Moneytalks - Report

Adriano Augusto - Grace Achenyo Okolo

STEP6 - Buy, Hold and Sell advisor based on classifiers

```
library(data.table)
library(dplyr)
library(tidyr)
library(ggplot2)
library(pdist)
library(caret)
data <- read.csv("dataset.csv")
```

In this last section, we try to design a simple classifier able to give advices of the type Buy, Hold or Sell at the beginning of the day, based on weekly return, monthly return and the change at the opening.

First, we label all the data. The data is labeled as follow: a Buy advice is given when the closing change is in percentage 0.50 greater; A Sell advice is given when the closing change is in percentage 0.50 smaller; otherwise we suggest to Hold. Then, we shuffle the order or the rows and divide the data in two sets, one of 80% of the entries (train set) and one of 20% of the entries (test set).

```
labeling <- function(data) {
  th <- 0.5
  data <- transform(data, a=ifelse( d_change > (o_change + th), "BUY", ifelse( d_change < (o_change - t
  return(data)
}
```

```
rdata <- data[sample(nrow(data)),]
rdata <- labeling(rdata)
```

before training the classifiers, we have to remove some features that are not needed. We will leave only the weekly change, the monthly change, and the opening change.

```
drops <- c("high", "price", "low", "volume", "d_change", "X.1", "X", "day", "month", "year", "stock", "
rdata <- rdata[ , !(names(rdata) %in% drops)]
tenp <- 38951
train <- rdata[1:(tenp*8),]
test <- rdata[((tenp*8)+1):nrow(rdata),]
filter(train, a=="SELL") %>% nrow()/(tenp*8)
```

```
## [1] 0.3119721
```

```
filter(train, a=="HOLD") %>% nrow()/(tenp*8)
```

```
## [1] 0.35105
```

```
filter(train, a=="BUY") %>% nrow()/(tenp*8)
```

```
## [1] 0.3369779
```

```
filter(test, a=="SELL") %>% nrow()/(tenp*2)
```

```
## [1] 0.3080665
```

```
filter(test, a=="HOLD") %>% nrow()/(tenp*2)
```

```
## [1] 0.350543
```

```
filter(test, a=="BUY") %>% nrow()/(tenp*2)
```

```
## [1] 0.3414033
```

The distribution of the advice SELL, HOLD and BUY is roughly 1/3 SELL, 1/3 HOLD, and 1/3 BUY, both for train and test set. This is very important to highlight, to show that there is no underfitting or overfitting bias.

Successively, we train a tree and a random forest using the default parameteres. We performe the predictions on the test set, and we show the accuracy of both classifiers.

```
ctrl <- trainControl(method="none", number = 1)

tree <- train(as.factor(a)~., data = train, method = 'C5.0Tree', trControl = ctrl)
rndf <- train(as.factor(a)~., data = train, method = 'rf', trControl = ctrl)
```

```
tree_pr <- predict(tree, test)
rndf_pr <- predict(rndf, test)
```

```
res <- data.frame(real=test$a, tree=tree_pr, rndf = rndf_pr)
```

```
t <- 0
f <- 0

for(i in 1:nrow(res)){
  if(res[i,1]==res[i,2]) t <- t+1
  if(res[i,1]==res[i,3]) f <- f+1
}

t/nrow(res)
```

```
## [1] 0.6398983
```

```
f/nrow(res)
```

```
## [1] 0.9746736
```

The accuracy for the tree classifier is 0.64, which is not enough for our requirements. However, the accuracy for the random forest is instead 0.97, which is very high and outperform the minimum requirement that we set at the beginning of the project, which was 0.75.

However, to improve the reliability of such accuracy evaluation, we will now perform a 5-fold validation. We will use only the random forest, since the tree already underperformed.

We first prepare the folds from the original shuffled dataset. Successively, we define 4 training sets and the 4 correspondent test sets. NOTE: the first fold was already analysed before, so that we will just use the previous accuracy value.

```
fold1L <- tenp*2
fold2L <- tenp*4
fold3L <- tenp*6
fold4L <- tenp*8
fold5L <- tenp*10

fold1 <- rdata[1:fold1L,]
fold2 <- rdata[(fold1L+1):fold2L,]
fold3 <- rdata[(fold2L+1):fold3L,]
fold4 <- rdata[(fold3L+1):fold4L,]
```

```
fold5 <- rdata[(fold4L+1):fold5L,]
```

```
  train2 <- rbind(fold1,fold3,fold4,fold5)
  test2 <- fold2

  train3 <- rbind(fold1,fold2,fold4,fold5)
  test3 <- fold3

  train4 <- rbind(fold1,fold2,fold3,fold5)
  test4 <- fold4

  train5 <- rbind(fold1,fold2,fold3,fold4)
  test5 <- fold5
```

Here, we train the random forests four times, and we save the results of the predictions.

```
rm(rndf)
rm(rndf_pr)
rndf <- train(as.factor(a)~., data = train2, method = 'rf', trControl = ctrl)
rndf_pr <- predict(rndf, test2)
res2 <- data.frame(real=test2$a, rndf = rndf_pr)
```

```
rm(rndf)
rm(rndf_pr)
rndf <- train(as.factor(a)~., data = train3, method = 'rf', trControl = ctrl)
rndf_pr <- predict(rndf, test3)
res3 <- data.frame(real=test3$a, rndf = rndf_pr)
```

```
rm(rndf)
rm(rndf_pr)
rndf <- train(as.factor(a)~., data = train4, method = 'rf', trControl = ctrl)
rndf_pr <- predict(rndf, test4)
res4 <- data.frame(real=test4$a, rndf = rndf_pr)
```

```
rm(rndf)
rm(rndf_pr)
rndf <- train(as.factor(a)~., data = train5, method = 'rf', trControl = ctrl)
rndf_pr <- predict(rndf, test5)
res5 <- data.frame(real=test5$a, rndf = rndf_pr)
```

Finally, we compute the accuracy for each fold, and compute the average.

```
f2 <- 0
f3 <- 0
f4 <- 0
f5 <- 0
for(i in 1:nrow(res2)){
  if(res2[i,1]==res2[i,2]) f2 <- f2+1
  if(res3[i,1]==res3[i,2]) f3 <- f3+1
  if(res4[i,1]==res4[i,2]) f4 <- f4+1
  if(res5[i,1]==res5[i,2]) f5 <- f5+1
}

f2/nrow(res)
```

```
## [1] 0.974648
```

```r
f3/nrow(res)
```

```
## [1] 0.9757006
```

```r
f4/nrow(res)
```

```
## [1] 0.9741217
```

```r
f5/nrow(res)
```

```
## [1] 0.9747378
```

```r
fold5acc <- (f+f2+f3+f4+f5)/5
```

From the results obtained, we can be confident saying that the first test on the first fold was not an outlier, neither a lucky strike. The random forest can successfully give good advices to investors with an accuracy of roughly 97%. Considering the huge amount of data used (5 years of historical data of the NASDAQ100 index), we would say the achievement reached is a reliable success. However, this success is probably due to the underlying relation between the weekly, monthly and opening changes of the stock prices; as well, the fascinating equilibrium of the price movements,

At this point, we leave the investor with the doubt of relying on a classifier to take decisions on how to invest their precious savings.