Data science report V2

Antonio Malatesta Karan Soneja Michele D'Ambrosio

19/12/2019

Abstract

A short summary of the work.

Contents

1 Introduction

What motivates us to do this work. Touch on important finding in literature.

2 Data

The data used in this report consists of the stock price of some of the 10 biggest French companies.

Table 1: :
Top ten firms in CAC40 by market cap, September 2019

Company	MNEMO	Sector	Weight%
TOTAL	FP	Oil and Gas	9,54
LVMH	MC	Personal and Household Goods	7,97
SANOFI	SAN	Health Care	7,54
AIRBUS	AIR	Industrial Goods and Services	5,47
L'OREAL	OR	Personal and Household Goods	5,10
AIR LIQUIDE	AI	Chemicals	4,40
DANONE	BN	Food and Beverage	4,14
VINCI	DG	Construction and Materials	3,97
BNP PARIBAS ACT.A	BNP	Banks	3,95
SAFRAN	SAF	Industrial Goods and Services	3,72

With the following code, we downloaded the daily open, high, low, close, Volume and adjusted price of the desired stock from 2016-12-31 to 2018-12-31.

```
getSymbols("COMPANY CODE", src="yahoo", from="2016-12-31", to="2018-12-31")
```

In some models, we also used economic and financial variables to test whether they could increase the predictability power of the analysis. See the following table for a detailed list of the variables we studied.

Table 2: Additional economical variables

Economic Factors	Source
Daily interest rate	sdw.ecb.europa.eu
Daily Exchange rate (EUR/USD)	exchangerates.org
Political Stability (yearly)	databank.worldbank.org
CPI Infaltion (monthly)	www.inflation.eu

some text in between to have better formatting

Table 3: Additional economical variables

Yearly Financial Statements Variables	Source
Dividend	
Revenue	
Net income	dividenmax.com
Basic earnings per share	
Diluted earnings per share	airbus.com
Total assets	
Intangible assets	airliquide.com
PPE	
Cash	danone.com
Total equity	
Non current liability	vinci.com
Cash flows from operating activities	
Net cash flows from investing activities	invest.bnpparibas.com
Net cash flows from financing activities	
Net cash and cash equivalents at end of period	

We gathered all the data mentioned above in one single master database that can be found in the file “master file.xlsx”.

3 Technical analysis: implementation of a naive strategy in R

3.1 try GP

Make an attempt to get the gp working naively using trading signal generation with functions from above

3.2 Some notes

3.2.1 Some useful links

1. A list of easy TA indicators. We have:
   - MA
   - Bollinger bands
   - RSI
   - MACD The code is shown but not shared. The package is {TTR}
2. A series of posts on trading
   - Attempt of backtesting in quantmod
   - Read quickly part1; no need for the code beacuse it's in the next post. Here however a good tip on charting several price charts together. First Search « Create a plot showing all series as lines; ». Second, use return normalization (search for « ## Loading required package: magrittr »). Not much more in this part.
   - Then read part2

3.2.2 what indicators (from Neftci [1991])

1. Trend crossing methods
   - New item
2. Moving average methods
   - Respect Markov times theorem
3. As a sidenote, don't forget about the importance of having an exit strategy. (reminder here)
4. Add a stop-loss rule
5. On signals and rules, here
6. Don't forget to add a benchmarking stock (as the SPY for the US market) to know what would have been like to have a long position on the market portfolio

4 Neural networks

Neural networks (NN) is a "black box" technique that finds non-linear connections and recognize patterns between one or multiple inputs to generate an output[1] . We will test a 1-input NN[2] and a multiple-input NN[3] in the attempt to predict the next-day stock price of our 5 stocks, over a period of 30 trading days, from the 16/11/2018 to the 31/12/2018. Based on the predictions, we will develop an investing strategy where everyday, for the above-mentioned 30 days, we will invest all our capital (starting at 10,000 euro) into the stock that will have the biggest predicted increase in price the next day, according to our model. Consequently, the next day we will sell all our stocks bought the previous day, and invest again all our capital with the same reasoning. In this analysis, all commission and brokerage fees are ignored.

4.1 Benchmark

To evaluate the performance of the NN as an investment decision-maker, instead of using a ROC Curve, we decided to compare it with a diversified, long-term investment strategy, where the initial capital of 10,000 euro is invested equally in the 5 stocks (2,000 euro per stock) at the beginning of the period (16/11/2018) and all stocks are then sold at the end of the period (31/12/2018).

Table 4: Benchmark strategy outcome

Dates	AIR.PA	FP.PA	MC.PA	OR.PA	SAN.PA		Capital
16/11/2018	95.86	92.88	45.19	65.09	77.09	stock price(euro)	Total budget(euro)
	2000	2000	2000	2000	2000	amount invested(euro)	10000
	20.86	21.53	44.25	30.72	25.94	N. of shares bought	
31/12/2018	98.59	83.95	39.47	61.5	72.01	stock price(euro)	Total budget(euro)
	2056.89	1807.72	1746.87	1889.7	1868.22	value in stocks(euro)	9369.43

As can be seen, the benchmark strategy has a return of -6%. Clearly, the higher the return of the NN-based investments, the more successful the NN algorithms will be. Moreover, in case of a return lower than -6% (lower than the "naive" strategy), the NN model will be considered as not useful in predicting stock prices.

4.2 1-input NN

The first NN model is a single hidden layer neural network. In this model there is one layer of input nodes that send weighted inputs to a subsequent layer of receiving nodes. The nnetar function in the forecast package fits a single hidden layer neural network model to a timeseries. The function model approach is to use lagged values of the time series as input data, reaching to a non-linear autoregressive model. The model takes as input the stock prices starting from the 31/12/2016 to 31/12/2018. The goal is to predict the stock prices between 16/11/2018 and 31/12/2018.

[1](https://pathmind.com/wiki/neural-network)

[2](http://rpubs.com/kapage/523169?fbclid=IwAR3RW_tVA_7SgCah7M0nmZL7anIL4gS_ZZ3dP_i_w8AOyQXoTMwXC_wQsoE)

[3](https://www.analyticsvidhya.com/blog/2017/09/creating-visualizing-neural-network-in-r/?fbclid=IwAR0z9tD0-WFxG3zVs8YmFaLF7RGnszbizQrhyGdyZG1-GL-D5coqBOnTiZE)

```
library(prophet)
library(quantmod)
library(forecast)
library(xlsx)
library(tseries)
library(timeSeries)
library(dplyr)
library(fGarch)
#Download the prices for the 5 stocks for the desired time frame
getSymbols("AI.PA", src="yahoo", from="2016-12-31", to="2018-12-31")
getSymbols("AIR.PA", src="yahoo", from="2016-12-31", to="2018-12-31")
getSymbols("BN.PA", src="yahoo", from="2016-12-31", to="2018-12-31")
getSymbols("BNP.PA", src="yahoo", from="2016-12-31", to="2018-12-31")
getSymbols("DG.PA", src="yahoo", from="2016-12-31", to="2018-12-31")

alpha <- 1.5^(-10)
```

We generate the predicted price of the 30 days for the 5 stocks with a `for loop`, where the prices are predicted one at the time.

```
for (n in 1:30){
  close_price <- as.numeric(AIR.PA[1:479+n,'AIR.PA.Close'])
#Hidden layers creation
  alpha <- 1.5^(-10)
  hn <- length(close_price)/(alpha*(length(close_price)+1))
#Fitting nnetar
  lambda <- BoxCox.lambda(close_price)
  dnn_pred <- nnetar(close_price, size= hn, lambda = lambda)
  dnn_forecast <- forecast(dnn_pred, h= 1, PI = TRUE)
  AIR.PA.P[n] = dnn_forecast[["mean"]]
}

for (n in 1:30){
   close_price <- as.numeric(BN.PA[1:479+n,'BN.PA.Close'])
   hn <- length(close_price)/(alpha*(length(close_price)+1))
   lambda <- BoxCox.lambda(close_price)
   dnn_pred <- nnetar(close_price, size= hn, lambda = lambda)
   dnn_forecast <- forecast(dnn_pred, h= 1, PI = TRUE)
   BN.PA.P[n] = dnn_forecast[["mean"]]
 }

 #We run the same for loop above also for AI.PA, BNP.PA and DG.PA
```

We create a table with the predicted daily increase in price for every stock during the 30 days. After that, we will identify the stock with the highest predicted increase in price for every day. This will be the stock where to invest all the capital for the day.

```
AIR.LOL <- AIR.PA.P - as.numeric(AIR.PA[480:509,'AIR.PA.Close'])
BN.LOL <- BN.PA.P - as.numeric(BN.PA[480:509,'BN.PA.Close'])
BNP.LOL <- BNP.PA.P - as.numeric(BNP.PA[480:509,'BNP.PA.Close'])
DG.LOL <- DG.PA.P - as.numeric(DG.PA[480:509,'DG.PA.Close'])

df = data.frame(AI.LOL,AIR.LOL,BN.LOL,BNP.LOL,DG.LOL)

for (n in 1:30){
   if (max(df[n,]) == df[n,"AI.LOL"]){
   AI.DIF[n] = "AI.PA"
   } else if (max(df[n,]) == df[n,"AIR.LOL"]){
   AI.DIF[n] = "AIR.PA"
   } else if (max(df[n,]) == df[n,"BN.LOL"]){
   AI.DIF[n] = "BN.PA"
   } else if (max(df[n,]) == df[n,"BNP.LOL"]){
   AI.DIF[n] = "BNP.PA"
   } else if (max(df[n,]) == df[n,"DG.LOL"]){
   AI.DIF[n] = "DG.PA"
   }
}
```

Table 5: 1-input NN strategy outcome

Day	Invest in	date	AI	AIR	BNP	BN	DG	#shares bought	€ Value
1	BNP	16/11/2018	95.86	92.88	45.19	65.09	77.09	221.26	10000
2	BNP	19/11/2018	95.40	92.57	45.28	64.70	76.80	221.26	10019.91
3	BNP	20/11/2018	93.5	91.07	44.30	64.98	76.37	221.26	9803.07
4	AI	21/11/2018	94.22	93.16	44.79	65.41	77.01	105.17	9910.38
5	AIR	22/11/2018	93.31	92.33	44.28	65.27	76.62	106.30	9814.77
6	AI	23/11/2018	93.86	93.41	44.36	65.87	76.86	105.78	9929.57
7	AI	26/11/2018	95.86	93.58	45.34	65.82	78.54	105.78	10141.15
8	AIR	27/11/2018	94.86	93.69	45.08	66.41	78.16	107.10	10035.36
9	BNP	28/11/2018	94.36	93.5	44.75	65.41	77.95	223.75	10013.94
10	AI	29/11/2018	94.68	94.86	44.77	65.54	77.58	105.81	10018.42
11	AI	30/11/2018	97.04	94.62	44.37	66.05	77.09	105.81	10268.51
12	AI	03/12/2018	97.27	95.69	44.84	65.41	75.40	105.81	10292.56
13	AI	04/12/2018	97.45	94.26	43.86	65.52	76.66	105.81	10311.81
14	BNP	05/12/2018	96.45	92.30	43.41	64.58	74.66	235.05	10206.00
15	DG	06/12/2018	94.63	88.63	41.78	63.43	70.98	138.37	9821.69
16	AI	07/12/2018	95.86	89.07	41.59	63.79	72.68	104.90	10056.92
17	DG	10/12/2018	95.77	87.48	40.40	62.97	71.66	140.20	10047.38
18	DG	11/12/2018	96.45	88.59	40.68	62.81	73.69	140.20	10333.41
19	AI	12/12/2018	97.54	91.48	41.90	64.12	74.77	107.48	10484.84
20	BN	13/12/2018	97.63	89.97	42.18	64.32	74.5	163.16	10494.62
21	BNP	14/12/2018	97.45	88.66	41.59	64.08	73.51	251.36	10455.46
22	DG	17/12/2018	96.68	87.65	40.58	63.23	72.36	140.96	10200.32

Day	Invest in	date	AI	AIR	BNP	BN	DG	#shares bought	€ Value
23	DG	18/12/2018	95.68	88.65	40.55	62.68	71.33	140.96	10056.54
24	AI	19/12/2018	97.86	87.19	40.88	62.82	71.59	103.13	10093.19
25	DG	20/12/2018	97.09	83.33	39.40	62.25	71.41	140.20	10013.50
26	AI	21/12/2018	97.81	83.09	39.5	62.23	71.41	102.36	10013.50
27	AIR	24/12/2018	97.09	81.62	38.79	61.61	70.5	121.75	9939.04
28	AIR	27/12/2018	95.31	82.09	38.54	60.27	70.63	121.75	9996.27
29	AI	28/12/2018	96.59	83.76	39.37	60.66	71.95	105.58	10198.39
30	AI	31/12/2018	98.59	83.95	39.47	61.50	72.01		10409.55

4.2.1 Evaluation of 1-input NN

As can be seen, the investment strategy based on the 1-input NN has a return of +4%, which is an extremely good result considering the benchmark strategy return of -6%. This implies that, even though all the 5 stocks had negative performances during the time frame, the NN successfully identified the right stock at the right time, leading to a positive return.

4.3 Multiple-input NN

The second NN model uses the neuralnet library for the analysis. It has 3 neurons in its hidden layer and the weights are calculated using a back propagation algorithm. The model takes several inputs that for simpliticy have been coded according to the following table:

Table 6: Variables coding

Variable	Identifier	Variable	Identifier	Variable	Identifier
Close price	A1	CPI monthly inflation	A10	PPE	A18
Open price	A2	Dividend	A11	Cash	A19
High price	A3	Revenue	A12	Total equity	A20
Low price	A4	Net Income	A13	Non current liability	A21
Volume	A5	Basic earnings per share	A14	Cash flows (used in)/from operating activities	A22
Adjusted price	A6	Diluted earnings per share	A15	Net cash flows (used in)/from investing activities	A23
Interest rate	A7	Total assets	A16	Net cash flows (used in)/from financing activities	A24
Exchange rate (1 USD)	A8	Intangible assets	A17	Net cash and cash equivalents at end of period	A25
Political stability	A9				

The goal is always to predict stock prices of the 30 days, hoping that the inclusion of additional variables like high price, low price, inflation and financial statements data will provide more accurate predictions.

```
#Download "AI1.csv" "AIR1.csv" "BN1.csv" "BNP1.csv"
#Download "DG1.csv" "PREDICTION SIMPLE.xlsx"
library(readr)
AI1 <- read_csv2("D:/2017-2018/data_analysis/technical_analysis/report/AI1.csv")
data <- AI1[, c(2:26)]
View(data)
#we take all the data frame as training data to predict the next-day stock price
datatrain = data[ 1:1531, ]
datatest = data[ 1532, ]
#we normalize the values to avoid results based only on the disporportionate
#size of certain inputs
max = apply(data , 2 , max)
min = apply(data, 2 , min)
scaled = as.data.frame(scale(data, center = min, scale = max - min))
# install library
install.packages("neuralnet ")
#load library
library(neuralnet)
# creating training and test set
# the test set will be the closing price of a single day
# the training set will be all the data of the days before the day of the test set
trainNN = scaled[1:1531 , ]
testNN = scaled[1532 , ]
```

We want to predict every next day price for the 5 stocks. But first, we want to check weather the the model gives more accurate results with or without the financial statement data as inputs.

```
# fit neural network
set.seed(2)
# Develop a NN model with tomorrow closing price as output and
# all the other variables (taken today) as input
NN = neuralnet( A1 ~ A2 + A3 + A4 + A5 + A6 + A7 + A8 + A9 + A10
+ A11 + A12 + A13 + A14 + A15 + A16 + A17 + A18 + A19 + A20 + A21
+ A22 + A23 + A24 + A25, trainNN, hidden = 3 , linear.output = T )
# plot neural network
plot(NN)
```

In the image below we can see a graphical representation of the neuronal net: It has 3 neurons in its hidden layer. The black lines show the connections with weights. The weights are calculated using a back propagation algorithm. The blue line displays the bias term.

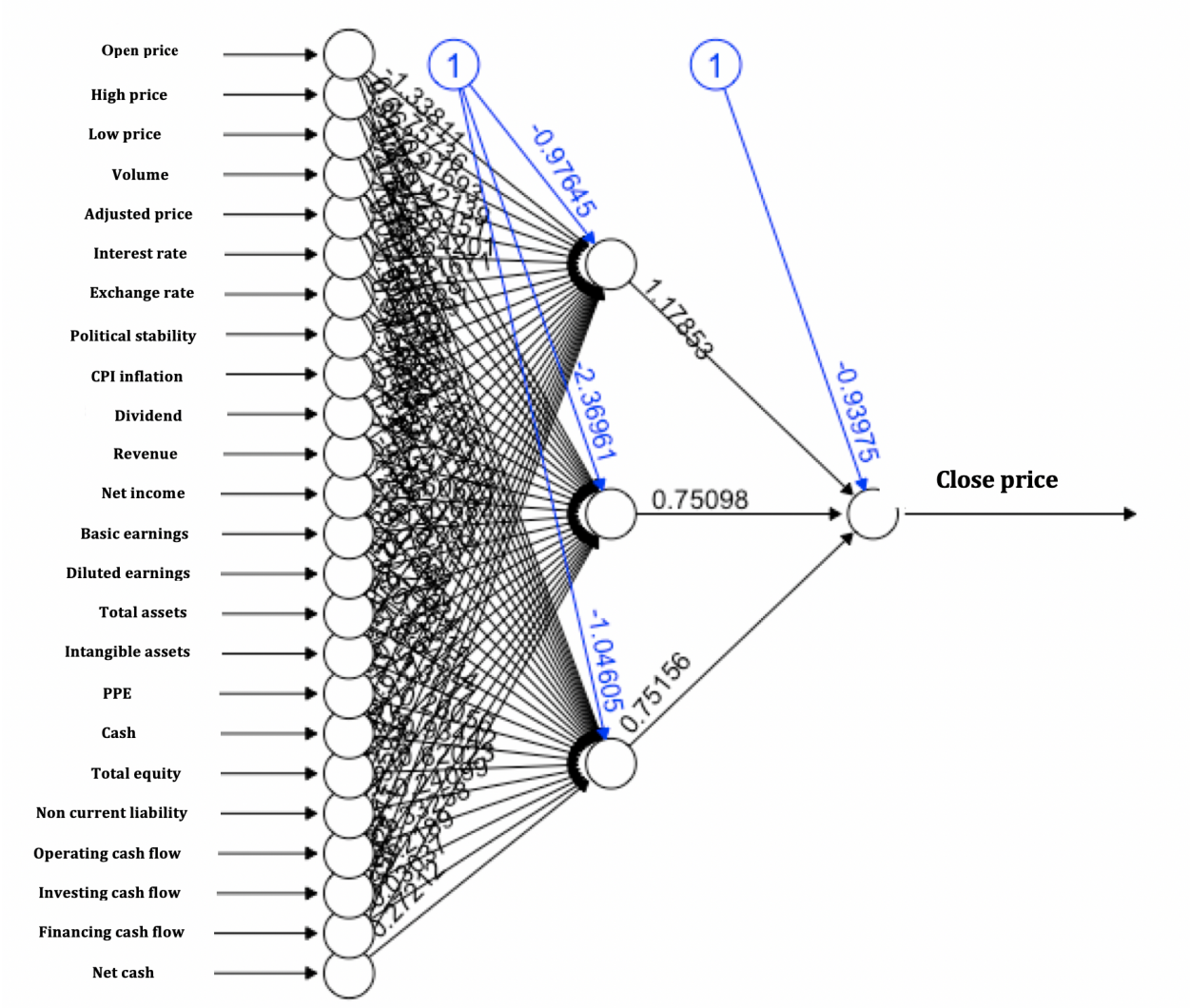

Figure 1: Caption for the picture.

```
predict_testNN = compute(NN, testNN[,c(2:25)])
# We descale the output to obtain the results in the real scale
predict_testNN = (predict_testNN$net.result * (max(data$A1) - min(data$A1)))
+ min(data$A1)
# We create an array where to store the results per day for each stck
AI.P <- c()
View(data)
View(data)
# We run the model for every single day per stock, to predict the prices
# one by one
View(predict_testNN)
for (n in 1:30){
  datatrain = data[ 1:1501+n, ]
  datatest = data[ 1502+n, ]
```

```
  max = apply(data , 2 , max)
  min = apply(data, 2 , min)
  scaled = as.data.frame(scale(data, center = min, scale = max - min))
  trainNN = scaled[1:1501+n , ]
  testNN = scaled[1502+n , ]
# model with financial statements data
  NN = neuralnet(A1 ~ A2 + A3 + A4 + A5 + A6 + A7 + A8 + A9 + A10
  + A11 + A12 + A13 + A14 + A15 + A16 + A17 + A18 + A19 + A20 + A21
  + A22 + A23 + A24 + A25, trainNN, hidden = 3 , linear.output = T )
  predict_testNN = compute(NN, testNN[,c(2:25)])
  predict_testNN = (predict_testNN$net.result * (max(data$A1) - min(data$A1)))
  + min(data$A1)
  AI.P[n] <- predict_testNN
}
View(AI.P)
AI.P2 <- c()
for (n in 1:30){
  datatrain = data[ 1:1501+n, ]
  datatest = data[ 1502+n, ]
  max = apply(data , 2 , max)
  min = apply(data, 2 , min)
  scaled = as.data.frame(scale(data, center = min, scale = max - min))
  trainNN = scaled[1:1501+n , ]
  testNN = scaled[1502+n , ]
# model without financial statements data
  NN = neuralnet(A1 ~ A2 + A3 + A4 + A5 + A6 + A7 + A8 + A9 + A10,
  trainNN, hidden = 3 , linear.output = T )
  predict_testNN = compute(NN, testNN[,c(2:10)])
  predict_testNN = (predict_testNN$net.result * (max(data$A1) - min(data$A1)))
  + min(data$A1)
  AI.P2[n] <- predict_testNN
}
# Calculate Root Mean Square Error (RMSE)
Reals <- data[1503:1532,1]
View(Reals)
View(data)
RMSE <- sqrt(sum((Reals-AI.P)^2)) # RMSE = 6.072
RMSE2 <- sqrt(sum((Reals-AI.P2)^2)) # RMSE2 = 6.064
View(AI.P)
```

As can be seen, the model without the financial statement variables (RMSE = 6.072) is more accurate than the model with them (RMSE2 = 6.064) according to the RMSE value. Hence, we will continue without taking into account the financial statement variables.

```
#We predict the prices of the other 4 stocks one by one,
#without the financial statement variables as input
library(readr)
AIR1 <- read_csv2("D:/2017-2018/data_analysis/technical_analysis/data/AIR1.csv")
```

```
data <- AIR1
AIR.P2 <- c()
View(data)
for (n in 1:30){
  datatrain = data[ 1:1501+n, ]
  datatest = data[ 1502+n, ]
  max = apply(data , 2 , max)
  min = apply(data, 2 , min)
  scaled = as.data.frame(scale(data, center = min, scale = max - min))
  trainNN = scaled[1:1501+n , ]
  testNN = scaled[1502+n , ]
  NN = neuralnet(A1 ~ A2 + A3 + A4 + A5 + A6 + A7 + A8 + A9 + A10,
  trainNN, hidden = 3 , linear.output = T )
  predict_testNN = compute(NN, testNN[,c(2:10)])
  predict_testNN = (predict_testNN$net.result * (max(data$A1) - min(data$A1)))
  + min(data$A1)
  AIR.P2[n] <- predict_testNN
}
BN.P2 <- c()
library(readr)
BN1 <- read_csv2("D:/2017-2018/data_analysis/technical_analysis/data/BN1.csv")
View(BN1)
for (n in 1:30){
  datatrain = data[ 1:1501+n, ]
  datatest = data[ 1502+n, ]
  max = apply(data , 2 , max)
  min = apply(data, 2 , min)
  scaled = as.data.frame(scale(data, center = min, scale = max - min))
  trainNN = scaled[1:1501+n , ]
  testNN = scaled[1502+n , ]
  NN = neuralnet(A1 ~ A2 + A3 + A4 + A5 + A6 + A7 + A8 + A9 + A10,
  trainNN, hidden = 3 , linear.output = T )
  predict_testNN = compute(NN, testNN[,c(2:10)])
  predict_testNN = (predict_testNN$net.result * (max(data$A1) - min(data$A1)))
  + min(data$A1)
  BN.P2[n] <- predict_testNN
}
View(data)
data <- BN1
View(data)
for (n in 1:30){
  datatrain = data[ 1:1501+n, ]
  datatest = data[ 1502+n, ]
  max = apply(data , 2 , max)
  min = apply(data, 2 , min)
  scaled = as.data.frame(scale(data, center = min, scale = max - min))
  trainNN = scaled[1:1501+n , ]
```

```
  testNN = scaled[1502+n , ]
  NN = neuralnet(A1 ~ A2 + A3 + A4 + A5 + A6 + A7 + A8 + A9 + A10,
  trainNN, hidden = 3 , linear.output = T )
  predict_testNN = compute(NN, testNN[,c(2:10)])
  predict_testNN = (predict_testNN$net.result * (max(data$A1) - min(data$A1)))
  + min(data$A1)
  BN.P2[n] <- predict_testNN
}
library(readr)
BNP1 <- read_csv2("D:/2017-2018/data_analysis/technical_analysis/data/BNP1.csv")
View(BNP1)
data <- BNP1
View(data)
BNP.P2 <- c()
for (n in 1:30){
  datatrain = data[ 1:1501+n, ]
  datatest = data[ 1502+n, ]
  max = apply(data , 2 , max)
  min = apply(data, 2 , min)
  scaled = as.data.frame(scale(data, center = min, scale = max - min))
  trainNN = scaled[1:1501+n , ]
  testNN = scaled[1502+n , ]
  NN = neuralnet(A1 ~ A2 + A3 + A4 + A5 + A6 + A7 + A8 + A9 + A10,
  trainNN, hidden = 3 , linear.output = T )
  predict_testNN = compute(NN, testNN[,c(2:10)])
  predict_testNN = (predict_testNN$net.result * (max(data$A1) - min(data$A1)))
  + min(data$A1)
  BNP.P2[n] <- predict_testNN
}
DG.P2 <- c()
library(readr)
DG1 <- read_csv2("D:/2017-2018/data_analysis/technical_analysis/data/DG1.csv")
View(DG1)
data <- DG1
View(data)
for (n in 1:30){
  datatrain = data[ 1:1501+n, ]
  datatest = data[ 1502+n, ]
  max = apply(data , 2 , max)
  min = apply(data, 2 , min)
  scaled = as.data.frame(scale(data, center = min, scale = max - min))
  trainNN = scaled[1:1501+n , ]
  testNN = scaled[1502+n , ]
  NN = neuralnet(A1 ~ A2 + A3 + A4 + A5 + A6 + A7 + A8 + A9 + A10,
                 trainNN, hidden = 3 , linear.output = T )
  predict_testNN = compute(NN, testNN[,c(2:10)])
  predict_testNN = (predict_testNN$net.result *
```

```
                        (max(data$A1) - min(data$A1))) + min(data$A1)
  DG.P2[n] <- predict_testNN
}
```

We want to create a table with the predicted daily increase in price for every stock during the 30 days. After that, we will identify the stock with the highest predicted increase in price for every day. This will be the stock where to invest all the capital for the day.

```
View(AIR.P2)
View(AI1)
AI.R <- AI1[1502:1531, c(2)]
View(AI.R)
View(AI1)
View(AIR1)
AIR.R <- AIR1[1502:1531, c(1)]
View(BN1)
BN.R <- BN1[1502:1531, c(1)]
View(BNP1)
BNP.R <- BNP1[1502:1531, c(1)]
View(DG1)
DG.R <- DG1[1502:1531, c(1)]
AI.DIF <- AI.P2 - AI.R
AIR.DIF <- AIR.P2 - AIR.R
BN.DIF <- BN.P2 - BN.R
BNP.DIF <- BNP.P2 - BNP.R
DG.DIF <- DG.P2 - DG.R
df = data.frame(AI.DIF,AIR.DIF,BN.DIF,BNP.DIF,DG.DIF)
View(df)
#identify the stock with the highest predicted increase in price for every day
X <- c()
for (n in 1:30){
  if (max(df[n,]) == df[n,"A1"]){
    X[n] = "AI.PA"
  } else if (max(df[n,]) == df[n,"A1.1"]){
    X[n] = "AIR.PA"
  } else if (max(df[n,]) == df[n,"A1.2"]){
    X[n] = "BN.PA"
  } else if (max(df[n,]) == df[n,"A1.3"]){
    X[n] = "BNP.PA"
  } else if (max(df[n,]) == df[n,"A1.4"]){
    X[n] = "DG.PA"
  }
}
View(X)
```

Table 7: Multiple-input NN strategy outcome

Day	Invest in	date	AI	AIR	BNP	BN	DG	#shares bought	€ Value
1	DG	16/11/2018	95.86	92.88	65.09	45.19	77.09	129.70	10000
2	DG	19/11/2018	95.40	92.57	64.70	45.28	76.80	129.70	9961.09
3	DG	20/11/2018	93.5	91.07	64.98	44.30	76.37	129.70	9906.61
4	AIR	21/11/2018	94.22	93.16	65.41	44.79	77.01	107.21	9989.62
5	AIR	22/11/2018	93.31	92.33	65.27	44.28	76.62	107.21	9899.55
6	DG	23/11/2018	93.86	93.41	65.87	44.36	76.86	130.30	10015.35
7	DG	26/11/2018	95.86	93.58	65.82	45.34	78.54	130.30	10234.27
8	AIR	27/11/2018	94.86	93.69	66.41	45.08	78.16	108.69	10184.75
9	DG	28/11/2018	94.36	93.5	65.41	44.75	77.95	130.36	10163.01
10	DG	29/11/2018	94.68	94.86	65.54	44.77	77.58	130.36	10113.47
11	DG	30/11/2018	97.04	94.62	66.05	44.37	77.09	130.36	10050.90
12	DG	03/12/2018	97.27	95.69	65.41	44.84	75.40	130.36	9829.29
13	DG	04/12/2018	97.45	94.26	65.52	43.86	76.66	130.36	9993.54
14	AIR	05/12/2018	96.45	92.30	64.58	43.41	74.66	105.43	9732.82
15	DG	06/12/2018	94.63	88.63	63.43	41.78	70.98	131.66	9345.87
16	DG	07/12/2018	95.86	89.07	63.79	41.59	72.68	131.66	9569.70
17	AIR	10/12/2018	95.77	87.48	62.97	40.40	71.66	107.84	9435.40
18	DG	11/12/2018	96.45	88.59	62.81	40.68	73.69	129.64	9555.11
19	AI	12/12/2018	97.54	91.48	64.12	41.90	74.77	99.39	9695.13
20	DG	13/12/2018	97.63	89.97	64.32	42.18	74.5	130.25	9704.18
21	AIR	14/12/2018	97.45	88.66	64.08	41.59	73.51	108.00	9576.52
22	AIR	17/12/2018	96.68	87.65	63.23	40.58	72.36	108.00	9466.36
23	AIR	18/12/2018	95.68	88.65	62.68	40.55	71.33	108.00	9574.36
24	DG	19/12/2018	97.86	87.19	62.82	40.88	71.59	131.53	9417.76
25	AIR	20/12/2018	97.09	83.33	62.25	39.40	71.41	112.73	9394.08
26	AIR	21/12/2018	97.81	83.09	62.23	39.5	71.41	112.73	9368.16
27	AI	24/12/2018	97.09	81.62	61.61	38.79	70.5	94.78	9202.44
28	AIR	27/12/2018	95.31	82.09	60.27	38.54	70.63	110.04	9034.42
29	DG	28/12/2018	96.59	83.76	60.66	39.37	71.95	128.08	9217.09
30	AIR	31/12/2018	98.59	83.95	61.50	39.47	72.01		9224.77

4.3.1 Evaluation of Multiple-input NN

As can be seen, the multiple-input based investment has a return of -7%. This shows that this method cannot be used to predict stock prices. We believe this is because the method predicts prices based on just the previous-day values of several variables, hence without considering the time-serie behind the price trend. Instead, the 1-input NN took into account the whole trend of the prices during the previous 1 year and a half.

4.4 Neuronal Network Conclusion

To conclude, we can argue that the 1-input NN can be successfully used to invest in stocks. We recommend to further test and validate this method over a larger number of stocks (100) and over a longer time frame (1 year of predicted prices) before investing in real life with this model.

5 Random forest

5.1 What is Random Forest?

Random forest, consists of large number of individual decision trees that operate as an ensemble. Each decision tree in the forest spits out a class prediction and the class with the majority votes becomes our model's prediction. So, in this case the technique of random forest is being used for Stock Price prediction.

5.2 Why Random Forest?

As it is termed as one of the easiest machine learning algorithms according to Sadia, Sharma and Paul (2019), it gives accurate results and reduces overfitting of the model with a good accuracy in the prediction. But, because of the high volatility and instability in the stock markets, prediction has become very challenging. The random forest algorithm randomly selects different observations and features to build several decision trees and then takes aggregate of all the decision trees.

5.3 Data

The data is split into partitions based on the conditions of the attribute. The data set we used is from 1 January 2013 to 31 December 2018 Of Top 5 Stocks in France Namely Air Liquide, Airbus, BNP Paribas, Danone, Vinci. After taking these datasets we have considered 19 variables which according to us affect the stock prices. They are : Interest Rates , Exchange Rate , Political Stability , Inflation , Dividend , Revenue , Net income , Basic Earnings Per Share , Diluted Earnings Per Share , Total Assets , Intangible Assets , PPE , Cash , Total Equity, Non- Current Liability, Cash Flows From Operating Activities , Net Cash Flows From Investing Activities , Net Cash Flows From Financing Activities , Net Cash And Cash Equivalents. And, in this model we are considering the Adjusted Price to be the dependent variable. This means we see the affect of all these variables on the Adjusted prices of the stock on daily basis. But, some of the variables like political stability and inflation are constant either for the whole year or the whole month. So, there is very less volatility in some variables which show that they don't affect the prices as compared to the other variables.

5.4 Methodology

70% of the data was used to train and 30% is used to test. The basic approach of the learning model is to learn the pattern and relationships in the data from the training set and then reproduce in the test set. In this model we are doing a random forest regression by considering 19 variables that could affect the stock prices. For applying the Random Forest algorithm, we are using R language to build the model. After running a linear multiple regression for all the stocks separately, we found that which is the variable which is affecting the prices the most. Then, we divide the dataset into testing data and training data. This will help us to train the model first and then test the remaining data according the train data.

After training and testing, we apply the random forest algorithm and make predicitions.

6 Conclusions

7 References

K. Hiba Sadia, Aditya Sharma, Adarrsh Paul, SarmisthaPadhi, Saurav Sanyal. Stock Market Prediction Using Machine Learning Algorithms. International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249 – 8958, Volume-8 Issue-4, April 2019

Gerard Biau. Analysis of a Random Forests Model. Journal of Machine Learning Research 13 (2012) 1063-1095

Misha Denil, David Matheson, Nando de Freitas. Narrowing the Gap: Random Forests In Theory and In Practice. University of Oxford, United Kingdom and University of British Columbia, Canada

www.dividenmax.com www.airliquide.com www.invest.bnpparibas.com www.danone.com www.vinci.com

References

Salih N. Neftci. Naive Trading Rules in Financial Markets and Wiener-Kolmogorov Prediction Theory: A Study of "Technical Analysis". *The Journal of Business*, 64(4):549, 1991. ISSN 0021-9398. doi: 10.1086/296551.