Coffee Future Price Pediction 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).[1] 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.[2] The goal of this project is to create a decision trade system which balanced accuracy, or F1 score, is above 50%, [3]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[4]. 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[5]. The future trader decisions could be **buy** the contract, **sell** or not get in a position, what is the same to be **neutral** in a asset, this decisions will be our **outcome** $\mathbf{y} = y_t$, which is categorical.

[1] http://www.mlfactor.com/index.html

[2] https://www.b3.com.br/en_us/products-and-services/trading/commodities/product-sheet-8AE490C96D41D3A2016D46017EC97262.htm

[3] "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%."

[4] https://corretora.clear.com.br/plataformas/profit-trader-clear/

[5] https://en.wikipedia.org/wiki/Decision_support_system

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.[6] 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 trough 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 Regression Trees and Classification and Regression with Random Forest

Data cleaning

For this project we going to use a subset of dataset provided by Nelogica and 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)
library(corrplot)
```

```
##Data Clean
```

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

```r
#create tempfile and download
dl <- tempfile()
download.file("https://github.com/mrabelosoares/Coffee-Future-Price-Prediction/blob/2044135ec7c5bcfb6f35
#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 XLXS 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.

```r
##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.

```r
#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
```

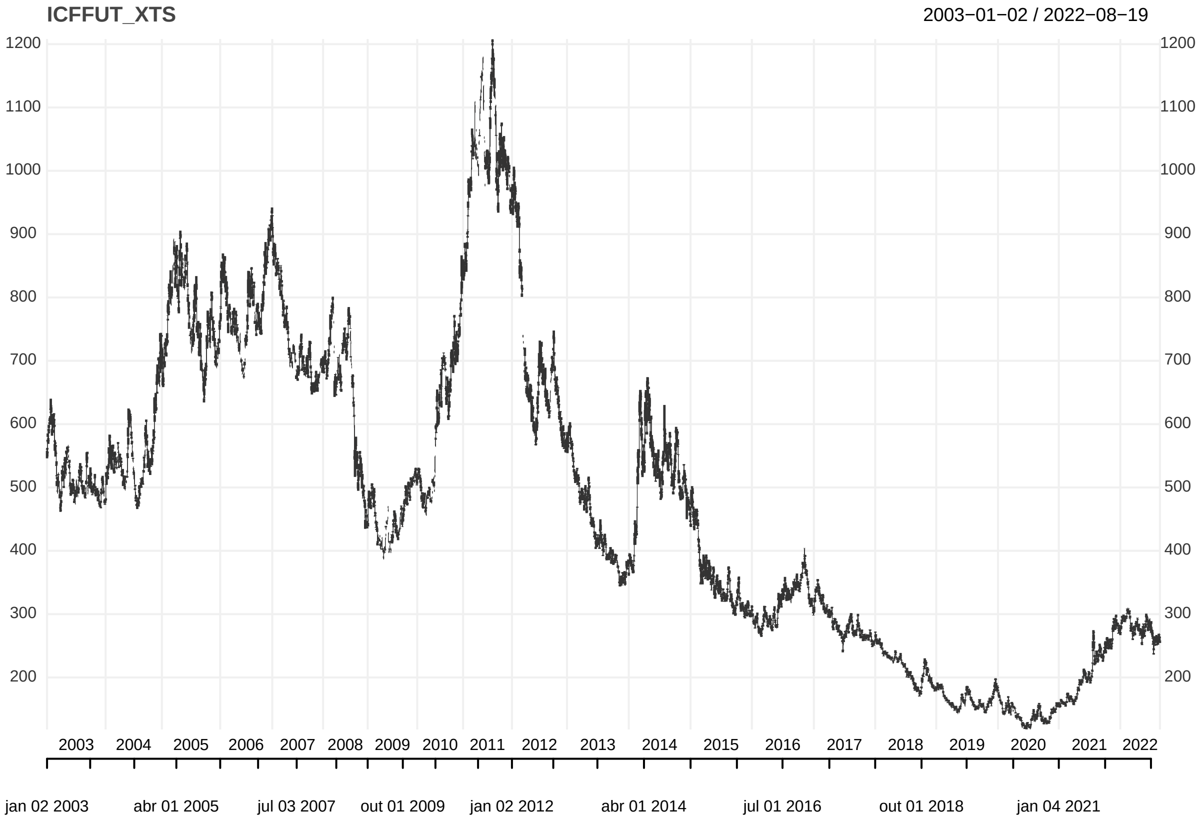
The extensions are technical indicators based on a daily Close Price of ICF and are select based on Chartered Financial Analyst (CFA®) program curriculum. This choice make sense economically since this knowledge are wild spread to the market and the decisions are made at the closing call every day.

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: 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 trough 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")
Notrade
```

```
##         Date ICFFUT.Open ICFFUT.High ICFFUT.Low ICFFUT.Close ICFFUT.Volume
## 1 2008-03-06      760.81      760.81     760.81       760.81             0
## 2 2011-01-25      885.65      886.97     884.20       886.97             0
## 3 2011-03-07     1040.40     1046.24    1040.40      1046.24             0
## 4 2011-03-08     1046.24     1052.12    1046.24      1052.12             0
##        SMA     SMA.1       dn      mavg        up      pctB ICFFUT.Close.1
## 1  782.018  752.7982 707.7564  757.3215  806.8866 0.5351911    0.006567172
## 2  874.010  856.6527 826.5829  858.0130  889.4431 0.9606576    0.039426829
## 3 1038.644 1010.0505 955.2408 1013.9420 1072.6432 0.7751053    0.007001822
## 4 1042.742 1013.6918 962.3474 1018.3930 1074.4386 0.8008889   -0.004125986
##        rsi     fastK     fastD     slowD     macd   signal Decision
## 1 54.83479 0.4545190 0.6465015 0.7787692 2.590852 2.723433     Sell
## 2 70.30439 1.0000000 1.0000000 0.8148056 2.732781 2.876085      Buy
## 3 72.84037 0.8727137 0.8264090 0.7831283 3.142953 3.514374      Buy
## 4 74.18861 0.9173004 0.8644236 0.8234052 3.086364 3.428772      Buy
##   OpenInterest
## 1            0
## 2            0
## 3            0
## 4            0
```

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
##    <dttm>                <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)
```

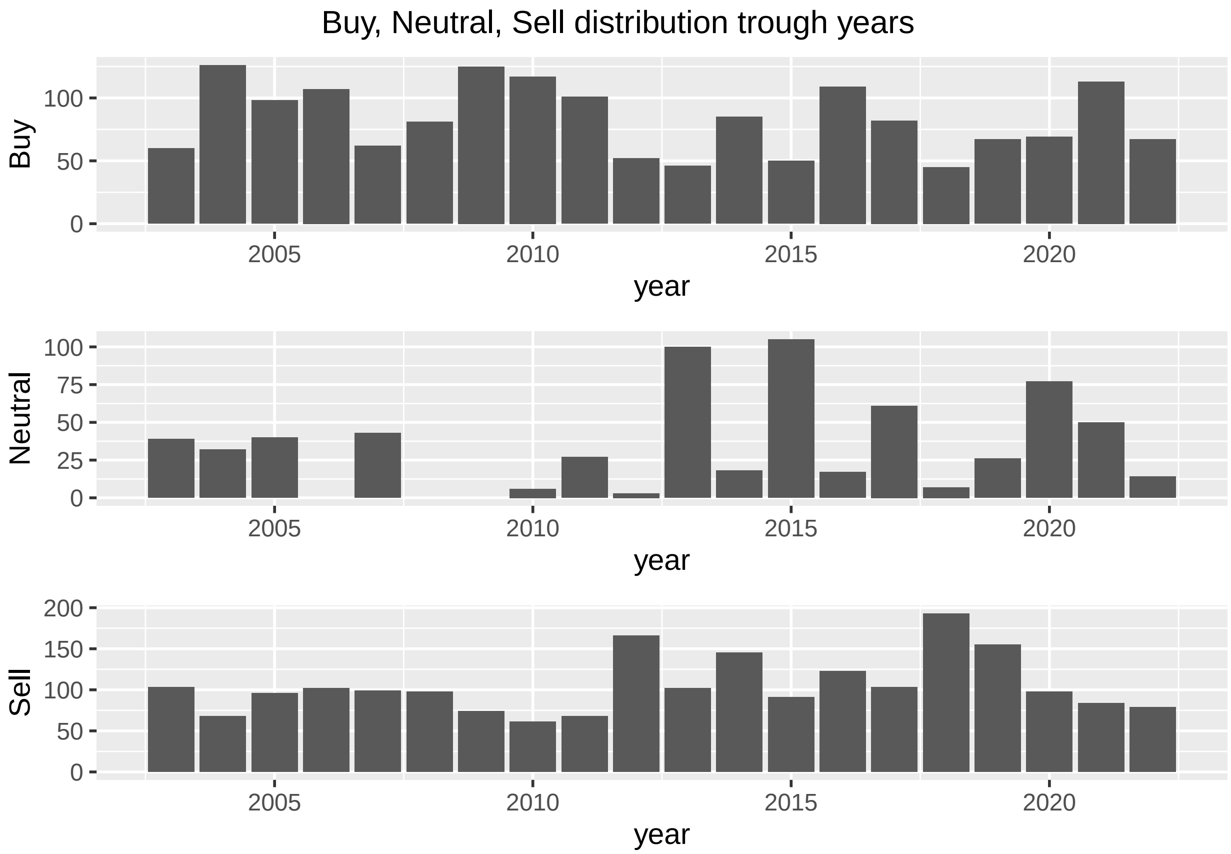
```
##       Date                         ICFFUT.Open      ICFFUT.High
##  Min.   :2003-02-18 00:00:00.00   Min.   : 119.4   Min.   : 121.0
##  1st Qu.:2008-02-21 12:00:00.00   1st Qu.: 271.6   1st Qu.: 275.0
##  Median :2013-08-28 00:00:00.00   Median : 470.6   Median : 475.9
##  Mean   :2013-03-28 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 ...
```

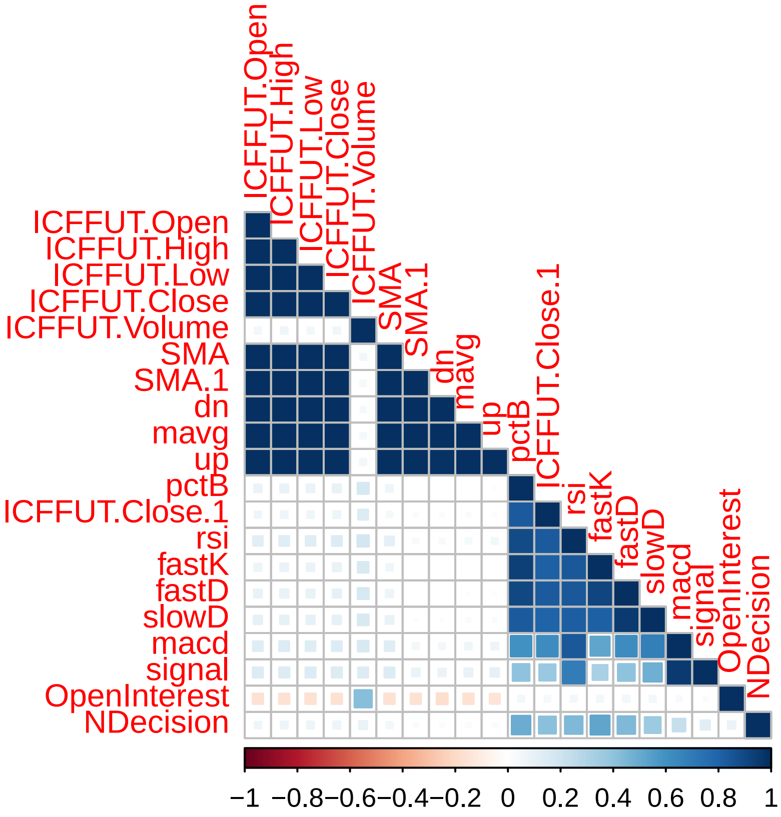
As presented above our matrix have $4,435 \times 21$ variables, mostly numerical column. The close price, the principal column, has 119.6 as minimal value and 1206.0 as maximal, and a median of 470.3. The decision column, our outcome, has a unbalanced prevalence: $Buy = 1662$, $Neutral = 665$, $Sell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

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



The graphs show different prevalence for different years, this a relevant information for building a general model, since the prevalence is one off principal components of machine learning measure of performance. The correlation between variables and outcome could bring more information about the behavior of data, which reflects the decisions of traders.

```r
#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')
```



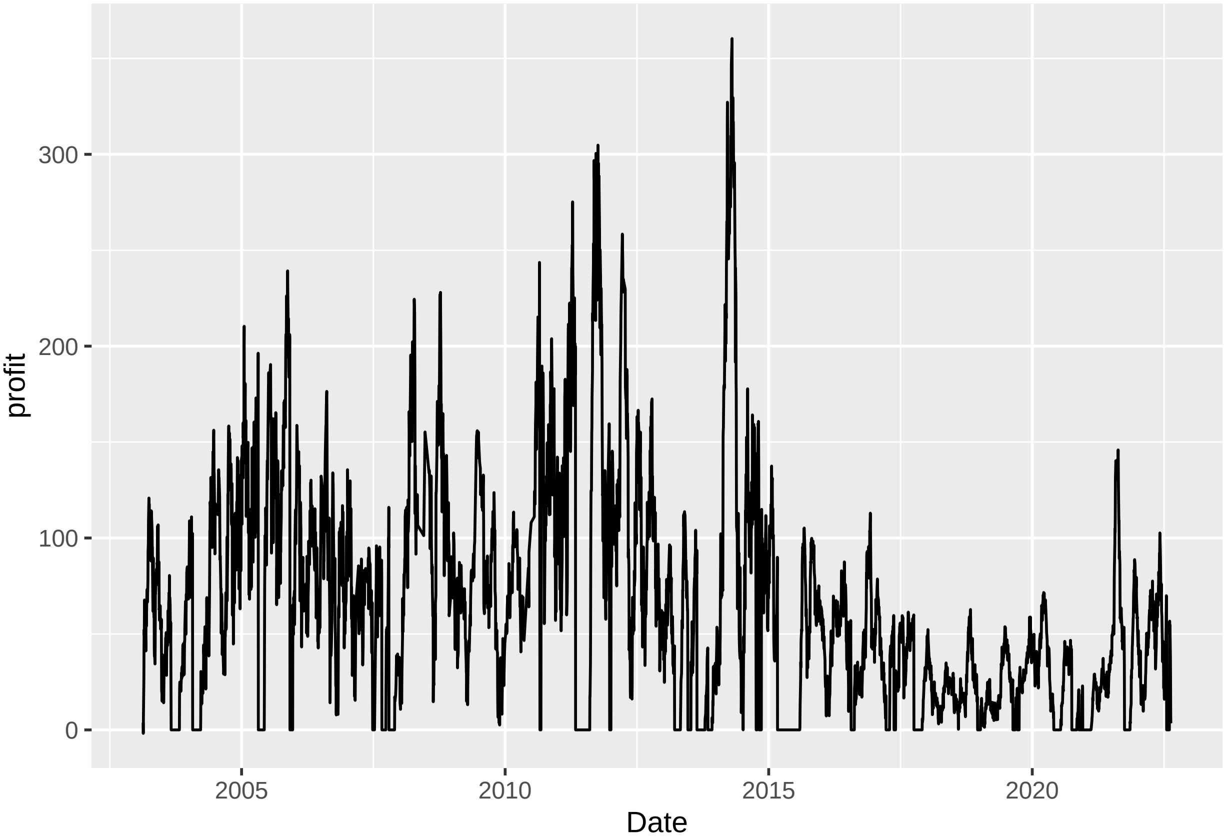
The correlation between the variables and between variable and outcome are presented below. The prices, mean movies and Bollinger Bands are high correlated

The result of decisions made trough years, or the sum of profits, could be calculated using the code below. The protocol propoused to

```
#empirical optimal results - sum last 22 days
profitICFFUT <- CDFICFFUT |>
  arrange(Date) |>
  mutate(return = ICFFUT.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
```

```
#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 recommendation system is better as its error has decreased, for this project the error is the typical error we make when predicting a movie rating $\left(\hat{y}_{u,i} - y_{u,i}\right)$. The loss function used to evaluate the models is based on the residual mean squared error $(RMSE)$ on a test set. The definition of $RMSE$ includes N , the number of user/movie combination, and the sum occurring over all these combination. In this case the Loss function is:

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

The model which $RMSE = 0$ is a perfect model prediction, or without errors. For this project the reported $RMSE < 0.86490$ is the goal. The $RMSE > 1$ means our error is larger than one star, which means a bad model.

Machine Learning

“If we have enough data, I know we can make predictions,” Simons told a colleague.

Zuckerman, Gregory. The Man Who Solved the Market (p. 2). Penguin Publishing Group. Edição do Kindle.

The machine learning decisions are based on algorithms build with data, so for this project the dataset XXXXX are going to be used to train and test the model. The train_set and test_set are build trough function `createDataPartition` as presented:

```
#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)
head(testing_sample)
```

```
##           Date ICFFUT.Open ICFFUT.High ICFFUT.Low ICFFUT.Close ICFFUT.Volume
## 1 2020-01-02      183.16      184.14     179.08       180.69      31405541
## 2 2020-01-03      178.53      180.81     177.85       180.56       5166389
## 3 2020-01-06      173.96      176.86     172.91       174.76      30451609
## 4 2020-01-07      172.91      175.68     170.81       175.38      19172994
## 5 2020-01-08      174.51      174.51     169.20       170.99      25606202
## 6 2020-01-09      170.44      172.54     167.66       168.15       9397438
##       SMA    SMA.1       dn     mavg       up         pctB ICFFUT.Close.1
## 1 181.998 182.3991 170.6534 183.6730 196.6926  0.385442050    -0.01155510
## 2 182.602 182.9436 171.8471 184.0435 196.2399  0.357191696    -0.08393750
## 3 181.564 183.1264 171.4344 183.9295 196.4246  0.133076183    -0.07748649
## 4 179.366 183.2277 171.7057 183.9975 196.2893  0.149461253    -0.07325640
## 5 176.476 182.9527 170.2190 183.6020 196.9850  0.028805115    -0.05584168
## 6 173.968 182.6859 168.2867 183.1080 197.9293 -0.004610492    -0.08348151
##        rsi      fastK       fastD      slowD        macd   signal Decision
## 1 51.48632 0.16728625 0.351920694 0.31407104  2.12394370 2.932564     Sell
## 2 51.36586 0.16038237 0.249070632 0.31327079  1.84973838 2.715999     Sell
## 3 46.17506 0.00000000 0.109222871 0.23673807  1.35944560 2.444688     Sell
## 4 46.79403 0.02869042 0.063024263 0.14043926  0.98614391 2.152980     Sell
## 5 43.02153 0.00000000 0.009563474 0.06060354  0.48653615 1.819691     Sell
## 6 40.73367 0.00000000 0.009563474 0.02738374 -0.04022586 1.447708     Sell
##   OpenInterest
## 1            0
## 2         2617
## 3            0
## 4         2678
## 5         2705
## 6         2678
```

k-Nearest Neighbour Classification approach

The simplest model to recommend a movie to any user, is the model that predict the same rate μ for all movies regardless of user. In the naive approach the variation of the differences is random, and the independent error $\epsilon_{u,i}$ centered at 0. The model looks like:

$$Y_{u,i} = \mu + \varepsilon_{u,i}$$

The average of all ratings is the estimate that minimize the $RMSE$, and if we fill the blank cells in the matrix with the μ we obtain the $\varepsilon_{u,i}$.

```
#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
```

```
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
```

```
#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
```

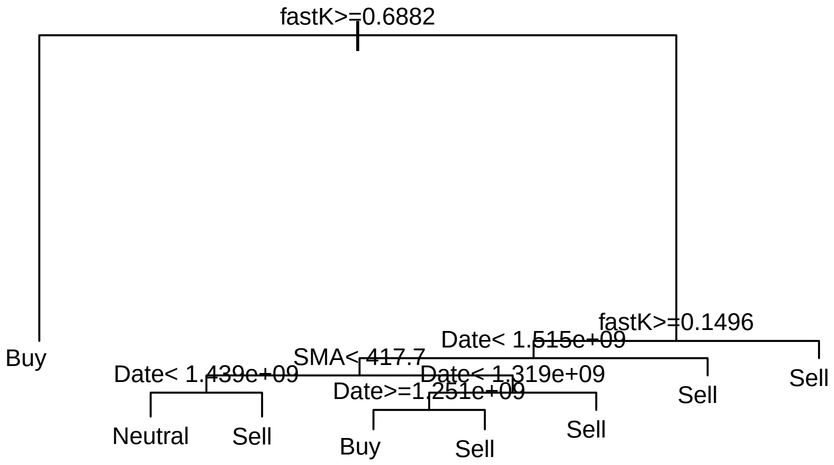
```
cm1[["byClass"]][ , "F1"]
```

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

Recursive Partitioning and Regression Trees approach

The values of b_i and b_u that minimize the full model and the code to calculate the $RMSE$ is presented below.Finally we achieve our goal, the $RMSE$ is lower than 0.86490. But can we do better?

```
#defining the predictors - Model 2
fit <- rpart(Decision ~ ., data = training_sample)
plot(fit, margin = 0.1)
text(fit, cex = 0.75)
```



```
#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
```

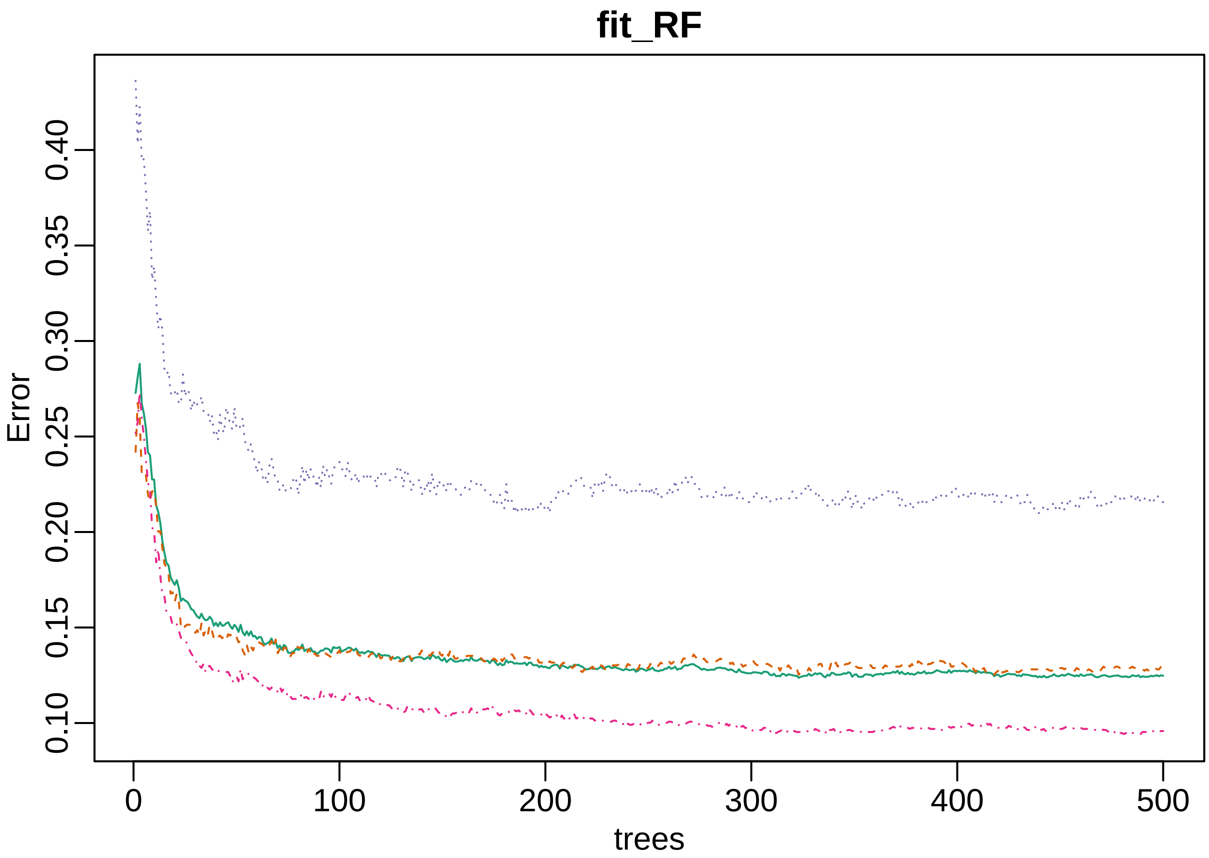
```
cm2[["byClass"]][ , "F1"]
```

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

Classification and Regression with Random Forest approach

A matrix Factorization is the last approach to solve recommendation system problem presented in thisThe model improved substantially and we achieve the best result so far, we have our choice to train and test the dataset.

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

rafalib::mypar()
plot(fit_RF)
```



```
varImp(fit_RF)
```

```
##                  Overall
## Date           179.00982
## ICFFUT.Open     76.27939
## ICFFUT.High     77.57936
## ICFFUT.Low      80.66319
## ICFFUT.Close    78.75611
## ICFFUT.Volume   82.75799
## SMA             91.80401
## SMA.1           92.34527
## dn             106.75145
## mavg            90.90132
## up              89.13221
## pctB           186.25270
## ICFFUT.Close.1 117.57677
## rsi            153.37391
## fastK          224.87914
## fastD          119.40587
## slowD          104.42693
## macd           108.80197
## signal         117.05089
## OpenInterest   103.03690
```

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

```
## Accuracy
##  0.46851
```

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

```
##     Class: Buy Class: Neutral    Class: Sell
##      0.4115756      0.2857143      0.5255255
```

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

```
##     Class: Buy Class: Neutral    Class: Sell
##      0.5140562      0.0141844      0.6704981
```

```
cm3[["byClass"]][ , "F1"]
```

```
##     Class: Buy Class: Neutral    Class: Sell
##     0.45714286     0.02702703     0.58922559
```

Results

This section presents the modeling results and discusses the model performance of the `movielens` data frame. The code below provided by HarvardX and create the `edx` and `validation` datasets that will be used to train and test our final algorithm.a $RMSE$ of 0.78.

Conclusion

The recommendation system, developed in this project, provide suggestions for items that are most pertinent to a particular user. The goal of this project was achieved and the $RMSE$ of matrix factorization of 0.78 is lower than our target 0.86490. A larger dataset could provide a even better model since we have more data to train, and is one of the limitations of this project. The addition of more variables could be a good outlook for future works since can capture more structures and latent factors in the data.

Evaluation metrics

Avoiding False Positives: A Protocol

Model that makes sense economically: Renaissance Technologies Trading Strategies

1- Mean Reversion Strategy: Buy if the close price move a certain level below their recent trend line, Sell if above certain level above and be Neutral if the price is too close of trend line.

2- Stochastic Strategy: Non linear machine learning Kernel

3- The Kelly Criterion: scientific gambling method Assumption: future returns depend on decisions toked on close price. The relationship between the characteristics of market and performance is largely unknown and probably time-varying. This is why ML can be useful: to detect some hidden patterns beyond the documented asset pricing anomalies. Moreover, dynamic training allows to adapt to changing market conditions.

1. Research Motivation: ex ante/ex post economic foundation
a) Does the model have a solid economic foundation?

Coffee It is one of the most important commodities traded globally and one of the most popular beverages in the world. World production is spread over 55 developing countries and is conducted by about 26 million producers, cultivated in tropical and subtropical areas.

The development of local organizations or instruments to help managing price risk, such as organized futures, in developing countries, like Brazil, is a important step to minimize the risk and challenges of small producers, since the larger and more liquids futures markets are located in the United States and Europe.

The Futures price volatility is large once coffee market has low price elasticity of demand and supply in the short run. In addition, previous studies have pointed to the existence of asymmetric price transmission within the coffee chain. Because of the concentration of roasting and retail sectors, retail prices respond faster to price increases than to price decreases, while farm prices follow more closely supply and demand conditions.

b) Did the economic foundation or hypothesis exist before the research was conducted?
Price changes are commonly caused by the arrival of new information from ICO crop reports, as presented in work The Reaction of Coffee Futures Price Volatility to Crop Reports (2017)

2. Multiple Testing and Statistical Methods
a) Did the researcher keep track of all models and variables that were tried (both successful and unsuccessful) and are the researchers aware of the multiple-testing issue?

b) Is there a full accounting of all possible interaction variables if interaction variables are used?

c) Did the researchers investigate all variables set out in the research agenda or did they cut the research as soon as they found a good model?

3. Data and Sample Choice
a) Do the data chosen for examination make sense? And, if other data are available, does it make sense to exclude these data?
b) Did the researchers take steps to ensure the integrity of the data?
c) Do the data transformations, such as scaling, make sense? Were they selected in advance? And are the results robust to minor changes in these transformations?
d) If outliers are excluded, are the exclusion rules reasonable?
e) If the data are winsorized, was there a good reason to do it? Was the winsorization rule chosen before the research was started? Was only one winsorization rule tried (as opposed to many)?

4. Cross-Validation
a) Are the researchers aware that true out-of-sample tests are only possible in live trading?
b) Are steps in place to eliminate the risk of out-of-sample "iterations" (i.e., an in-sample model that is later modified to fit out-of-sample data)?
c) Is the out-of-sample analysis representative of live trading? For example, are trading costs and data revisions taken into account?

5. Model Dynamics
a) Is the model resilient to structural change and have the researchers taken steps to minimize the overfitting of the model dynamics?
b) Does the analysis take into account the risk/likelihood of overcrowding in live trading?
c) Do researchers take steps to minimize the tweaking of a live model?

6. Complexity
a) Does the model avoid the curse of dimensionality?
b) Have the researchers taken steps to produce the simplest practicable model specification?
c) Is an attempt made to interpret the predictions of the machine learning model rather than using it as a black box?

7. Research Culture
a) Does the research culture reward quality of the science rather than finding the winning strategy?
b) Do the researchers and management understand that most tests will fail?
c) Are expectations clear (that researchers should seek the truth not just something that works) when research is delegated?

Category #1: Research Motivation

Category #2: Multiple Testing and Statistical Methods Keep track of: what is tried, combinations of variables, Beware the parallel universe problem.

Category #3: Sample Choice and Data Define the test sample ex ante. Ensure data quality Document choices in data transformations Do not arbitrarily exclude outliers. Select Winsorization level before constructing the model.

Category #4: Cross-Validation Acknowledge out of sample is not really out of sample Understand iterated out of sample is not out of sample. Do not ignore trading costs and fees.

Category #5: Model Dynamics Be aware of structural changes. Acknowledge the Heisenberg Uncertainty Principle and overcrowding. Refrain from tweaking the model.

Category #6: Model Complexity Beware the curse of dimensionality. Pursue simplicity and regularization. Seek interpretable machine learning.

Category #7: Research Culture Establish a research culture that rewards quality. Be careful with delegated research

References

Coqueret, Guillaume, and Tony Guida. 2020. *Machine learning for factor investing: R version.* Chapman; Hall/CRC.

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.