Quiz4

Jason Liu

Oct 7, 2016

Load Packages

```r
library(caret)
library(AppliedPredictiveModeling)
library(rpart)
library(ElemStatLearn)
library(pgmm)
library(rpart.plot)
library(randomForest)
library(gbm)
library(lubridate)
library(forecast)
library(e1071)
library(Metrics)
library(ggplot2)
```

Question 1

We load the data based on the instructions on the quiz page.

```r
data(vowel.train)
data(vowel.test)
vowel.train$y<-as.factor(vowel.train$y)
vowel.test$y<- as.factor(vowel.test$y)
```

Instructions: Set the variable y to be a factor variable in both the training and test set. Then set the seed to 33833. Fit (1) a random forest predictor relating the factor variable y to the remaining variables and (2) a boosted predictor using the "gbm" method. Fit these both with the train() command in the caret package.

```r
set.seed(33833)
ModelFit1<- train(y~.,data = vowel.train,method='rf')
Pred1<- predict(ModelFit1,newdata=vowel.test[,-1])
confusionMatrix(Pred1,vowel.test$y)$overall[1]

##  Accuracy
## 0.6147186
```

```r
ModelFit2<- train(y~.,data= vowel.train,method='gbm',verbose = FALSE)
Pred2<- predict(ModelFit2,newdata= vowel.test[,-1])
confusionMatrix(Pred2,vowel.test$y)$overall[1]

##  Accuracy
## 0.5367965
```

The results are quite close to one of the option in quiz. Now we try to find out the accuracy when two models agree.

```r
predDF<- data.frame(Pred1,Pred2,outcome=vowel.test$y)
sum(Pred1[predDF$Pred1 == predDF$Pred2] ==
        predDF$outcome[predDF$Pred1 == predDF$Pred2]) /
    sum(predDF$Pred1 == predDF$Pred2)

## [1] 0.6656051
```

Question 2

We load the data based on the instructions on the quiz page.

```r
library(gbm)
set.seed(3433)
library(AppliedPredictiveModeling)
data(AlzheimerDisease)
adData = data.frame(diagnosis,predictors)
inTrain = createDataPartition(adData$diagnosis, p = 3/4)[[1]]
training = adData[ inTrain,]
testing = adData[-inTrain,]
```

Instructions: Set the seed to 62433 and predict diagnosis with all the other variables using a random forest ("rf"), boosted trees ("gbm") and linear discriminant analysis ("lda") model. Stack the predictions together using random forests ("rf"). What is the resulting accuracy on the test set? Is it better or worse than each of the individual predictions?

```r
set.seed(62433)
Fit1<- train(diagnosis~.,data=training,method='rf',verbose=FALSE)
Fit2<- train(diagnosis~.,data=training,method='gbm',verbose=FALSE)
Fit3<- train(diagnosis~.,data=training,method='lda')
Pred1<- predict(Fit1,newdata=testing[,-1])
Pred2<- predict(Fit2,newdata=testing[,-1])
Pred3<- predict(Fit3,newdata=testing[,-1])
confusionMatrix(Pred1,testing$diagnosis)$overall[[1]]

## [1] 0.7682927

confusionMatrix(Pred2,testing$diagnosis)$overall[[1]]

## [1] 0.7926829
```

```
confusionMatrix(Pred3,testing$diagnosis)$overall[[1]]

## [1] 0.7682927

StackFrame<-data.frame(Pred1,Pred2,Pred3,Outcome=testing$diagnosis)
StackModel<- train(Outcome~.,data=StackFrame,method='rf')

## note: only 2 unique complexity parameters in default grid.
Truncating the grid to 2 .

Pred4<- predict(StackModel,newdata=StackFrame)
confusionMatrix(Pred4,testing$diagnosis)$overall[[1]]

## [1] 0.804878
```

Question 3

We load the data based on the instructions on the quiz page.

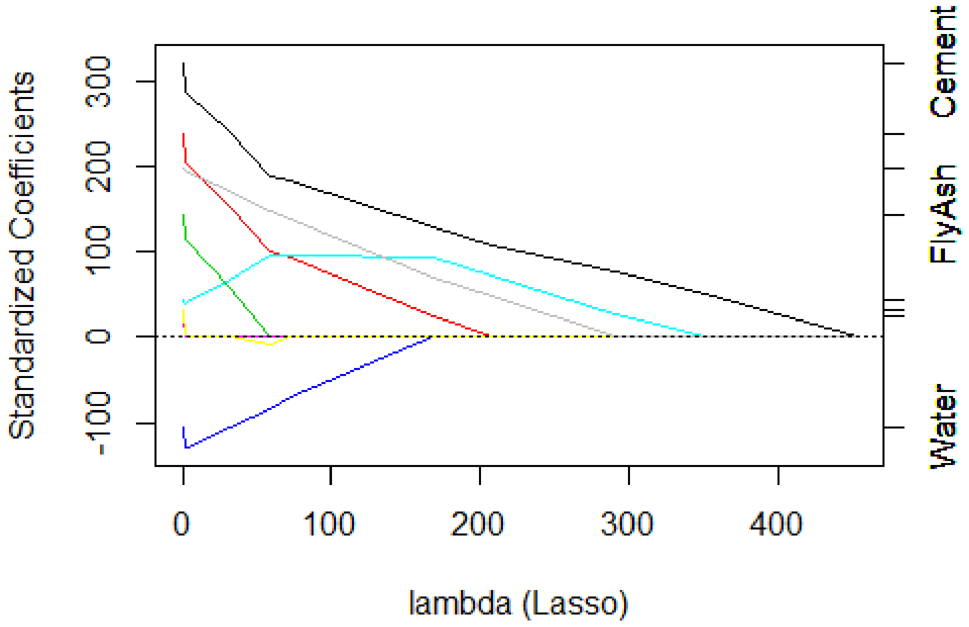
```
set.seed(3523)
library(AppliedPredictiveModeling)
data(concrete)
inTrain = createDataPartition(concrete$CompressiveStrength, p =
3/4)[[1]]
training = concrete[ inTrain,]
testing = concrete[-inTrain,]
```

Instructions: Set the seed to 233 and fit a lasso model to predict Compressive Strength. Which variable is the last coefficient to be set to zero as the penalty increases?

```
set.seed(233)
LassoFit<-train(CompressiveStrength ~.,data = training,method='lasso')
LassoFit$finalModel

##
## Call:
## enet(x = as.matrix(x), y = y, lambda = 0)
## Cp statistics of the Lasso fit
## Cp: 1201.603 1014.917  861.914  572.237  422.521  129.536  105.671
34.171    6.767    8.340    9.000
## DF: 1 2 3 4 5 6 7 7 7 8 9
## Sequence of  moves:
##      Cement Superplasticizer Age BlastFurnaceSlag Water
FineAggregate
## Var       1                5   8                2     4
7
## Step      1                2   3                4     5
6
##      FlyAsh FineAggregate FineAggregate CoarseAggregate
## Var       3            -7             7               6 11
## Step      7             8             9              10 11
```

```
plot.enet(LassoFit$finalModel, xvar = "penalty", use.color = TRUE)
```



The answer is 'Cement', which locates at the top of the right y-axle.

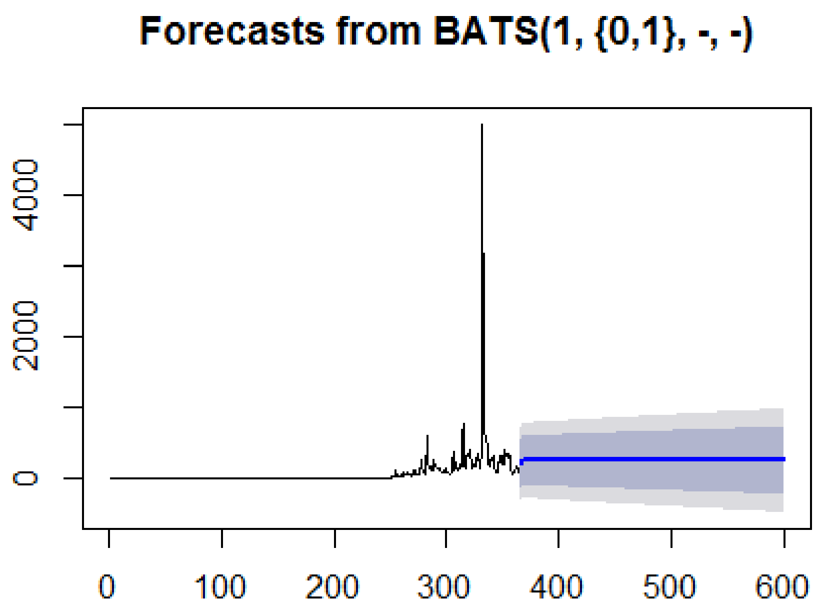
Question 4

We load the data based on the instructions on the quiz page.

```
filename_1 <- "gaData.csv"
fileUrl <-
"https://d396qusza40orc.cloudfront.net/predmachlearn/gaData.csv"
download.file(fileUrl, destfile = filename_1, method = "curl")
dat = read.csv("gaData.csv")
training = dat[year(dat$date) < 2012, ]
testing  = dat[year(dat$date) > 2011, ]
tstrain  = ts(training$visitsTumblr)
```

Instructions: Fit a model using the bats() function in the forecast package to the training time series. Then forecast this model for the remaining time points. For how many of the testing points is the true value within the 95% prediction interval bounds?

```
bats <- bats(tstrain, use.parallel = TRUE, num.cores = 4)
forecast <- forecast(bats,h=nrow(testing))
plot(forecast)
```



```
lower_bound<- forecast$lower[,2]
upper_bound<- forecast$upper[,2]
table(
  (testing$visitsTumblr>lower_bound)&
    (testing$visitsTumblr<upper_bound)

)

##
## FALSE  TRUE
##     9   226

Accuracy<- 226/235
Accuracy

## [1] 0.9617021
```

Question 5

We load the data based on the instructions on the quiz page.

```
set.seed(3523)
library(AppliedPredictiveModeling)
data(concrete)
inTrain = createDataPartition(concrete$CompressiveStrength, p =
3/4)[[1]]
```

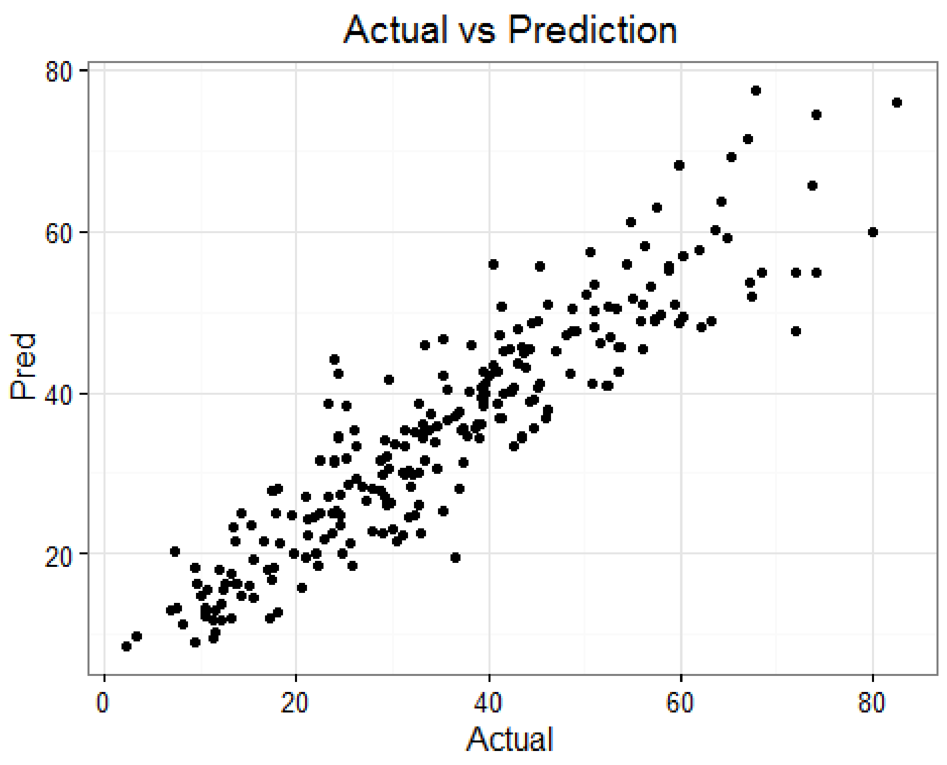
```
training = concrete[ inTrain,]
testing = concrete[-inTrain,]
```

Instructions: Set the seed to 325 and fit a support vector machine using the e1071 package to predict Compressive Strength using the default settings. Predict on the testing set.

```
set.seed(325)
ModelFit<- svm(CompressiveStrength ~., data = training)
Pred<- predict(ModelFit,newdata=testing[,-9])
rmse(testing$CompressiveStrength,Pred)

## [1] 6.715009

Tmp<- data.frame(Actual=testing$CompressiveStrength,Pred=Pred)
qplot(x = Actual,y=Pred,data=Tmp)+geom_point()+ggtitle('Actual vs
Prediction')+theme_bw()
```


Actual vs Prediction
Pred
Actual