call.Delta, data = training)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -20.6214  -1.3061  -0.0169   1.6672  12.5900
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                9.002e-02  1.747e-01   0.515   0.6067
## Previous.Price.Perc.Delta -5.618e-02  5.809e-02  -0.967   0.3342
```

```
## Net.Put.Delta              3.940e-06  1.966e-06   2.004   0.0459 *
## Put.Call.Delta            -1.427e+00  8.222e-01  -1.736   0.0836 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.084 on 308 degrees of freedom
## Multiple R-squared:  0.01973,    Adjusted R-squared:  0.01018
## F-statistic: 2.066 on 3 and 308 DF,  p-value: 0.1046
```

Let us see how the model does out of sample

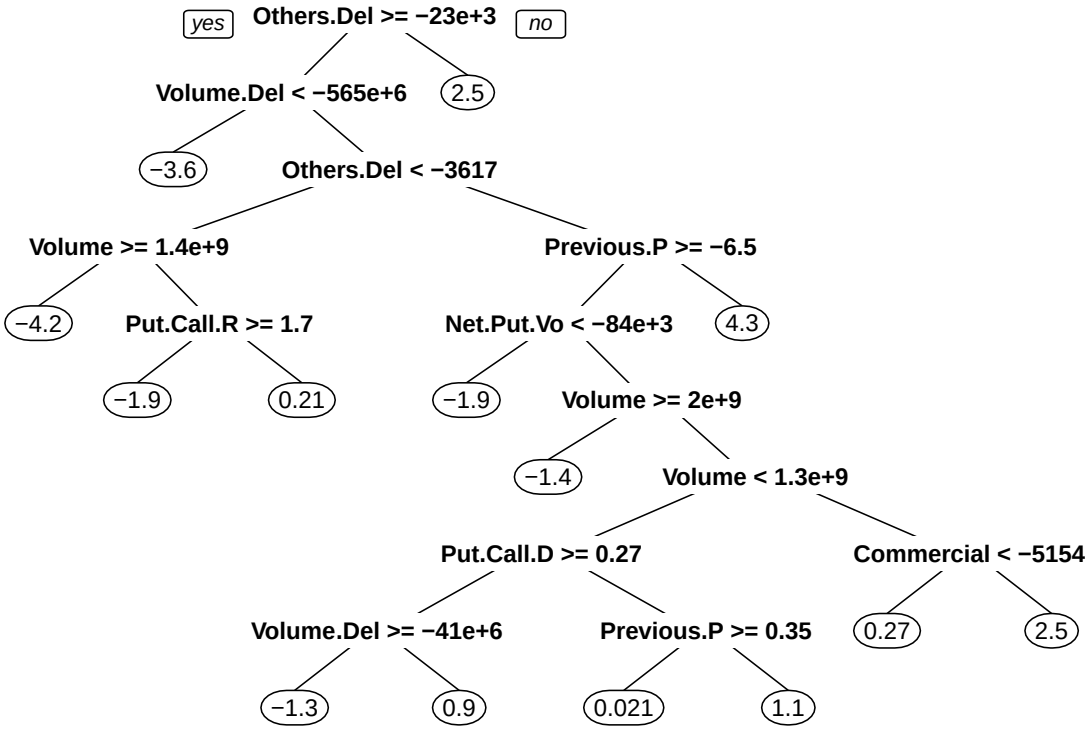


The RMSE in and out-of-sample are `3.0637344` and `1.826383` respectively.

4.4 Regression Tree model

We will look at a decision tree model to see whether it is possible to find non-linear relationships between the dependent and independent variables. We will use an algorithm where each tree leaf needs to have at least 30 observations.

The following shows the decision tree as fitted to the training set using default values:



The tree is complex and likely to be overfit. The summary of errors at different levels of complexity is shown below:

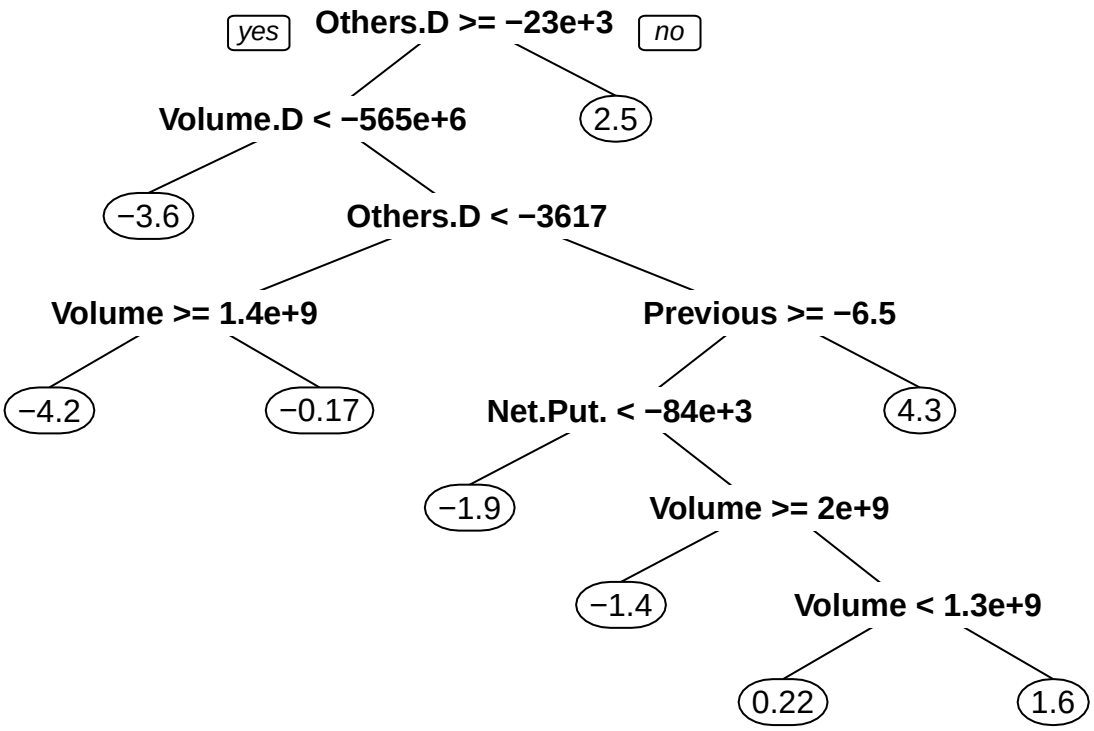
```
##
## Regression tree:
## rpart(formula = Price.Perc.Change ~ Speculators.Net + Commercials.Net +
##     Others.Net + Speculators.Delta + Commercials.Delta + Others.Delta +
##     Net.Put.Vol + Put.Call.Ratio + Net.Put.Delta + Put.Call.Delta +
##     Previous.Price.Perc.Delta + Volume + Volume.Delta, data = training,
##     method = "anova")
##
## Variables actually used in tree construction:
## [1] Commercials.Delta         Net.Put.Vol
## [3] Others.Delta              Previous.Price.Perc.Delta
## [5] Put.Call.Delta            Put.Call.Ratio
## [7] Volume                    Volume.Delta
##
## Root node error: 2987.5/312 = 9.5754
##
## n= 312
##
##         CP nsplit rel error xerror    xstd
## 1 0.034905      0   1.00000 1.0054 0.16592
## 2 0.013639      5   0.80529 1.3297 0.21793
## 3 0.013445      8   0.76438 1.4155 0.21985
## 4 0.013190     10   0.73749 1.4296 0.21987
```

```
## 5 0.011885     11   0.72430 1.4367 0.21958
## 6 0.011341     12   0.71241 1.4391 0.21943
## 7 0.010000     13   0.70107 1.4440 0.21941
```

While looking at the table above, minimising the cross-validated error would suggest picking no node and going with the mean weekly return, **essentially the same as the average approach**.

However, to see whether increasing the complexity of the tree with 5 splits versus no split actually increases the prediction power versus the average approach, let us test such a tree on the test period.

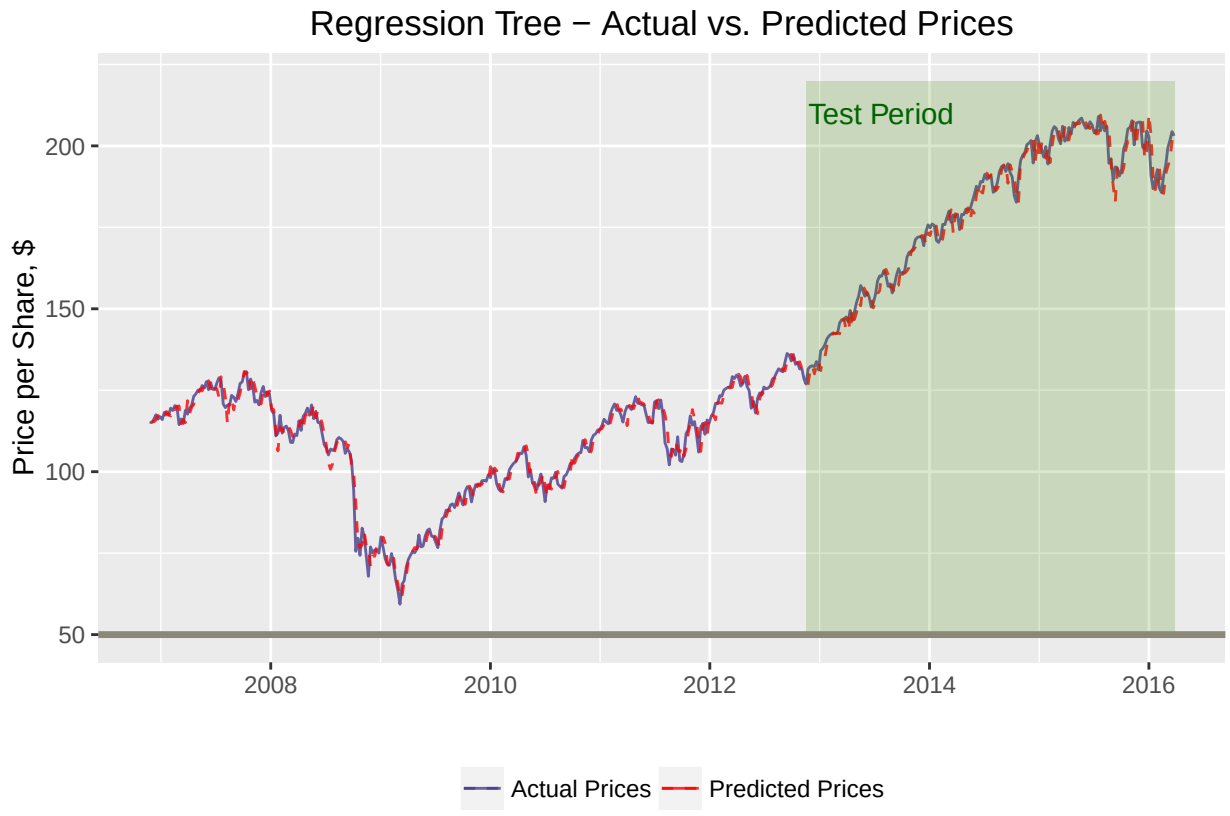
The following shows the result pruned tree:



The pruned tree yields interesting results, making use of the following variables:

- Change in other speculators' net positions
- Volume
- Change in volume
- Change in previous price
- Change in net put volume

Although the number of observations and different types of market conditions and regimes may potentially be low for the algorithm, let us see whether running the fitted model on the test set improves upon our benchmark models.

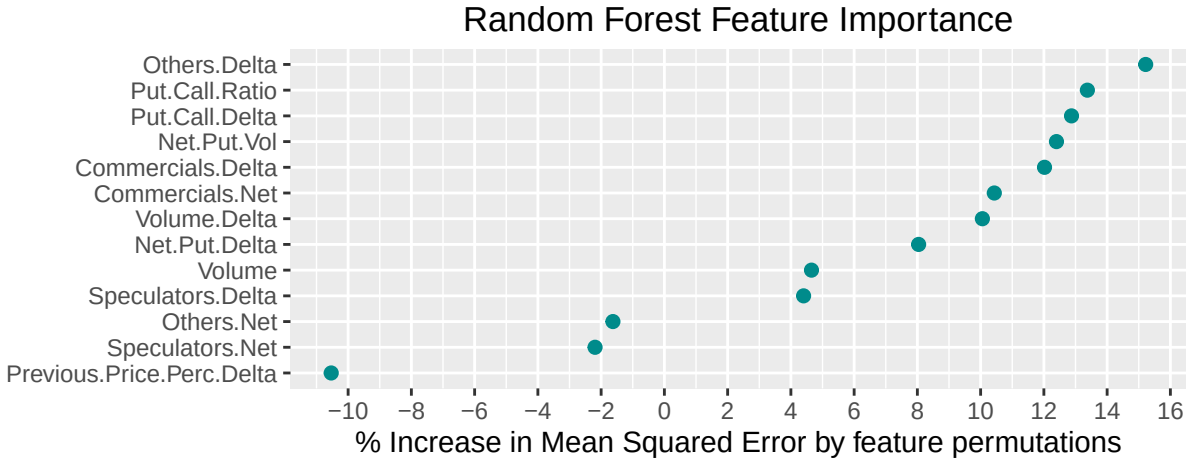


The RMSE in the training and test period are `2.7054022` and `1.950928` respectively.

4.5 Random Forest Algorithm

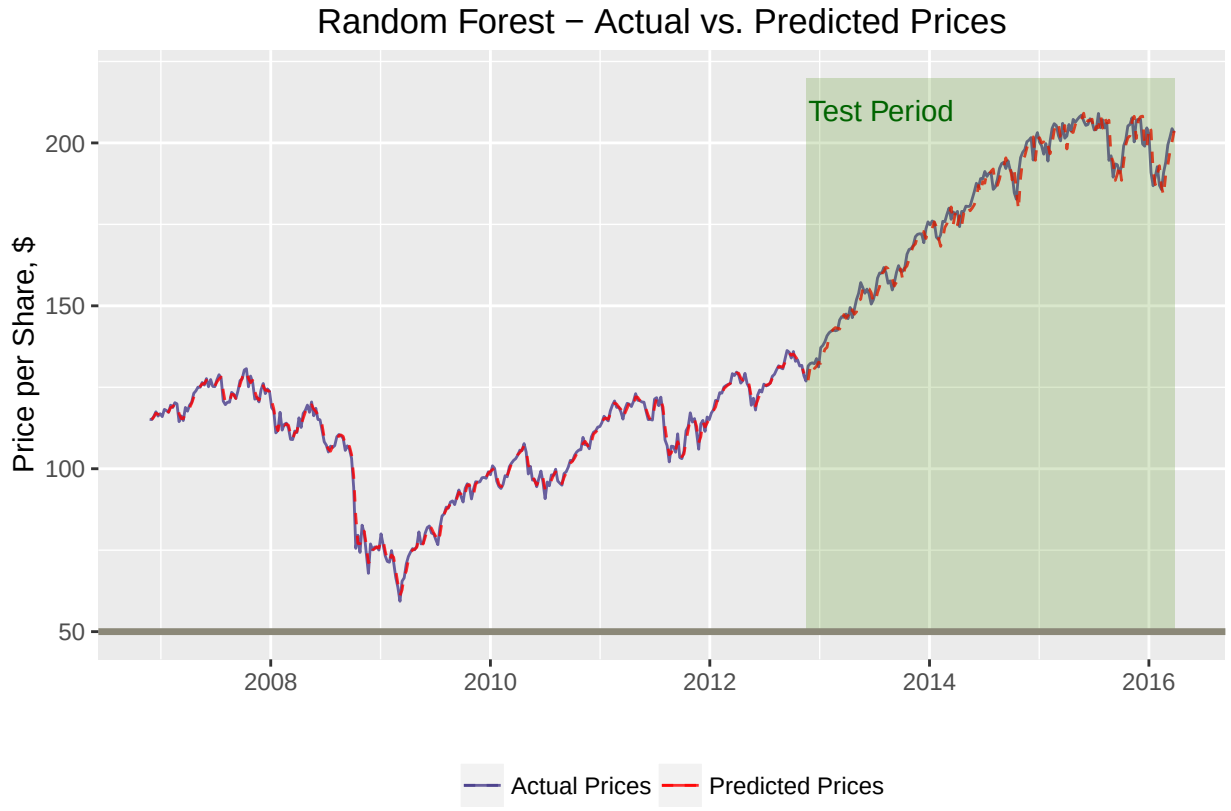
We will examine whether a random forest algorithm can improve upon the results of the regression tree. The random forest algorithm will construct a multitude of decision trees and will choose the mean predicted return of those individual trees. Using this method may correct the previous algorithm's tendency towards overfitting.

We will run the random forest algorithm with 5,000 trees. The following chart shows the importance of chosen variables in the trees:



The above chart shows the % increase in mean squared error as a variable is permuted using random values (within the dataset), and thus is an indicator of the importance of that variable in the algorithm.

Let us look at the predictive power of the random forest algorithm:



The RMSE for the random forest algorithm was `1.4415365` and `1.8800516` in the training and test periods.

5 Model Comparison

5.1 Root Mean Squared Error

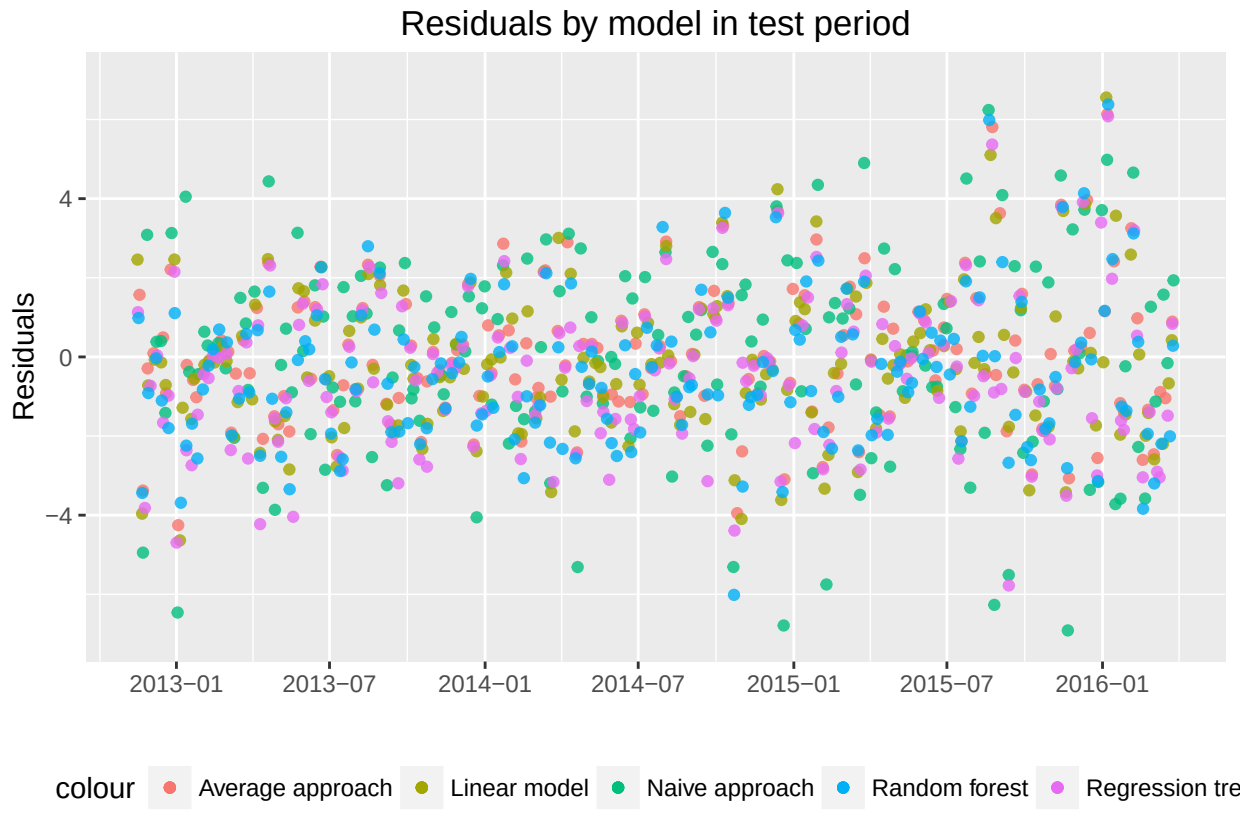
The scale of the price chart, the resolution used and the relatively small deviations between actual and predicted values mean that looking at a comparison of RMSE will be the best way of gauging the accuracy of various models:

Model	RMSE Training Set	RMSE Test Set
Average	3.0944138	1.6917216
Naive	4.5593547	2.5011942
Multiple Linear	3.0637344	1.826383
Regression Tree	2.7054022	1.950928
Random Forest	1.4415365	1.8800516

As may be expected, in the training period, model complexity improves accuracy in the training set as the multiple linear model, regression tree and random forest algorithms fit the training data better than our benchmark models.

However, as we apply the fitted models to the test period, the predictive power of the complex algorithms as measured by RMSE is worse than a simple average approach.

For graphical representation, the following chart shows the residuals by model during the test period



From the table and the chart, we can see that the worst model is the naive approach by far.

The multiple linear model, which has few variables used compared to the regression tree and random forest models, performs the best in the out-of-sample period out of the test algorithms, despite not having the best fit in the training set.

This would suggest potential overfitting for the regression tree and random forest algorithms, or a lack of observations or truly meaningful features to allow those algorithms to truly shine.

While bearing this in mind, there are quite a number of points we can draw from the overall analysis that could be useful for developing this project further.

5.2 Accuracy in prediction of correct direction of price move

Another analysis was performed to look at the percentage of times the predictions in the model were directionally correct (i.e. predicting positive or negative returns correctly, while ignoring magnitude). This could be very useful to develop directional trading model if accuracy is significantly better than random:

The following table shows the ratio of predictions that were correct in the training and test sets. Of course, the test set is what we are interested in:

Model	Correct Direction Training Set	Correct Direction Test
Average	0.525641	0.6136364
Naive	0.4615385	0.4772727
Multiple Linear	0.5480769	0.5340909
Regression Tree	0.5865385	0.4488636
Random Forest	0.9358974	0.5056818

The results at first sight, excluding for the average model, suggest that the models are not significantly different from random in their ability to guess the correct direction of a price move.

Nevertheless, it is important to bear two points in mind:

1. The models developed were all based on regression rather than classification analysis. It is possible that training models to classify the direction of a move may yield different and potentially better results
2. The direction of a price move is important, but so is the magnitude when looking at trading models, so a trading strategy could still make money by being accurate on price direction less than 50% of the time (before trading costs) as long as appropriate risk management is followed.

6 Thoughts and Conclusion

Fitting the models was an interesting exercise, but the results should be tempered by the following points:

1. Data is limited by the availability of COT information in the format required, and further observations may be able to improve the accuracy of the models and their robustness (although we should remember that markets do change constantly, so going back too far where dynamics are not representative of current or future market conditions would not help much)
2. The training and test periods are very particular, containing a verious significant crash and a strong recovery since propped by quantitative easing. Those conditions are not likely to stay constant, and models trained on this particular period may be inadequate as market and macroeconomic conditions change (e.g. rising interest rates, increase in the use of quantitative strategies at higher timegrames...). As a result, risk management and checking the validity of models is paramount to ensure we can continue to adapt over time.

Nevertheless, analysis shows interesting conclusions based on the sample size available, which could be used to investigate trading models that could generate profits. From looking at the distribution of weekly returns, which are close to zero but slightly positive, we can draw two interesting observations for further research for trading research:

1. Weekly returns are on average (mean and median) very slightly positive but close to zero. This means that a mean-reverting strategy that would fade intra-week price moves may statistically be able to be profitable over many trades given the distribution of returns (assuming the deviations are sufficient to offset trading costs)
2. Weekly returns are one thing - the charts show that there are longer-term trends. If looking at longer timeframes, and assuming an investor could tolerate intra-week or intra-month drawdowns, following the dominant long-term trend with appropriate risk management would be a viable strategy
3. The fact that the average approach works well, and correctly guesses price direction 60% of the time shows that there the distribution of returns is biased towards positive returns (as is evident in the long-term uptrend since 2009)

There are a number of additional analyses and projects that could be performed from this work that would be highly relevant and practical for any investor:

1. The aforementioned analysis of intra-week movements would be a worthwhile project, alongside the development of short-term mean reversion models that could take advantage of the distribution of weekly returns
2. On the other hand, a look at even higher timeframes (monthly or quarterly) could be undertaken to see whether variables have longer-term prediction power, and whether longer-term returns would favor other trading models (such as trend following)
3. The development of a complete trading backtesting system that could integrate input from the models we looked at, by taking positions when predicted price movements are sufficient to offest trading costs, to see whether those models can do beat other trading strategies
4. Looking at more sophisticated algorithms, such as neural networks.

We may look at the above analysis in another project, but to conclude, one maxim comes to mind with respect to this project **"Often simplicity trumps unnecessary complexity"**, as simple models which exploit market patterns that can easily be understood and are resilient to market changes will stand the test of time.