QF301. Lecture 15 In-Class Assignment.

2021-10-25

I pledge on my honor that I have not given or received any unauthorized assistance on this assignment/examination. I further pledge that I have not copied any material from a book, article, the Internet or any other source except where I have expressly cited the source.

By filling out the following fields, you are signing this pledge. No assignment will get credit without being pledged.

Name: Siddharth Iyer

CWID: 10447455

Date: 10/22/2021

Question 1 (100pt)

Question 1.1

```
CWID = 10447455 #Place here your Campus wide ID number, this will personalize
#your results, but still maintain the reproduceable nature of using seeds.
#If you ever need to reset the seed in this assignment, use this as your seed
#Papers that use -1 as this CWID variable will earn 0's so make sure you change
#this value before you submit your work.
personal = CWID %% 10000
set.seed(personal)
```

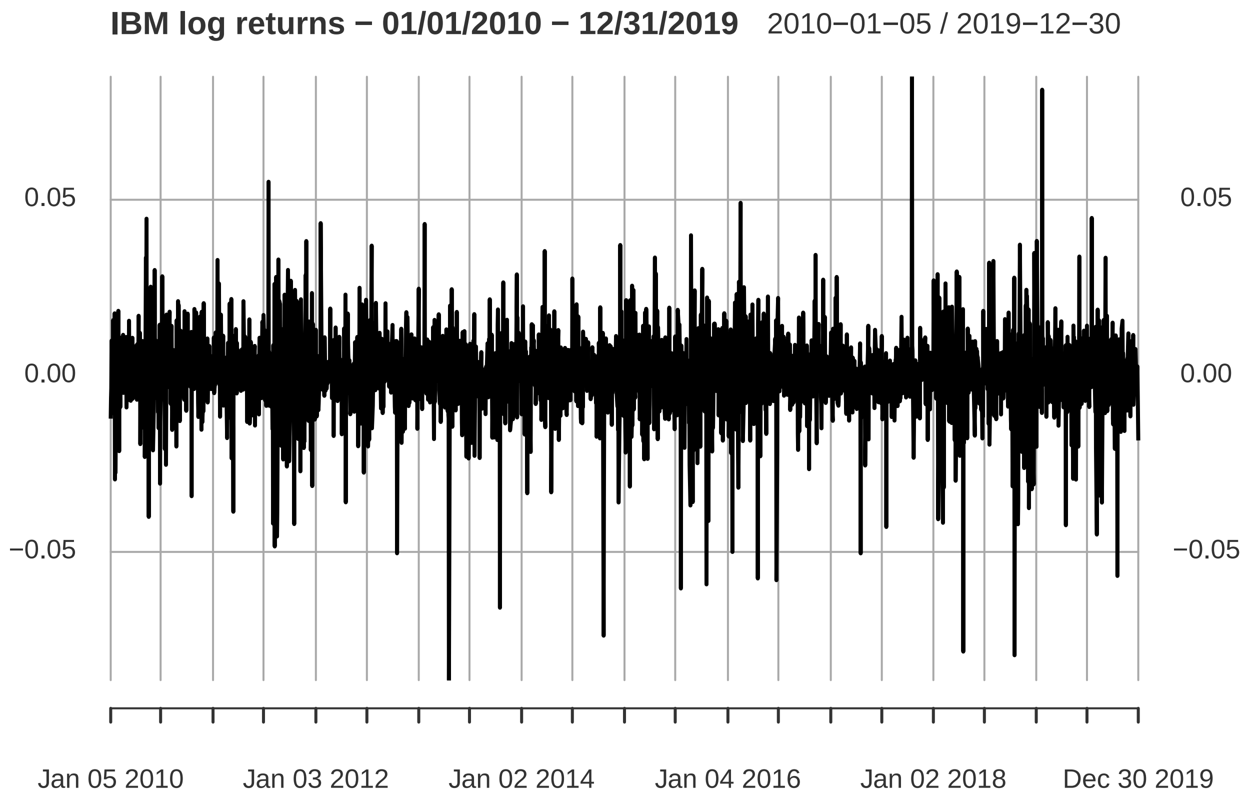
Obtain the daily log returns for IBM from January 1, 2010 until December 31, 2019. Plot these returns.

Solution:

```
library(quantmod)
getSymbols("IBM", from="2010-01-01", to="2019-12-31")
```

```
## [1] "IBM"
```

```
ibm_rets = dailyReturn(IBM$IBM.Adjusted, type = "log")[-1]
plot(ibm_rets, main = "IBM log returns - 01/01/2010 - 12/31/2019")
```



Question 1.2

Classify the direction of IBM returns (positive or negative) with a logistic regression to predict the current direction with the prior 10 days returns as inputs. Use 50% of your data for training and 50% for testing performance.
What is the accuracy? Print the confusion matrix.

Solution:

```
l0 = (ibm_rets[-10:-1] > 0)+0
l1 = as.numeric(lag(ibm_rets, k=1))[-10:-1]
l2 = as.numeric(lag(ibm_rets, k=2))[-10:-1]
l3 = as.numeric(lag(ibm_rets, k=3))[-10:-1]
l4 = as.numeric(lag(ibm_rets, k=4))[-10:-1]
l5 = as.numeric(lag(ibm_rets, k=5))[-10:-1]
l6 = as.numeric(lag(ibm_rets, k=6))[-10:-1]
l7 = as.numeric(lag(ibm_rets, k=7))[-10:-1]
l8 = as.numeric(lag(ibm_rets, k=8))[-10:-1]
l9 = as.numeric(lag(ibm_rets, k=9))[-10:-1]
l10 = as.numeric(lag(ibm_rets, k=10))[-10:-1]

df = data.frame(l0, l1, l2, l3, l4, l5, l6, l7, l8, l9, l10)
rownames(df) <- NULL
```

```r
colnames(df) <- c("l0", "l1", "l2", "l3", "l4", "l5", "l6", "l7", "l8", "l9", "l10")
head(df)
```

```
##   l0           l1           l2           l3           l4           l5
## 1  0  0.017750108 -0.004014003  0.015845710 -0.002147768  0.007923739
## 2  0 -0.029428120  0.017750108 -0.004014003  0.015845710 -0.002147768
## 3  0 -0.009643236 -0.029428120  0.017750108 -0.004014003  0.015845710
## 4  1 -0.027506950 -0.009643236 -0.029428120  0.017750108 -0.004014003
## 5  0  0.004928290 -0.027506950 -0.009643236 -0.029428120  0.017750108
## 6  1 -0.002938049  0.004928290 -0.027506950 -0.009643236 -0.029428120
##             l6           l7           l8           l9          l10
## 1 -0.010525352  0.009984588 -0.003467489 -0.006517099 -0.012153686
## 2  0.007923739 -0.010525352  0.009984588 -0.003467489 -0.006517099
## 3 -0.002147768  0.007923739 -0.010525352  0.009984588 -0.003467489
## 4  0.015845710 -0.002147768  0.007923739 -0.010525352  0.009984588
## 5 -0.004014003  0.015845710 -0.002147768  0.007923739 -0.010525352
## 6  0.017750108 -0.004014003  0.015845710 -0.002147768  0.007923739
```

```r
N <- nrow(df)
train = sample(N, .5*N, replace=FALSE)

logit.reg = glm(l0 ~ ., data=df, family=binomial, subset = train)
logit.probs=predict(logit.reg,type="response") # Predict the probability for direction on training data
logit.probs.test = predict(logit.reg , newdata=df[-train, -1] , type="response") # Compute probabilitie

# Use probabilities to make predictions
y.logit.pred=rep(0,length(train)) # Create repeated vector of "0"
y.logit.pred[logit.probs>.5] = 1 # Predict "1" if logistic regression gives greater probability to "1"

# Evaluate the accuracy
table(y.logit.pred , df[-train, 1]) # Confusion matrix of results
```

```
##
## y.logit.pred   0   1
##            0 209 230
##            1 394 419
```

```r
logistic.acc = mean(y.logit.pred==df[-train, 1]) # Directly compute the accuracy
cat("Logit Accuracy: ", logistic.acc, "\n")
```

```
## Logit Accuracy:  0.5015974
```

Question 1.3

Consider the same classification problem as in Question 1.2 but use linear regressions on one hot encoded variables. Use the same train/test split as in Question 1.2. What is the accuracy? Print the confusion matrix.

Solution:

```
df0 = data.frame(1-df[,1],df[,-1])
colnames(df0) <- c("l0", "l1", "l2", "l3", "l4", "l5", "l6", "l7", "l8", "l9", "l10")

lin1.reg = glm(l0 ~ . , data=df, subset=train)
lin0.reg = glm(l0 ~ . , data=df0, subset=train)

# Let's classify all our test points based on the greater linear regression
y1.pred = predict(lin1.reg , df[-train,-1])
y0.pred = predict(lin0.reg , df0[-train,-1])
y.lin.pred = (y1.pred > y0.pred)+0 # Again "+0" to make this 1/0

# Use predictions for classification
lin.err = 1/N * sum(abs(df[-train,1] - y.lin.pred)) # Error rate
lin.acc = 1 - lin.err # Accuracy
cat("Accuracy: ", lin.acc, "\n")
```

```
## Accuracy:  0.7507987
```

```
table(y.lin.pred , df[-train, 1])
```

```
##
## y.lin.pred   0   1
##          0 201 222
##          1 402 427
```

Question 1.4

Consider the same classification problem as in Question 1.2 but use a Naive Bayes classifier. Use the same train/test split as in Question 1.2. What is the accuracy? Print the confusion matrix.

Solution:

```
## Consider the Naive Bayes Classifier
# Note that our input data is independent in this case
library("e1071")
nb = naiveBayes(df[train,-1] , df[train,1])

y.nb.prob = predict(nb , newdata=df[-train,1] , type="raw") # Compute probabilities
y.nb.pred = (y.nb.prob[,1] < y.nb.prob[,2])+0 # 1st column is "0", 2nd column is "1"

# Evaluate the accuracy
nb.acc = mean(y.nb.pred==df[-train,1]) # Directly compute the accuracy
cat("Accuracy: ", nb.acc, "\n")
```

```
## Accuracy:  0.5183706
```

```
table(y.nb.pred , df[-train,1]) # Confusion matrix of results
```

```
##
## y.nb.pred   0   1
##         1 603 649
```

Question 1.5

Of the 3 methods considered in this question, which would you recommend in practice? Explain briefly (1 paragraph) why you choose this fit.

Solution:

The first method seems to be the best because it has fewer Type I and Type II errors compared with the other two models.