Random Forest

Jesse Keränen

12/8/2021

Prologue

One thing every investor would like to know is whether price of the given asset goes up or down next day. If investor would also know the magnitude of the change trading would become pretty easy. One attempt to solve this problem has been to use machine learning algorithms and volatility factors to predict the sign of the change of asset. Some researches go so far that they even try to estimate how much the asset is going to change in given time period. This makes them possible to use in portfolio optimization. In this file I am only going to try to estimate if price of one asset is going go up or down using random forest algorithm.

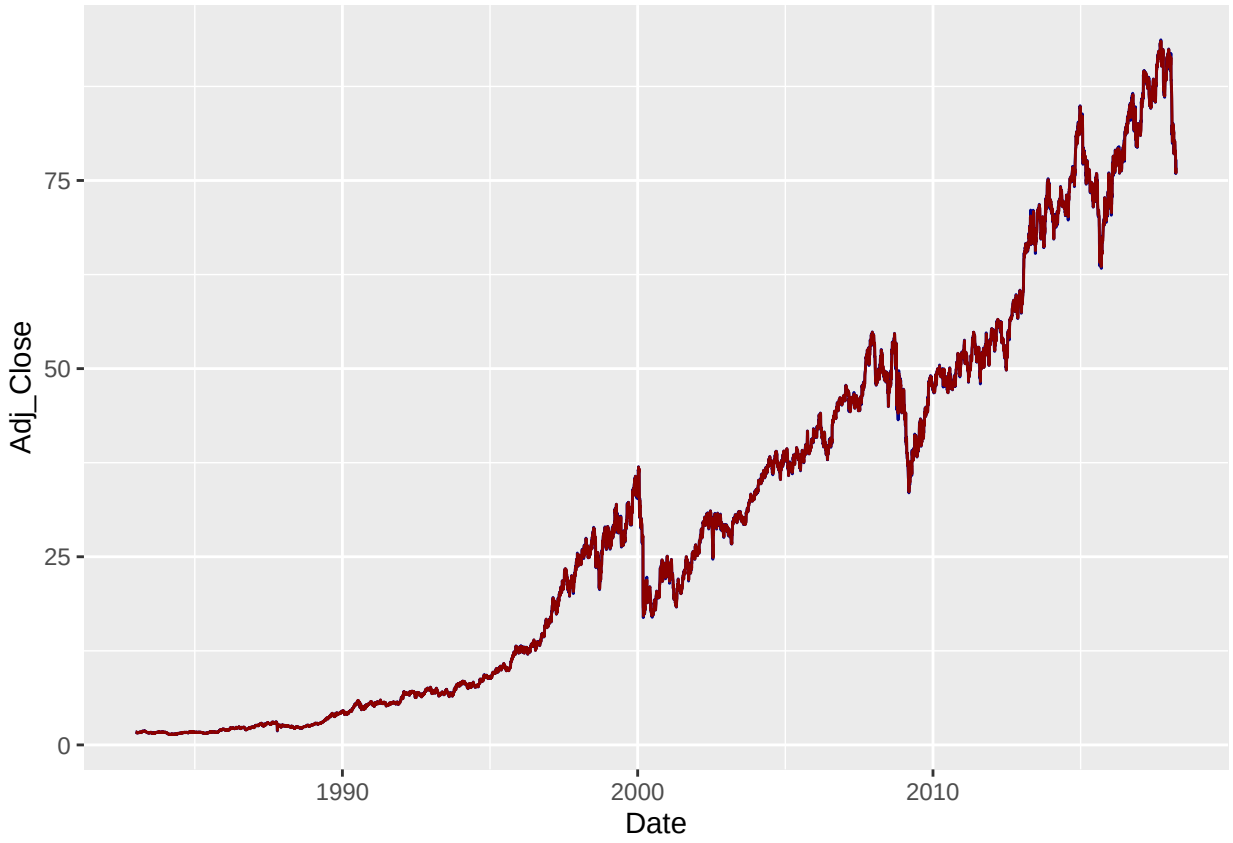
```
library(ggplot2)
library(Quandl)
library(tidyverse)
library(data.table)
library(zoo)
library(pracma)

Quandl.api_key("bx1qdehfWXg6SNKnicQC")

# For monthly data use collapse = "monthly"
price <- as.data.table(Quandl(c("WIKI/PG"), start_date
        = "1983-01-01", end_date = "2021-12-31", collapse = "daily"))

price <- price[, .(Date,`Adj. Open`, `Adj. High`, `Adj. Low`, `Adj. Close`,
                   `Adj. Volume`)]

price <- na.omit(price)
colnames(price) <- c("Date", "Adj_Open", "Adj_High", "Adj_Low", "Adj_Close", "Adj_Volume")

price <- price[order(Date)]

price[, "Adj_Return" := Adj_Close/shift(Adj_Close) -1]
price <- na.omit(price)
```

Data smoothing

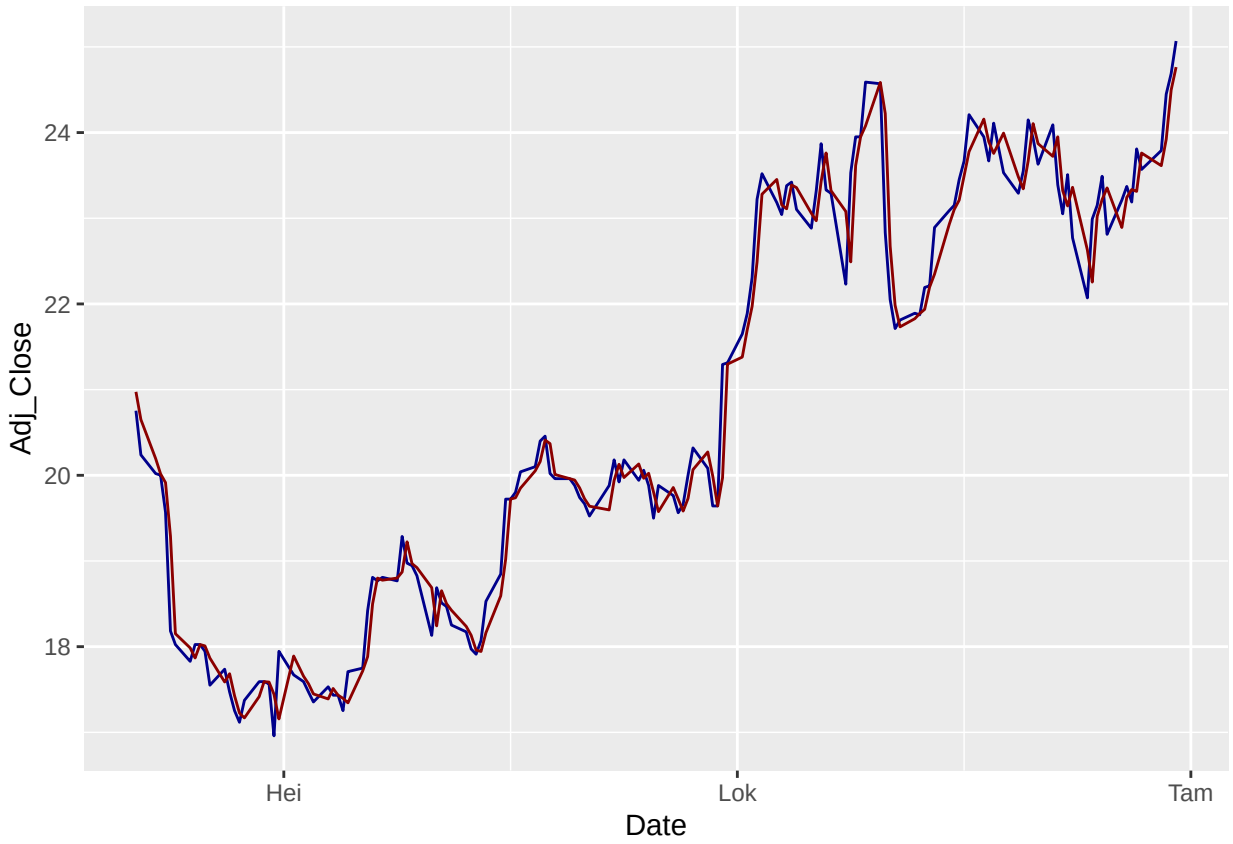
Financial data normally consists lots of noise. Impact of this noise can be reduced by smoothing the data. Prices are smoothed using exponential smoothing which means that sets the price of time t to be smoothed average of observed value at time t and smoothed value at time $t-1$. Smoothing factor α can be set. I have read other studies using α value of 0.2 so I will use the same.

$$\hat{P}_0 = P_0,$$
$$\hat{P}_{t+1} = \alpha P_{t+1} + (1-\alpha)\hat{P}_{t+1}$$

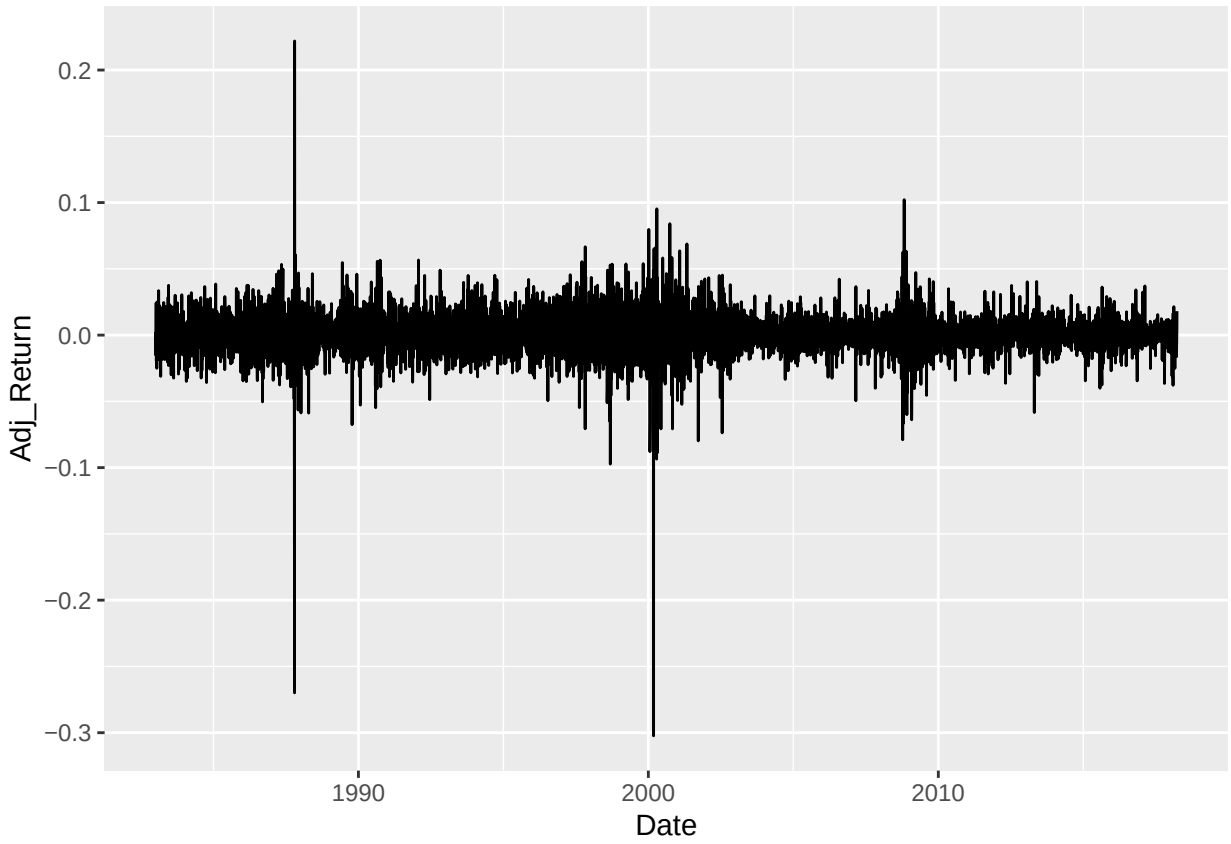
```
alpha <- 0.2
price[, Adj_Close_Smooth := alpha * Adj_Close + (1-alpha)*shift(Adj_Close)]
price[1, Adj_Close_Smooth := Adj_Close]

# In the big picture it is hard to see any difference between smoothed and not
# smoothed prices
ggplot(price, aes(Date, Adj_Close)) + geom_line(color = "darkblue") + geom_line(aes(Date, Adj_Close_Smo
```




```
# When you look shorter period, effect of smoothing can be seen
ggplot(price[year(Date) == 2000 & month(Date) >= 1 & month(Date) >= 6], aes(Date, Adj_Close)) + geom_li
```




```
ggplot(price, aes(Date, Adj_Return)) + geom_line()
```



Explanatory variables

So that our algorithm can make reasonable estimates, we need to give it relevant information on which it can base it's predictions. We are going to use set of trading indicators. I am not too familiar with quantitative trading so I am going to use same indicators I have seen in other studies. Using these indicators is convenient for us since they base largely on trading volume of asset. We can easily download trading volume from same data base as price data. Some accounting variables like book value we can't get from Quandl data base.

On Balance Volume

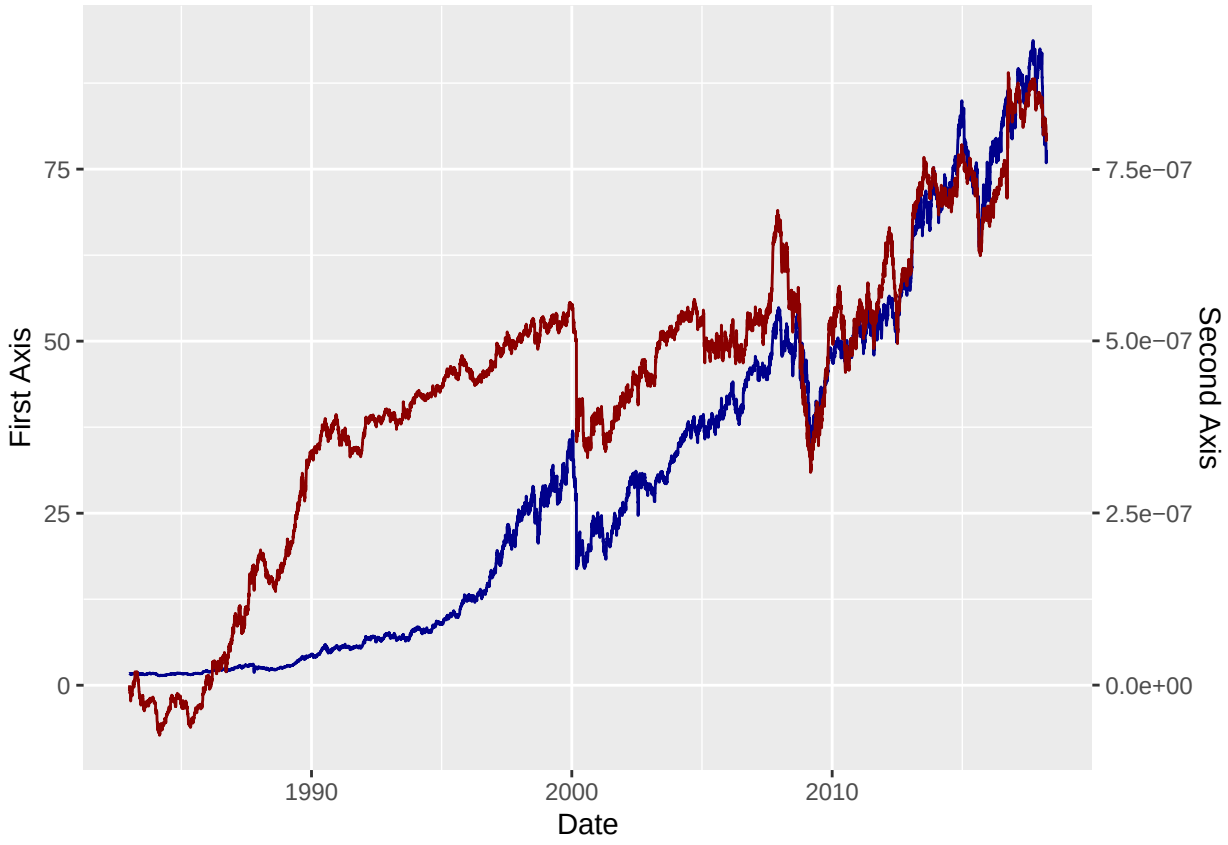
On Balance Volume is a cumulative trading pressure measure. OBV can remain same or it can be calculated by adding or reducing trading volume from last OBV. Different scenarios are show below.

$$OBV_t = OBV_{t-1} + \begin{cases} volume, & if\ Price_t > Price_{t-1} \\ 0, & if\ Price_t = Price_{t-1} \\ -volume, & if\ Price_t < Price_{t-1} \end{cases}$$

Idea of the On Balance Volume is that volume proceeds price. I think OBV as pressure for stock price. When volumes are high and prices increase there is still pressure for assets price to increase. Again when volumes and prices are decreasing prices are more likely to still decrease.

```
price[, OBV := 0]
for(i in 2:price[, .N]){
  price$OBV[i] = price$OBV[i-1] + sign(price$Adj_Return[i]) *
```

```
    price$Adj_Volume[i]
}

ggplot(price, aes(x = Date)) + geom_line(aes(y = Adj_Close), color = "darkblue") + geom_line(aes(y =
  scale_y_continuous(
    # Features of the first axis
    name = "First Axis",
    # Add a second axis and specify its features
    sec.axis = sec_axis( trans=~.*10^(-8), name="Second Axis"))
```



Stochastic Oscillator %K

Stochastic Oscillator is a measure indicating overbought and oversold situations. It will vary between 0 and 100. It will tell you how high is the current price of the asset compared to it's high and low values within K last days. I will use 14 last day, which again seems to be quite usual in literature. If Stochastic Oscillator gets values close to 100 it means that current price is close highest high value within last two weeks. When value is close to 0 it means that current value of the asset is close to its minimum low value.

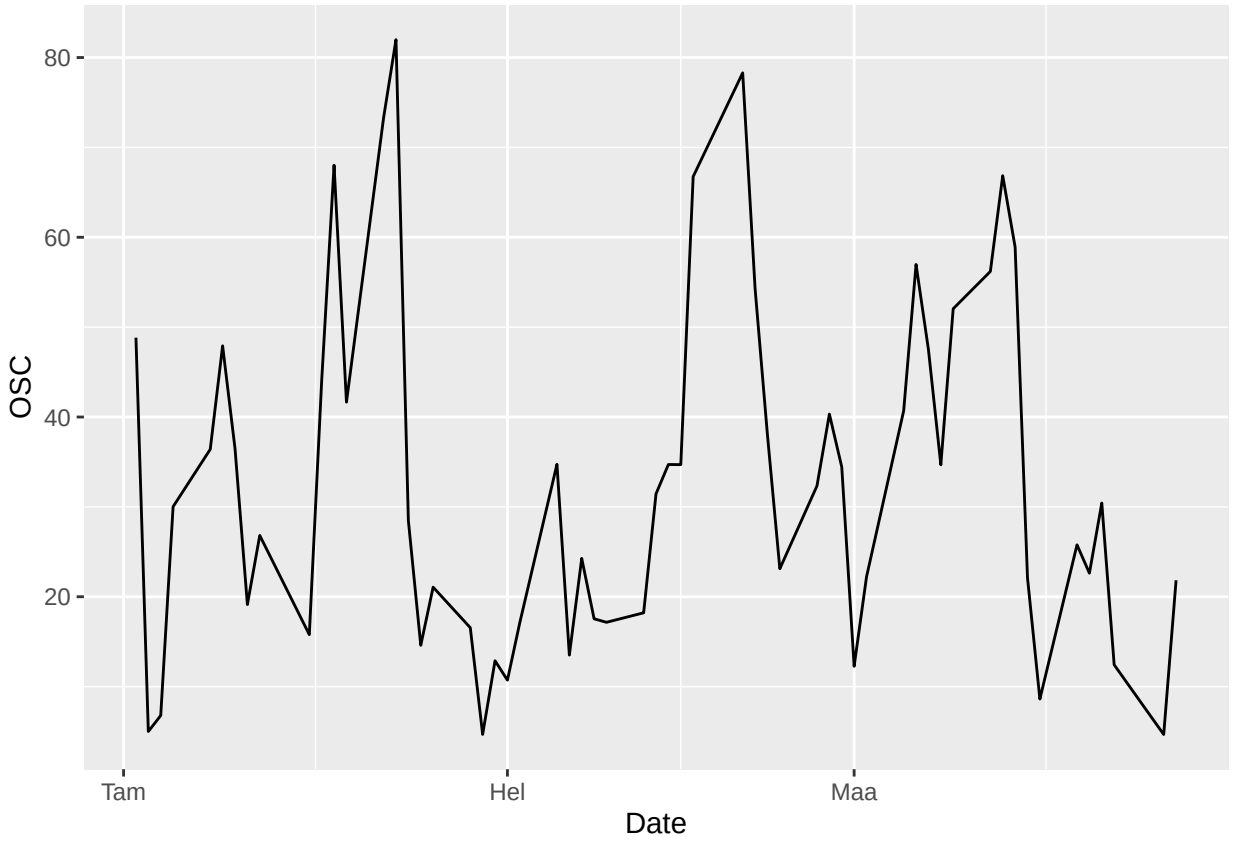
$$K = 100 * \frac{P_t - Low_K}{High_K - Low_K}$$

Stochastic Oscillator tries to predict turning point of price movement. Many times is said that asset is overbought when Stochastic Oscillator get values greater than 80 and oversold when it gets values below 20. It is important to remember that asset can remain overbought or oversold for long periods if the price of the asset is trending up or down. I have understood that this measure indicates possible overshooting in the markets.

```
stoch_osc <- function(data,t, K){
  P <- data[Date == t, Adj_Close_Smooth]
  dt <- data[Date <= t & Date > as.Date(t) - K, .(Adj_Close_Smooth, Adj_High,
              Adj_Low)]
  return(100*(P-min(dt$Adj_Low))/(max(dt$Adj_High)-min(dt$Adj_Low)))
}

K <- 14
price[, OSC := as.numeric(rbind(lapply(price$Date, function(t){stoch_osc(price,
      t, K)})))]

class(price$OSC)
```

```
## [1] "numeric"
```

```
ggplot(price[year(Date) > 2017], aes(Date, OSC)) + geom_line()
```



Moving Average Convergence Divergence

Moving Average Convergence Divergence calculates difference between two Moving Averages. I will use 12 and 26 day Exponential Moving Averages. Exponential moving average differences from normal moving average by giving more weight on recent observations.

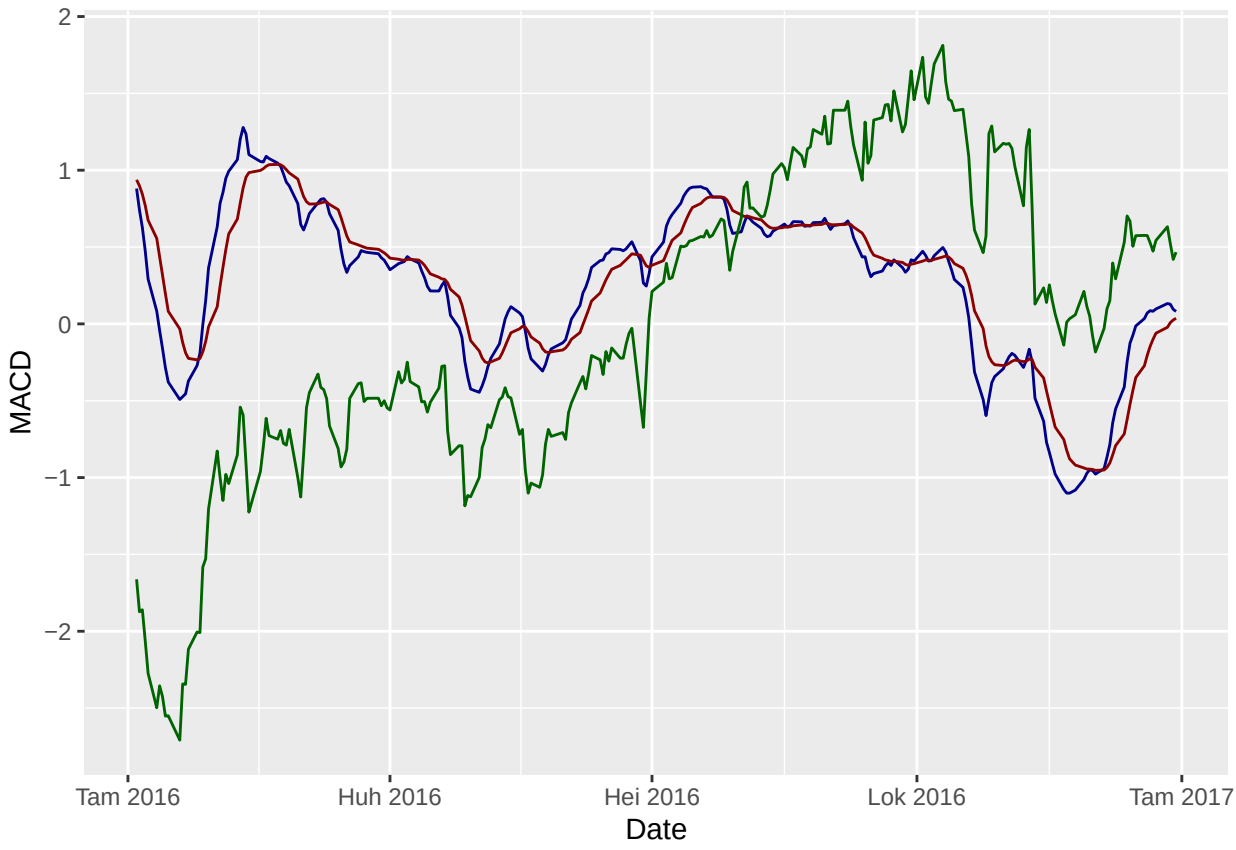
$$EMA = P_i(\frac{2}{n+1}) + EMA_{i-1}(1 - (\frac{2}{n+1}))$$

Where P_i is current closing price and i is the number of days. Then we can calculate our MACD and signal MACD.

$$MACD = EMA_{12} - EMA_{26}$$
$$SignalMACD = EMA_9(MACD)$$

Investor can get buying and selling signals by looking at intercepts of the MACD and it's signal line. Signal line is smoother than MACD line and reacts with lag to price changes.

```
price[, EMA12 := movavg(Adj_Close_Smooth, 12, type="e")]
price[, EMA26 := movavg(Adj_Close_Smooth, 26, type="e")]
price[, MACD := EMA12 - EMA26]
price[, SignalMACD := movavg(MACD, 9 , type ="e")]

ggplot(price[year(Date) == 2016], aes(x = Date)) + geom_line(aes(y = MACD), color = "darkblue") + geom_
```



Random forest