

# Coffee Future Price Prediction Project: ICFFUT B3 Dataset

HarvardX - PH125.9x Data Science: Capstone Course

Mauricio Rabelo Soares

03 set 2022

## Introduction

A Machine Learning for Factor Investing is a equity **investment strategies** that are built on **firm characteristics**, the goal of this approach is to determine the model that maps the time-t characteristics of firms to their future performance (Coqueret and Guida 2020).<sup>1</sup> Its premise is that differences in the returns of firms can be explained by the characteristics of these firms. In this project we use a similar approach but, instead a firm performance, our outcome is the trade decision taken at close call of 4/5 Arabica Coffee Futures contract traded at B3, which ticker is ICF.<sup>2</sup> The goal of this project is to create a decision trade system which balanced accuracy, or F1 score, is above 50%,<sup>3</sup> using all the tools we have learn throughout the multi-part course in HarvardX's Data Science Professional Certificate series. For this challenge we going to use a dataset provided by the Profit Trader Clear<sup>4</sup>. The dataset begin in January 2003 and has **4435 decisions** applied to **1 asset** with **20 features**.

## Methods

The methods section explains the process and techniques used in this project, in the first part, then explains the data cleaning used to extract and clean the data, in the second part. In the third part of this section we present the data exploration and visualization of the data to gain some insights. The fourth, and last part of this section, we show the modeling approach used in this project.

## The process and techniques used

The decision trade system is similar to a **decision support system (DSS)** which is an interactive software-based system intended to help decision makers (future contract trader) compile useful information from a combination of raw data, documents, and personal knowledge, or business models to identify and solve problems and make decisions<sup>5</sup>. The future trader decisions could be **buy** the contract, **sell** or not get in a

---

<sup>1</sup><http://www.mlfactor.com/index.html>

<sup>2</sup>[https://www.b3.com.br/en\\_us/products-and-services/trading/commodities/product-sheet-8AE490C96D41D3A2016D46017EC97262.htm](https://www.b3.com.br/en_us/products-and-services/trading/commodities/product-sheet-8AE490C96D41D3A2016D46017EC97262.htm)

<sup>3</sup>“The mean squared error is usually hard to interpret. It’s not easy to map an error on returns into the impact on investment decisions. The hit ratio is a more intuitive indicator because it evaluates the proportion of correct guesses (and hence profitable investments). Obviously, it is not perfect: 55% of small gains can be mitigated by 45% of large losses. Nonetheless, it is a popular metric and moreover it corresponds to the usual accuracy measure often computed in binary classification exercises. Here, an accuracy of 0.542 is satisfactory. Even if any number above 50% may seem valuable, it must not be forgotten that transaction costs will curtail benefits. Hence, the benchmark threshold is probably at least at 52%.” (Coqueret and Guida 2020)

<sup>4</sup><https://corretora.clear.com.br/plataformas/profit-trader-clear/>

<sup>5</sup>[https://en.wikipedia.org/wiki/Decision\\_support\\_system](https://en.wikipedia.org/wiki/Decision_support_system)

position, what is the same to be **neutral** in a asset, this decisions will be our **outcome**  $\mathbf{y} = y_t$ , which is categorical.

Every trade decision taken, in a future market, occur in a certain time  $t$ , in a specific contract and organized marked, for a unique value. For this project the contract is the 4/5 Arabica Coffee Futures traded at B3, which ticker is ICF, and are available for march, may, july, september and december. The ICFFUT is the representative time-series of current contract and some of thus features will be our the dataset. The dimension of the feature matrix  $X$  is  $T \times K$ : where  $T$  are daily **observations** and each one of them has  $K$  **features**, **inputs**, or **predictors** which will serve as **independent** and **explanatory** variables.

The decision trade system build in this project are made using a similar process and techniques presented at (Coqueret and Guida 2020) and some approaches used by Renaissance Technologies (Zuckerman, 2019).<sup>6</sup> The backtesting protocol propose by (Arnott, Harvey, and Markowitz 2018) are used as reference to evaluate the model.

The process begin with the download, then read the data and create some extensions using the Close price. After the data cleaning we explore the data to gain some insight throughout visualization and selected table. This insights is the basis of the modelling approach of this project, that have 3 non linear algorithms: k-Nearest Neighbour Classification, Recursive Partitioning and Classification Trees, and Classification and Regression with Random Forest

## Data cleaning

For this project we going to use a subset of dataset provided by Profit Trader Clear, and for the sake of **reproducibility**, we will illustrate the concepts based on a single financial dataset available at <https://github.com/mrabelosoares/Coffee-Future-Price-Prediction>. This dataset comprises information on 1 contract listed at B3, which the time range starts in January 2003 and ends in August 2022. For each point in time, 19 **characteristics** describe the decision in the sample. The dataset are divide between raw data and extensions:

**Raw data:** Date; Asset; Price (Open, High, Low, Close); Volume; Decision; Open Interest

```
library(caret)
library(data.table)
library(fields)
library(tidyverse)
library(knitr)
library(kableExtra)
library(grid)
library(ggplot2)
library(lattice)
library(gridExtra)
library(readxl)
library(dplyr)
library(purrr)
library(zoo)
library(runner)
library(quantmod)
library(rpart)
library(randomForest)
library(xts)
library(TTR)
```

---

<sup>6</sup><https://www.youtube.com/watch?v=lji-jNsXmAM>  
Renaissance Technologies - Trading Strategies Revealed | A Documentary

```
library(corrplot)
library(rpart.plot)
```

### *##Data Clean*

```
#create tempfile and download
dl <- tempfile()
download.file("https://github.com/mrabelosoares/Coffee-Future-Price-Prediction/blob/2044135ec7c5bcfb6f3...")
#read file XLSX format with decisions
ICFFUT <- read_xlsx("CoffeDatabase.xlsx", sheet = "ICFFUT")
#read file XLSX format with ICF Prices
ICFFUT_XLSX <- read_xlsx("CoffeDatabase.xlsx", sheet = "ICFFUT_XTS")
#Convert XLSX format to Data Frame
DFICFFUT_XTS <- as.data.frame(ICFFUT_XLSX)
#Convert Data Frame to XTS
ICFFUT_XTS <- xts(DFICFFUT_XTS[-1], order.by = as.Date(DFICFFUT_XTS$Date))
```

**Extensions:** moving average 5 days; moving average 22 days; Bollinger Bands; Rate of Change Oscillator; Relative Strength Index; Stochastic Momentum Index; MACD Oscillator. The description and equations of technical indicator based on price are available at reference manual: <https://cran.r-project.org/web/packages/TTR/TTR.pdf>

### *##Technical Indicators - Price-Based*

```
#moving average 5 days
Moving_Average_5 <- SMA(ICFFUT_XTS$ICFFUT.Close, n=5)
#moving average 22 days
Moving_Average_22 <- SMA(ICFFUT_XTS$ICFFUT.Close, n=22)
#Bollinger Bands
Bollinger_Bands <- BBands(ICFFUT_XTS$ICFFUT.Close)
#Rate of Change Oscillator
Rate_Change_Oscillator <- ROC(ICFFUT_XTS$ICFFUT.Close, n=10)
#Relative Strength Index
Relative_Strength_Index <- RSI(ICFFUT_XTS$ICFFUT.Close)
#Stochastic Oscillator / Stochastic Momentum Index
Stochastic <- stoch(ICFFUT_XTS$ICFFUT.Close)
#MACD Oscillator
MACD <- MACD(ICFFUT_XTS$ICFFUT.Close)
```

The raw data and the extensions are merged to create the complete data frame.

```
#merge XTS
Full_ICFFUT <- merge(ICFFUT_XTS,
                     Moving_Average_5,
                     Moving_Average_22,
                     Bollinger_Bands,
                     Rate_Change_Oscillator,
                     Relative_Strength_Index,
                     Stochastic,
                     MACD)
#create data frame ICFFUT - 4/5 Arabica Coffee Futures
DFICFFUT <- data.frame(Date=index(Full_ICFFUT), coredata(Full_ICFFUT))
#Merge Data frame ICFFUT - 4/5 Arabica Coffee Futures and ICFFUT
```

```
CDFICFFUT <- left_join(DFICFFUT,
                        ICFFUT |> select(Date, Decision, OpenInterest)) |>
  filter(Date > "2003-02-17") |> #excluding NA
  mutate(Decision = as.factor(Decision)) #Decision as a factor = Y
```

The extensions are technical indicators based on a daily Close Price of ICF and are select based on Chartered Financial Analyst (CFA®) program curriculum. This choice make sense economically since this knowledge are widely disseminated to the market and the decisions are made at the closing call every day.

The decisions was build manually by the author, which look the daily chart and choose the optimal decision every day. As present in the model of chapter 27.8 Case study: is it a 2 or a 7? (Irizarry 2019) the goal of this approach is to look at a picture and classifies what is the best decision at the closing call every day: Buy, Sell, Neutral. Is there a pattern inside the own prices of ICFFUT that bring some prediction?

## Data exploration and visualization

The exploration and visualization of the data provide insightful information about the behavior of price throughout time. The graphic below show the price range between 2003 and 2022, as we can see the volatility was different for different years.

```
##Data exploration and visualization
#Chart ICFFUT Price
chart_Series(ICFFUT_XTS,type = "candlesticks")
```



A functional trade system implies that every day has at least one trade, in this case the days in which volume are zero we consider a no trade day. As presented below we have 4 days in this condition. For sake of simplicity we going to maintain this days because even without trade the adjust occur.

```
#Day without trade
Notrade <- CDFICFFUT |>
  filter(ICFFUT.Volume == "0") |>
  select(Date, ICFFUT.Close, ICFFUT.Volume, Decision)
Notrade
```

```
##           Date ICFFUT.Close ICFFUT.Volume Decision
## 1 2008-03-06      760.81           0      Sell
## 2 2011-01-25      886.97           0       Buy
## 3 2011-03-07     1046.24           0       Buy
## 4 2011-03-08     1052.12           0       Buy
```

The proprieties of dataset reveal the possibilities and limitations of model approach. Lets check the class, the summary and the structure of data.

```
#class, type and proprieties of data frame
as_tibble(CDFICFFUT)
```

```
## # A tibble: 4,435 x 21
##   Date           ICFFU~1 ICFFU~2 ICFFU~3 ICFFU~4 ICFFU~5   SMA SMA.1   dn
##   <dtm>          <dbl>  <dbl>  <dbl>  <dbl>  <dbl> <dbl> <dbl> <dbl>
## 1 2003-02-18 00:00:00    568.   572.   562.   562. 131191.  573.  598.  562.
## 2 2003-02-19 00:00:00    565.   565.   549.   559. 199476.  566.  595.  557.
## 3 2003-02-20 00:00:00    559.   567.   559.   564. 171859.  565.  594.  554.
## 4 2003-02-21 00:00:00    563.   563.   553.   555.  70105.  562.  590.  550.
## 5 2003-02-24 00:00:00    555.   555.   534.   536. 211943.  555.  587.  542.
## 6 2003-02-25 00:00:00    536.   541.   518.   519. 182403.  547.  582.  530.
## 7 2003-02-26 00:00:00    519.   519.   502.   510. 128849.  537.  578.  518.
## 8 2003-02-27 00:00:00    509.   517.   505.   513.  81067.  527.  574.  508.
## 9 2003-02-28 00:00:00    513.   515.   494.   495.  87501.  515.  569.  496.
## 10 2003-03-05 00:00:00    492.   508.   492.   508. 12552.  509.  566.  489.
## # ... with 4,425 more rows, 12 more variables: mavg <dbl>, up <dbl>,
## #   pctB <dbl>, ICFFUT.Close.1 <dbl>, rsi <dbl>, fastK <dbl>, fastD <dbl>,
## #   slowD <dbl>, macd <dbl>, signal <dbl>, Decision <fct>, OpenInterest <dbl>,
## #   and abbreviated variable names 1: ICFFUT.Open, 2: ICFFUT.High,
## #   3: ICFFUT.Low, 4: ICFFUT.Close, 5: ICFFUT.Volume
```

```
summary(CDFICFFUT)
```

```
##           Date           ICFFUT.Open           ICFFUT.High
## Min.      :2003-02-18 00:00:00.00 Min.      : 119.4 Min.      : 121.0
## 1st Qu.:2008-02-21 12:00:00.00 1st Qu.: 271.6 1st Qu.: 275.0
## Median :2013-08-28 00:00:00.00 Median : 470.6 Median : 475.9
## Mean    :2013-03-28 08:47:56.75 Mean    : 472.7 Mean    : 478.1
## 3rd Qu.:2018-02-22 12:00:00.00 3rd Qu.: 666.8 3rd Qu.: 675.1
## Max.    :2022-08-19 00:00:00.00 Max.    :1202.8 Max.    :1207.6
## ICFFUT.Low ICFFUT.Close ICFFUT.Volume SMA
## Min.      : 118.2 Min.      : 119.6 Min.      : 0 Min.      : 120.7
```

```

## 1st Qu.: 268.2    1st Qu.: 271.0    1st Qu.: 3779024    1st Qu.: 271.8
## Median : 466.0    Median : 470.3    Median : 14741320    Median : 473.0
## Mean   : 467.6    Mean   : 472.7    Mean   : 23839784    Mean   : 472.8
## 3rd Qu.: 660.4    3rd Qu.: 668.1    3rd Qu.: 31069015    3rd Qu.: 668.8
## Max.   :1198.0    Max.   :1206.0    Max.   :450228233    Max.   :1192.4
##      SMA.1          dn          mavg          up
## Min.    : 122.6    Min.    : 113.8    Min.    : 122.7    Min.    : 127.3
## 1st Qu.: 275.3    1st Qu.: 258.1    1st Qu.: 274.9    1st Qu.: 291.0
## Median : 474.8    Median : 436.6    Median : 474.5    Median : 500.0
## Mean   : 473.5    Mean   : 442.3    Mean   : 473.4    Mean   : 504.5
## 3rd Qu.: 673.6    3rd Qu.: 625.0    3rd Qu.: 673.5    3rd Qu.: 709.8
## Max.   :1143.8    Max.   :1085.5    Max.   :1149.3    Max.   :1256.7
##      pctB          ICFFUT.Close.1          rsi          fastK
## Min.    :-0.3313    Min.    :-0.242930    Min.    : 9.05    Min.    :0.00000
## 1st Qu.: 0.1967    1st Qu.: -0.042496    1st Qu.:39.26    1st Qu.:0.07407
## Median : 0.4276    Median : -0.005513    Median :47.42    Median :0.39871
## Mean   : 0.4759    Mean   : -0.001844    Mean   :48.86    Mean   :0.45748
## 3rd Qu.: 0.7644    3rd Qu.: 0.036730    3rd Qu.:58.13    3rd Qu.:0.85201
## Max.   : 1.4078    Max.   : 0.346360    Max.   :92.60    Max.   :1.00000
##      fastD          slowD          macd          signal
## Min.    :0.0000    Min.    :0.0000    Min.    :-7.2943    Min.    :-6.8434
## 1st Qu.:0.1122    1st Qu.:0.1215    1st Qu.: -1.6575    1st Qu.: -1.5510
## Median :0.3937    Median :0.3932    Median : -0.4381    Median : -0.4210
## Mean   :0.4574    Mean   :0.4573    Mean   : -0.1715    Mean   : -0.1705
## 3rd Qu.:0.8249    3rd Qu.:0.8103    3rd Qu.: 1.3178    3rd Qu.: 1.2582
## Max.   :1.0000    Max.   :1.0000    Max.   :11.0073    Max.   :10.0766
##      Decision      OpenInterest
## Buy    :1662    Min.    : 0
## Neutral: 665    1st Qu.: 0
## Sell   :2108    Median : 3739
##                               Mean   : 3514
##                               3rd Qu.: 5164
##                               Max.   :21781

```

```
str(CDFICFFUT)
```

```

## 'data.frame':    4435 obs. of  21 variables:
## $ Date           : POSIXct, format: "2003-02-18" "2003-02-19" ...
## $ ICFFUT.Open     : num  568 565 559 563 555 ...
## $ ICFFUT.High     : num  572 565 567 563 555 ...
## $ ICFFUT.Low      : num  562 549 559 553 534 ...
## $ ICFFUT.Close    : num  562 559 564 555 536 ...
## $ ICFFUT.Volume   : num  131191 199476 171859 70105 211943 ...
## $ SMA             : num  573 566 565 562 555 ...
## $ SMA.1           : num  598 595 594 590 587 ...
## $ dn              : num  562 557 554 550 542 ...
## $ mavg            : num  597 594 591 588 584 ...
## $ up              : num  632 630 627 625 627 ...
## $ pctB            : num  0.0107 0.0262 0.1368 0.0604 -0.0655 ...
## $ ICFFUT.Close.1 : num  -0.06 -0.0526 -0.0658 -0.0891 -0.1325 ...
## $ rsi             : num  39.6 38.2 41.5 37.7 31.7 ...
## $ fastK           : num  0 0 0.103 0 0 ...
## $ fastD           : num  0.0505 0.0135 0.0345 0.0345 0.0345 ...
## $ slowD           : num  0.0819 0.0382 0.0328 0.0275 0.0345 ...

```

```
## $ macd          : num  -0.785 -1.081 -1.227 -1.462 -1.893 ...
## $ signal        : num   0.2991 0.0231 -0.227 -0.4741 -0.7579 ...
## $ Decision      : Factor w/ 3 levels "Buy","Neutral",...: 3 3 3 3 3 3 3 3 3 3 ...
## $ OpenInterest  : num   0 0 0 0 0 0 0 0 0 0 ...
```

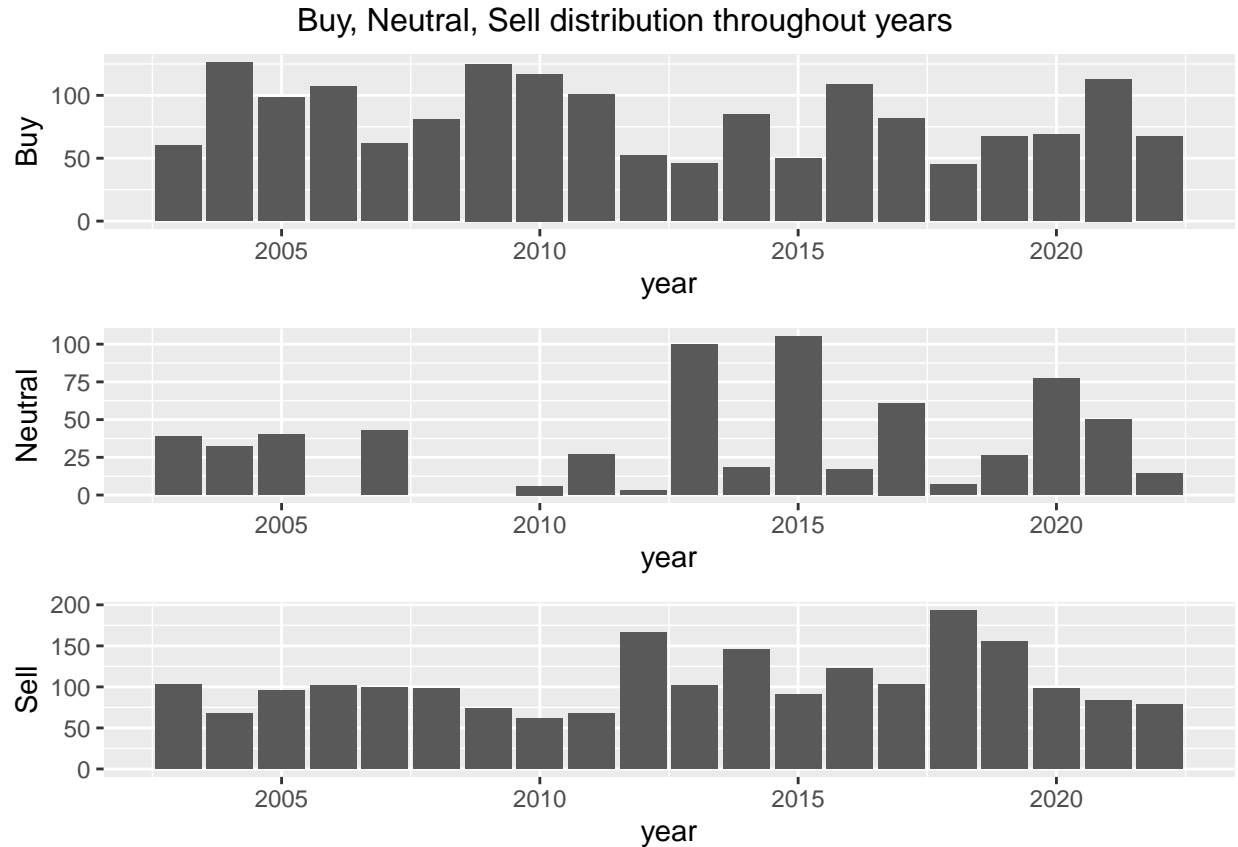
As presented above our matrix have  $4,435 \times 21$  variables, mostly numerical column. The close price, the principal column, has 119.6 as minimal value and 1206.0 as maximal, and a median of 470.3. The decision column, our outcome, has a unbalanced prevalence: *Buy* = 1662, *Neutral* = 665, *Sell* = 2108. Lets explore the distribution of decisions throughout years.

```
#distribution decision by year
p1 <- CDFICFFUT |>
  mutate(year = year(as.Date(Date)))|>
  filter(Decision == "Buy") |>
  ggplot(aes(x = year)) +
  geom_bar(stat="count") +
  ylab("Buy")

p2 <- CDFICFFUT |>
  mutate(year = year(as.Date(Date)))|>
  filter(Decision == "Neutral") |>
  ggplot(aes(x = year)) +
  geom_bar(stat="count") +
  ylab("Neutral")

p3 <- CDFICFFUT |>
  mutate(year = year(as.Date(Date)))|>
  filter(Decision == "Sell") |>
  ggplot(aes(x = year)) +
  geom_bar(stat="count") +
  ylab("Sell")

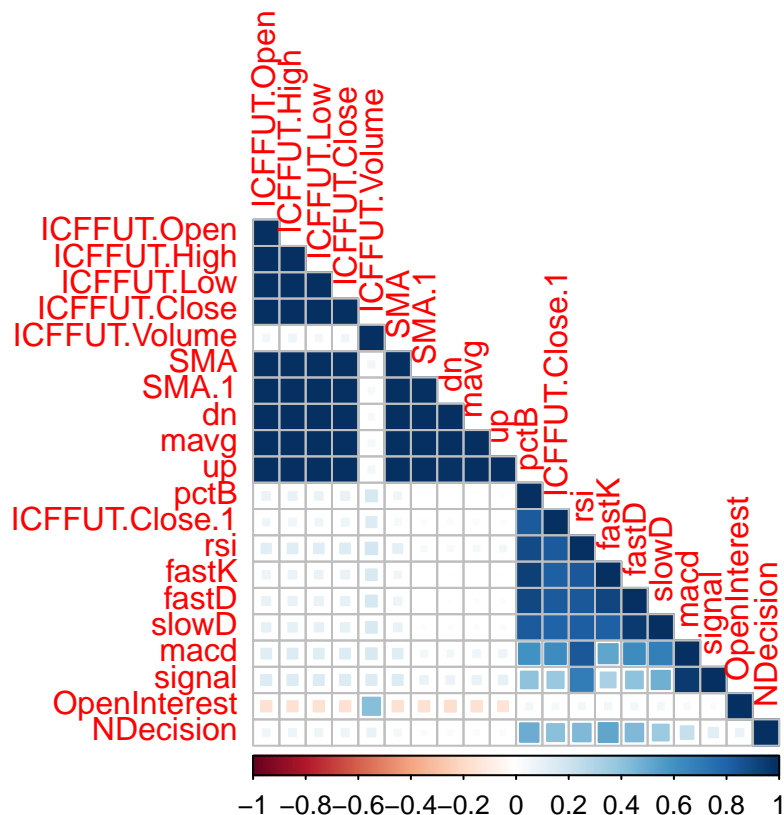
#Buy, Neutral, Sell distribution throughout years
gridExtra::grid.arrange(p1, p2, p3,
  nrow = 3,
  top = "Buy, Neutral, Sell distribution throughout years")
```



The graphs show different prevalence for different years, this is a relevant information for building a general model, since the prevalence affects the precision and recall of the model, and consequently the measurement of performance. The correlation between variables and outcome could bring more information about the behavior of the data, which reflects the decisions of traders. To transform the decision from a factor to a number we replace *Neutral* by 0, *Buy* for 1, and *Sell* for -1.

```
#correlation between variables and outcome
NDFICFFUT <- CDFICFFUT |> mutate( #Decision as a number
  NDecision = case_when(
    Decision == "Neutral" ~ 0, #If Neutral, then 0
    Decision == "Buy" ~ 1, #If Buy, then 1
    Decision == "Sell" ~ -1) #If sell, then -1
) |> select(-Decision, -Date) #less categorical and date column
C <- cor(NDFICFFUT) #correlation matrix
corrplot(C, method = 'square', type = 'lower')
```





The outcome are high correlated with extension fastK, which is 1 of 3 components of Stochastic Momentum Index. On the other side, the prices, the moving averages and the Bollinger Bands are low correlated with decisions. The volume has some correlations with Open interest, but almost nothing with others variables.

The result of decisions made throughout years, or the sum of profits, could be calculated using the code below. Considering the decision made at every day close call the potential return are the difference between today close price less the yesterday close price. Another important feature in this model is the status of daily position, which could be hold or change, either *Neutral*, *Buy* or *Sell*.

The combination of status and current, or lag decision, determine the multiplier we use to calculate the adjust we going to receive or pay. Lets show a example to illustrate the idea, my decision last trade day was to be *Neutral*, and today is *Buy*, which means a change, the multiplier will be 0, because we have no adjust to pay or receive this day. In the next day we hold the *Buy* position, then or multiplier will be 1 and we receive the positive change in price or we pay the negative difference in price. To complete the model we measure the sum of profit, daily adjust payed or received, for the last 22 days, which is a proxy of the monthly profit.

```
#empirical optimal results - sum last 22 days
profitICFFUT <- CDFICFFUT |>
  arrange(Date) |>
  mutate(return = ICFUT.Close - lag(ICFFUT.Close) ,
         status = as.factor(case_when(
           Decision == lag(Decision) ~ "Hold",
           Decision != lag(Decision) ~ "Change")),
         multiplier = case_when(
           status == "Change" & lag(Decision) == "Neutral" ~ 0,
           status == "Hold" & Decision == "Neutral" ~ 0,
```

```

status == "Change" & lag(Decision) == "Buy" ~ 1,
status == "Hold" & Decision == "Buy" ~ 1,
status == "Change" & lag(Decision) == "Sell" ~ -1,
status == "Hold" & Decision == "Sell" ~ -1),
adjust = return * multiplier,
profit = case_when(
  Decision == "Neutral" ~ 0,
  Decision != "Neutral" ~ sum_run(adjust, k=22)
))
summary(profitICFFUT |> select(return, status, adjust, profit))

```

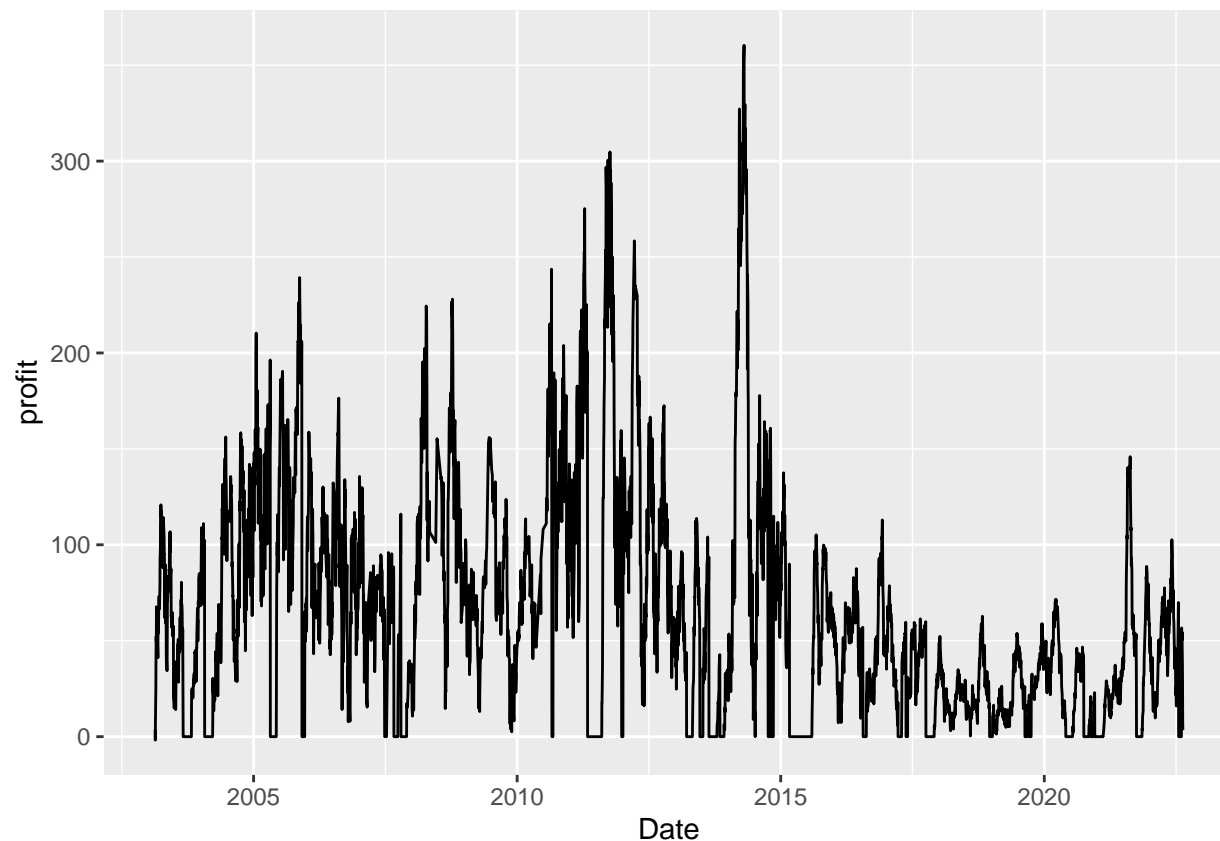
##	return	status	adjust	profit
##	Min. : -71.42000	Change: 245	Min. : -34.590	Min. : -1.83
##	1st Qu.: -4.11500	Hold : 4189	1st Qu.: -0.390	1st Qu.: 19.48
##	Median : -0.09500	NA's : 1	Median : 1.070	Median : 51.70
##	Mean : -0.06865		Mean : 3.179	Mean : 64.28
##	3rd Qu.: 3.99000		3rd Qu.: 5.838	3rd Qu.: 95.08
##	Max. : 59.60000		Max. : 71.420	Max. : 360.34
##	NA's : 1		NA's : 1	NA's : 1

The daily change, return, show the volatility of the asset, which reached 71.42 in one day, and has a median and mean below zero, which is compatible with the theory that the future price converge to spot price, or the tendency to lost value throughout time. The status was changed in 245 periods, or 5,5% of the days, this a important feature of the model that intended to minimize the trade cost. The profit, the ultimate goal of decision trade system, are consistently above zero and has median of 51.70 for the last 22 days. The annual profit could be verified in the table below, and show this median are too high for the last 5 years.

```

#profit adjust from the last 22 days
profitICFFUT |>
  ggplot(aes(Date, profit)) +
  geom_line()

```



```
#profit by year
profitY <- profitICFFUT |> mutate(Year = as.factor(year(as.Date(Date))))
profit_table <- as.data.frame(tapply(profitY$adjust,
                                   profitY$Year,
                                   FUN=sum,
                                   na.rm = TRUE))
colnames(profit_table) <- "profit_year"
profit_table
```

```
##      profit_year
## 2003      479.80
## 2004      843.87
## 2005     1318.64
## 2006      846.63
## 2007      559.28
## 2008      944.44
## 2009      618.12
## 2010     1013.97
## 2011     1361.92
## 2012     1011.10
## 2013      503.53
## 2014     1511.67
## 2015      536.17
## 2016      537.58
## 2017      394.78
```

## 2018	246.49
## 2019	270.64
## 2020	298.84
## 2021	464.89
## 2022	333.93

## Modeling approach

The decision trade system is better as its errors has decrease, for this project the error has a categorical metrics evaluation. The summary metrics for this project is the F1-score, or balanced accuracy, that is the harmonic average of precision and recall. As presented in 27.4.5 Balanced accuracy and F1 score (Irizarry 2019) “The  $F1$  -score can be adapted to weigh specificity and sensitivity differently. To do this, we define  $\beta$  to represent how much more important sensitivity is compared to specificity and consider a weighted harmonic average”. For this project the  $\beta = 1$ , because the sensitivity has the same importance of specificity:

$$F1 = \frac{1}{\frac{\beta^2}{1+\beta^2} \frac{1}{\text{recall}} + \frac{1}{1+\beta^2} \frac{1}{\text{precision}}}$$

For this project the  $F1 > 0.50$  is the goal. The value was taken from (Coqueret and Guida 2020) where was suggested thata any accuracy above 50% may seem valuable.

## Machine Learning

*“If we have enough data, I know we can make predictions,” James Harris Simons<sup>7</sup>*

The machine learning decisions are based on algorithms build with data, so for this project the dataset CDFICFFUT are going to be used to train and test the model. The training\_sample and testing\_sample are build by slicing the time, where the sample before 2020 are the training set and after the test set. The proportion of data for testing is 17% of total, in line with typical choice of 10%-20%(Irizarry 2019):

```
#Modeling approach
#create a partition
separation_date <- as.Date("2020-01-02")
training_sample <- filter(CDFICFFUT, Date < separation_date)
testing_sample <- filter(CDFICFFUT, Date >= separation_date)
```

## k-Nearest Neighbor Classification approach

The first model in this project is the k-nearest neighbors (kNN) approach, used by Renaissance Technologies and presented at chapter 29.1 Motivation with k-nearest neighbors (Irizarry 2019). The multiple dimensions adaptability is a important feature, specially in finance, to estimate  $p(x_1, x_2)$ :

$$p(x_1, x_2) = \Pr(Y = 1 \mid X_1 = x_1, X_2 = x_2).$$

As presented “for any point  $(x_1, x_2)$  for which we want an estimate of  $p(x_1, x_2)$ , we look for the k nearest points to  $(x_1, x_2)$  and then take an average of the 0s and 1s associated with these points. We refer to the set of points used to compute the average as the *neighborhood*.” (Irizarry 2019). We going to use the knn3 function of caret package, with a  $k = 5$ , to train the dataset. Then we show the first 5 rows of model by probability and class of decision.

---

<sup>7</sup>Zuckerman, Gregory. The Man Who Solved the Market (p. 2). Penguin Publishing Group. Kindle edition

```

#defining the predictors - Model 1
knn_fit <- knn3(Decision ~ .,
               data = training_sample, k=5)
knn_fit

## 5-nearest neighbor model
## Training set outcome distribution:
##
##      Buy Neutral   Sell
##      1413      524   1847

#probability and class of model
head(predict(knn_fit, testing_sample, type = "prob"))

##      Buy Neutral Sell
## [1,] 0.6      0 0.4
## [2,] 0.0      0 1.0
## [3,] 0.6      0 0.4
## [4,] 0.6      0 0.4
## [5,] 0.6      0 0.4
## [6,] 0.4      0 0.6

head(predict(knn_fit, testing_sample, type = "class"))

## [1] Buy  Sell Buy  Buy  Buy  Sell
## Levels: Buy Neutral Sell

#balanced accuracy - model 1
y_hat_knn <- predict(knn_fit, testing_sample, type = "class")
confusionMatrix(y_hat_knn, testing_sample$Decision)$overall["Accuracy"]

## Accuracy
## 0.4285714

cm1 <- confusionMatrix(y_hat_knn, testing_sample$Decision)
cm1[["byClass"]][ , "Precision"]

##      Class: Buy Class: Neutral   Class: Sell
##      0.4654545      NA      0.4015957

cm1[["byClass"]][ , "Recall"]

##      Class: Buy Class: Neutral   Class: Sell
##      0.5140562      0.0000000      0.5785441

KNN <- cm1[["byClass"]][ , "F1"]
KNN

##      Class: Buy Class: Neutral   Class: Sell
##      0.4885496      NA      0.4740973

```

The accuracy of 0.4285714 is better than guessing, which is 0.3333, but is below our goal of 0.5. The F1 - score is below 0.5 for *Buy* and *Sell* decisions, and is zero for *Neutral*. This could be a consequence of low prevalence of *Neutral*, or undetected pattern related with this decision.

## Recursive Partitioning for classification trees approach

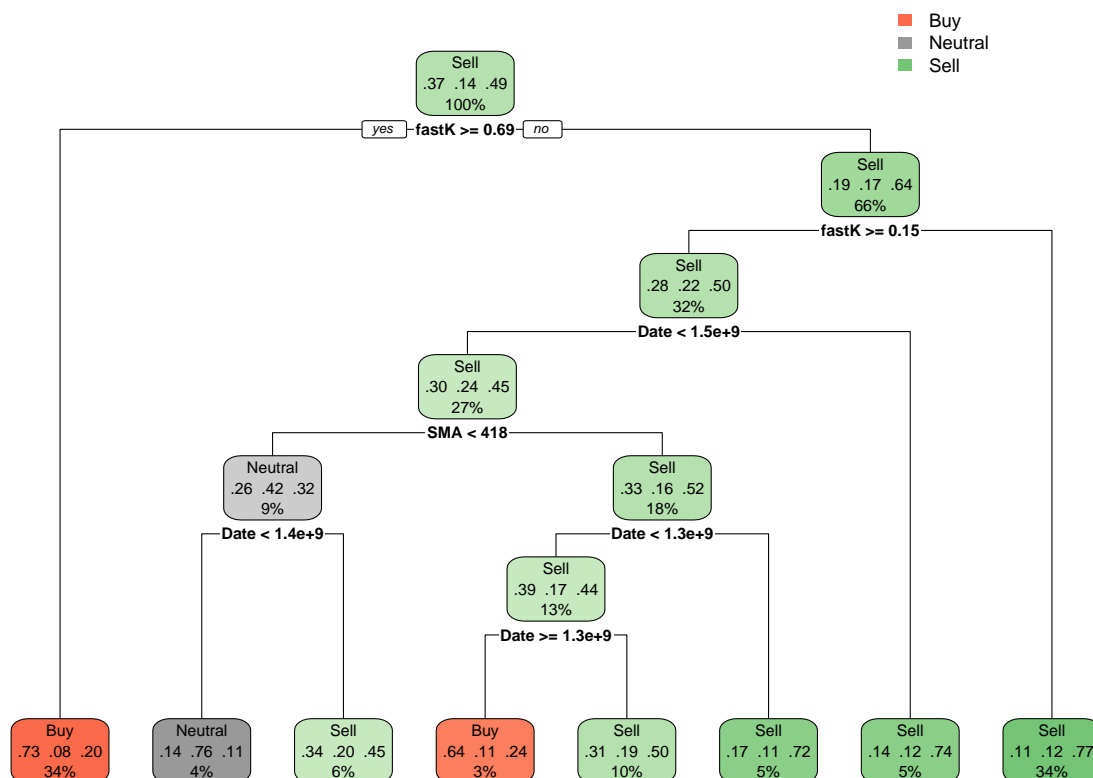
The tree based method, presented in chapter 6 (Coqueret and Guida 2020), is our second model, and was chosen because has efficient forecast, are easy to visualize and could model human decision. The goal of classification tree is to split the dataset into homogeneous cluster, by minimizing dispersion inside each cluster. The output show the proportion of each class, in each cluster, in this case for  $J$  classes, we denote these proportions with  $p_j$ , and for each cluster  $k$ , the usual loss functions are:

$$\text{Gini}(j) = 1 - \sum_{j=1}^J p_j^2;$$

$$\text{entropy}(j) = - \sum_{k=1}^K \hat{p}_{j,k} \log(\hat{p}_{j,k}), \text{ with } 0 \times \log(0) \text{ defined as } 0$$

The first split is the most important of the model because show the most general rule of data aggregation and reveal the most relevant step to taken in trade decision. To implement this classification tree we going to use rpart function in the rpart package, with default complexity parameter.

```
#defining the predictors - Model 2
fit <- rpart(Decision ~ ., data = training_sample)
rpart.plot(fit)
```



```
#Structural break
as.POSIXlt(1.514808e+09, origin="1970-01-01")
```

```
## [1] "2018-01-01 10:00:00 -02"
```

```
as.POSIXlt(1.438603e+09,origin="1970-01-01")
```

```
## [1] "2015-08-03 08:56:40 -03"
```

```
as.POSIXlt(1.319285e+09,origin="1970-01-01")
```

```
## [1] "2011-10-22 10:03:20 -02"
```

```
as.POSIXlt(1.251158e+09,origin="1970-01-01")
```

```
## [1] "2009-08-24 20:53:20 -03"
```

```
#accuracy - model 2
```

```
y_hat_RT <- predict(fit, testing_sample, type = "class")
confusionMatrix(y_hat_RT, testing_sample$Decision)$overall["Accuracy"]
```

```
## Accuracy
```

```
## 0.5176651
```

```
cm2 <- confusionMatrix(y_hat_RT, testing_sample$Decision)
cm2[["byClass"]][ , "Precision"]
```

```
##      Class: Buy Class: Neutral Class: Sell
##      0.5476190          NA      0.4987469
```

```
cm2[["byClass"]][ , "Recall"]
```

```
##      Class: Buy Class: Neutral Class: Sell
##      0.5542169      0.0000000      0.7624521
```

```
Classification_Tree <- cm2[["byClass"]][ , "F1"]
Classification_Tree
```

```
##      Class: Buy Class: Neutral Class: Sell
##      0.5508982          NA      0.6030303
```

The general rule of trade decision in ICFFUT is to be *Sold*, but if the fastK is bigger than 0.69 the best decision is to change the decision to *Buy*. The *Buy* has a proportion of 34% of decisions and has a 73% of probability of being right in this decision, this correspond to a F1-score of 0.55. The *Sell* decision and *Neutral* decision has structural breaks<sup>8</sup>, which means that decisions change throughout time. The *Sell* decision has a F1-score of 0.60, much higher than our goal, but the *Neutral* decision has no F1-score. This reveal the limitations of this approach, since the prevalence differ by class and by year, or could indicate the necessity to review the parameters.

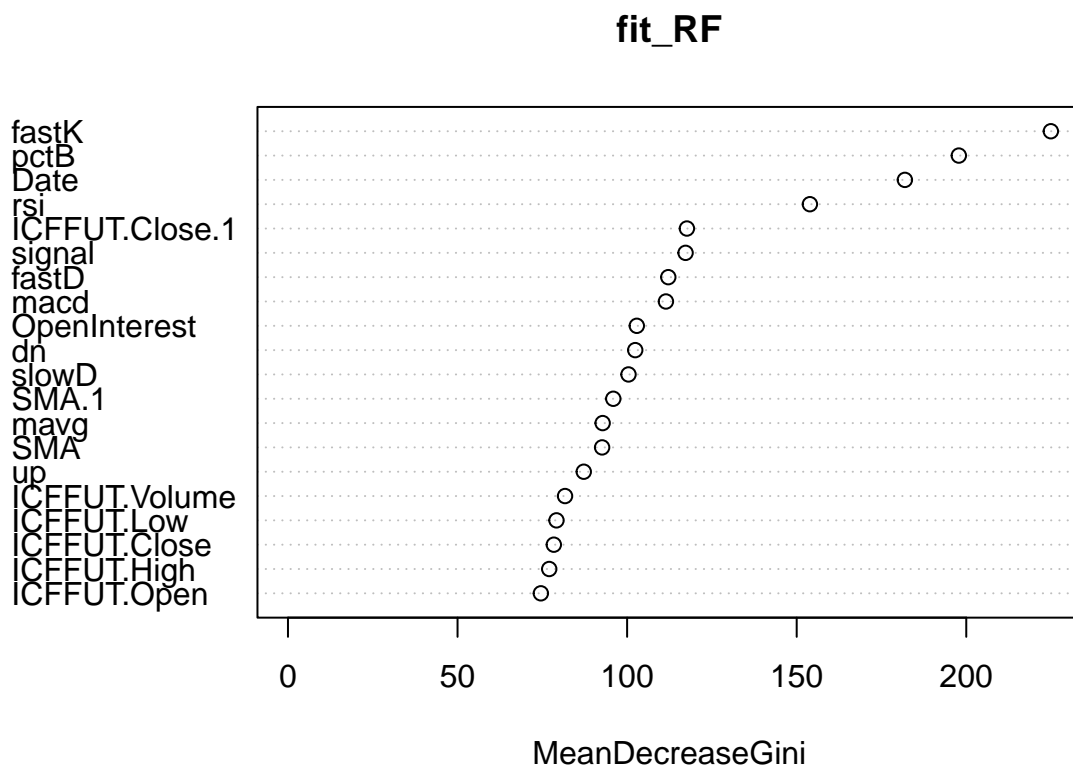
---

<sup>8</sup>[https://en.wikipedia.org/wiki/Structural\\_break](https://en.wikipedia.org/wiki/Structural_break)

## Classification and Regression with Random Forest approach

A Random Forest is the last approach to solve the trade decision system problem presented in this project. The goal of this approach is to take the averaging of multiples simple trees to reduce the instability and improve the prediction performance, and the algorithm do this by bootstrapping the sample to induce randomness. The Random Forest used in this project follow the default setup, but the algorithm has more than 27 arguments and has 17 components values, which provides a wide range of tuning possibilities.

```
#defining the predictors - Model 3
fit_RF <- randomForest(Decision ~., data = training_sample)
#Variable Importance Plot
varImpPlot(fit_RF)
```



```
#accuracy - model 3
y_hat_RF <- predict(fit_RF, testing_sample, type = "class")
confusionMatrix(y_hat_RF, testing_sample$Decision)$overall["Accuracy"]
```

```
## Accuracy
## 0.4669739
```

```
cm3 <- confusionMatrix(y_hat_RF, testing_sample$Decision)
cm3[["byClass"]][ , "Precision"]
```

```
## Class: Buy Class: Neutral Class: Sell
## 0.4133333 0.2500000 0.5189504
```



```
cm3[["byClass"]][ , "Recall"]
```

```
##      Class: Buy Class: Neutral   Class: Sell
##      0.4979920    0.0141844    0.6819923
```

```
Random_Forest <- cm3[["byClass"]][ , "F1"]
Random_Forest
```

```
##      Class: Buy Class: Neutral   Class: Sell
##      0.45173042    0.02684564    0.58940397
```

The model reveal that the most important variable is the fastK, followed by pctB, which is a component of Bollinger Bands indicator and quantifies the price relative to the upper and lower Bollinger Band. The Date is a important variable too, showing that model most to adapt to different time frame. The F1 score of *Neutral* decision is 0.013, which is very low, but it's not null, showing some improvement regarding the two models before. The *Sell* decision has a F1 score of 0.58, above our goal, but smaller than the model of tree classification. The improvement in *Neutral* decision come with a cost to *Buy* decision that has a F1 score of 0.45 for this model, below our goal.

## Results

This section presents the modeling results and discusses the performance of each approach. The table below shows the F1 score by decision and by approach.

```
#Results
knitr::kable(tibble(Decision = c("Buy", "Neutral", "Sell"),
                        KNN,
                        Classification_Tree,
                        Random_Forest),
              caption = "F1 Score by approach",
              position = "h")
```

Table 1: F1 Score by approach

Decision	KNN	Classification_Tree	Random_Forest
Buy	0.4885496	0.5508982	0.4517304
Neutral	NA	NA	0.0268456
Sell	0.4740973	0.6030303	0.5894040

The highest F1 score for *Buy* and *Sell* are achieved using classification tree, and are above our goal of 50%. The *Sell* decision has the best result, which could be influence of the highest prevalence. The *Neutral* decision was predicted only by random forest approach, and has a poor performance F1 score that is below 3%. The profit, or loss, of the test set was not presented since the true out-of-sample tests are only possible in live trading.

## Conclusion

The decision traded system, developed in this project, provide suggestions for decision that are most pertinent to a particular combination of close price and their own extensions. The goal of this project was partially

achieved and the *F1score* of *Buy* and *Sell* is above 50%, our goal, using Classification Tree . The add of other assets and economic variable could provide a even better model since we have more data to train, and can capture more structures and latent factors in the data.

## References

- Zuckerman, Gregory. 2019. *The Man Who Solved the Market*, Penguin Publishing Group. Kindle edition
- Arnott, Robert D., Campbell R. Harvey, and Harry Markowitz. 2018. "A Backtesting Protocol in the Era of Machine Learning." *SSRN Electronic Journal*. <https://doi.org/10.2139/ssrn.3275654>.
- Coqueret, Guillaume, and Tony Guida. 2020. *Machine Learning for Factor Investing*. Chapman; Hall/CRC. <https://doi.org/10.1201/9781003034858>.
- Irizarry, Rafael A. 2019. *Introduction to Data Science*. Chapman; Hall/CRC. <https://doi.org/10.1201/9780429341830>.