QF301. Lecture 11 In-Class Assignment.

2021-10-10

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: 10/8/2021

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)
```

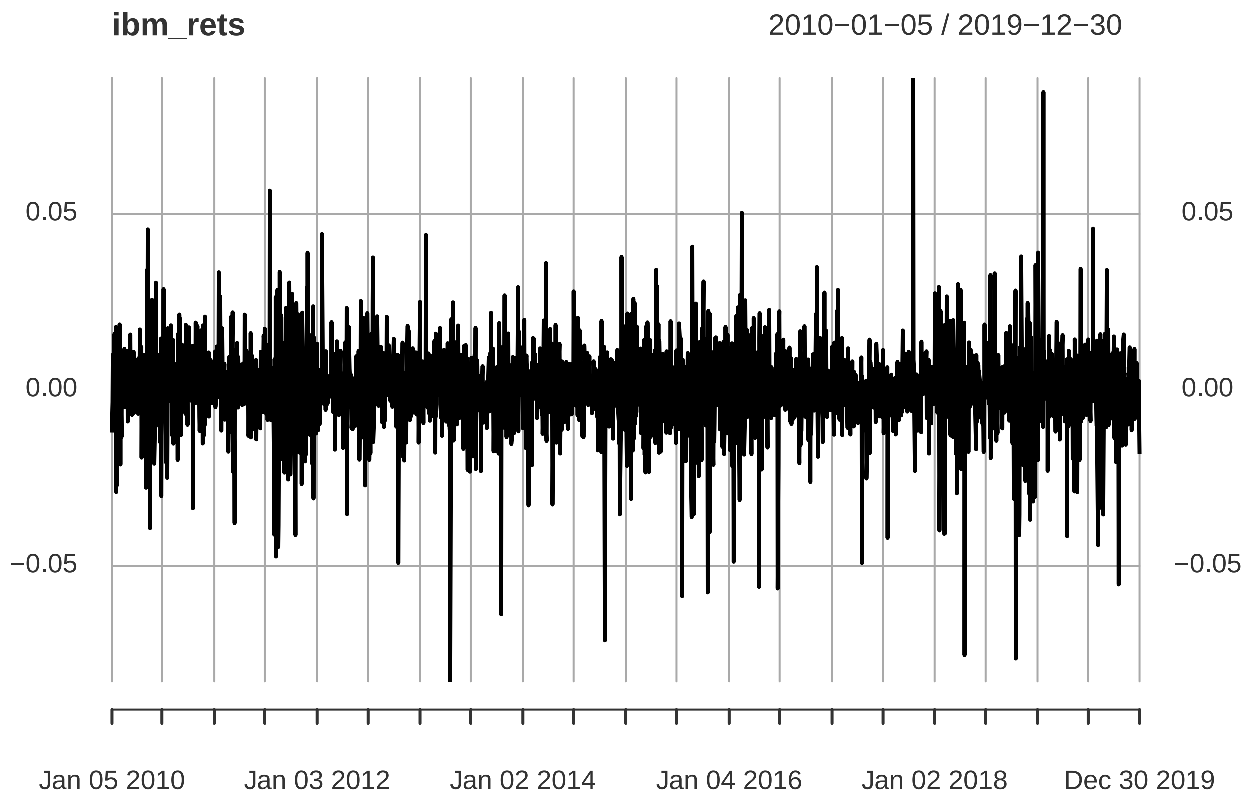
Obtain the daily log returns for IBM from January 1, 2010 until December 31, 2019. Plot these returns.

Solution:

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

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

```
ibm_rets = dailyReturn(IBM)[-1]
plot(ibm_rets)
```



Question 1.2

Create a regression tree to predict the current returns with the prior 10 days as inputs. Use 50% of your data for training and 50% for testing performance.
What is the test mean squared error? Do you suspect that your regression tree is overfitting?

Solution:

```
library(tree)
l1 = as.numeric(lag(ibm_rets$daily.returns,k=1))[-10:-1]
l2 = as.numeric(lag(ibm_rets$daily.returns,k=2))[-10:-1]
l3 = as.numeric(lag(ibm_rets$daily.returns,k=3))[-10:-1]
l4 = as.numeric(lag(ibm_rets$daily.returns,k=4))[-10:-1]
l5 = as.numeric(lag(ibm_rets$daily.returns,k=5))[-10:-1]
l6 = as.numeric(lag(ibm_rets$daily.returns,k=6))[-10:-1]
l7 = as.numeric(lag(ibm_rets$daily.returns,k=7))[-10:-1]
l8 = as.numeric(lag(ibm_rets$daily.returns,k=8))[-10:-1]
l9 = as.numeric(lag(ibm_rets$daily.returns,k=9))[-10:-1]
l10= as.numeric(lag(ibm_rets$daily.returns,k=10))[-10:-1]

df = data.frame(ibm_rets[-10:-1], l1, l2, l3, l4, l5, l6, l7, l8, l9, l10)

# training sample
```

```r
train = sample(nrow(df),nrow(df)/2,replace=FALSE)

# tree regression
tree.reg = tree(daily.returns~.,data=df,subset=train)

#Training MSE
mean((df$daily.returns[train] - predict(tree.reg, df[train,]))^2)
```

```
## [1] 0.0001566937
```

```r
# Test MSE
y.tree.pred = predict(tree.reg, df[-train,])
tree.MSE = mean((df$daily.returns[-train] - y.tree.pred)^2)
tree.MSE
```

```
## [1] 0.000150419
```

The training and test MSE are very close to each other, so I don't suspect major over fitting. If there was, there would be a huge drop in test MSE as new data comes in.

Question 1.3

Create a random forest to predict the current returns with 300 trees and using 3 predictors in each tree with the prior 10 days as inputs. Use the same train/test split as in Question 1.2. What is the test mean squared error? Do you suspect that your random forest is overfitting?

Solution:

```r
library(randomForest)
rf.ret = randomForest(daily.returns~., subset = train, data = df, importance = TRUE, mtry=3, ntree=300)
#Training MSE
mean((df$daily.returns[train] - predict(rf.ret, df[train,]))^2)
```

```
## [1] 3.197074e-05
```

```r
# Test MSE
y.rf.pred = predict(rf.ret, df[-train,])
rf.MSE = mean((df$daily.returns[-train] - y.rf.pred)^2)
rf.MSE
```

```
## [1] 0.0001565542
```

Yes, in this case the random Forest regression with 300 trees and 3 predictors is over fitting because the Test MSE is 5 times the training MSE.

Question 1.4

Create a bagged model of 300 trees to predict the current returns with the prior 10 days as inputs. Use the same train/test split as in Question 1.2. What is the test mean squared error? Do you suspect that your bagged model is overfitting?

Solution:

```r
bag.ret = randomForest(daily.returns~., subset = train, data = df, importance = TRUE, mtry=ncol(df)-1,
#Training MSE
mean((df$daily.returns[train] - predict(bag.ret, df[train,]))^2)
```

```
## [1] 2.92934e-05
```

```r
# Test MSE
y.bag.pred = predict(bag.ret, df[-train,])
bag.MSE = mean((df$daily.returns[-train] - y.bag.pred)^2)
bag.MSE
```

```
## [1] 0.0001582421
```

Yes, I still suspect overfittting in my bagged model because the test MSE is 5 times higher than the train MSE.

Question 1.5

Using multiple linear regressions, produce an AR(10) linear regression.
Use the same train/test split as in Question 1.2. What is the test mean squared error? Do you suspect that your AR(10) model is overfitting? (Hint: Look at Question 1.3 of the Lecture 5 In-Class Assignment.)

Solution:

```r
lin.reg = lm(daily.returns~., data=df, subset=train)

#Training MSE
mean((df$daily.returns[train] - predict(lin.reg, df[train,]))^2)
```

```
## [1] 0.0001558803
```

```r
# Test MSE
y.lin.pred = predict(lin.reg, df[-train,])
lin.MSE = mean((df$daily.returns[-train] - y.lin.pred)^2)
lin.MSE
```

```
## [1] 0.0001515545
```

No, the AR(10) model did not over fit because test MSE roughly = train MSE. However, the overall MSE is quite high compared with the other models above.

Question 1.6

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

Solution:

I would go for a bagged Random Forest model. Although the performance is equally atrocious compared with the others, bag RF has advantages in over fitting. Also, since it's an ensemble model, the combination of several trained models is less likely to give strange outliers because they are unlikely to occur in all models at the same time.

While it is more of a black box than the others, I would need to know whether the task requires explainable outputs. If not, bagged RF is a good tool for most ML regression tasks.