Statistical Machine Learning – Assignment 3

Harshad Kumar Elangovan – 19200349

18/04/2020

Problem Statement:

Use multinomial logistic regression and random forests to predict the classification of the images. In doing so, compare appropriately the two models. Evaluate the generalization performance of the best model on the test data. Write a report about your findings, discussing in particular the following questions/tasks:

- Which of the two models performs better?
- Comment on the predictive performance of the best classifier.

Solution:

The dataset Satellite available in package 'mlbench' contains 6435 images displaying different scenes recorded by the Landsat satellite program. The package and required libraries for modelling the Satellite data are loaded in R. The class variable is re-ordered and factored, which is then used as the response variable for modelling the data. The data is then split for training & validating and the rest of the data is used for testing the model.

```
library(mlbench)
library(randomForest)
library(nnet)
install.packages("randomForest")

data("Satellite")
satellitedata<-data.frame(Satellite)

# this will re-order alphabetically class labels and remove spacing
satellitedata$classes <- gsub(" ", "_", satellitedata$classes)
satellitedata$classes <- factor( as.character(satellitedata$classes) )

# to have the same initial split
set.seed(19200349)
D <- nrow(Satellite)
keep <- sample(1:D, 5500)
test <- setdiff(1:D, keep)
dat <- satellitedata[keep,]
dat_test <- satellitedata[test,]




#Splitting the training data into training and validation data that can be used for fitting the model.
```

```
N<-nrow(dat)
train<-sample(1:N,size = 0.75*N)
val<-setdiff(1:N,train)
dat_train<-dat[train,]
dat_val<-dat[val,]
```

This split data is then used for modelling based on “Multinomial Logistical Regression” and “Random Forest”. Both the models are first fitted using their model function and then the fitted model is used for prediction. Model is fitted using the training data and the prediction occurs on the validation data of the fitted model.

```
#Fitting the model
# multinomial logistic regression
fitLog<-multinom(classes~.,data = dat_train)

#Random Forest
fitRF <- randomForest(classes~., data = dat_train, importance=TRUE)

#Prediction
# Multinomial logistic regression
predLogR <- predict(fitLog, type = "class", newdata = dat_val)
tabValLog <- table(dat_val$classes, predLogR)
accLog <- sum(diag(tabValLog))/sum(tabValLog)

# Random Forest Prediction
predRF<-predict(fitRF,type = "class",newdata = dat_val)
tabValRF <- table(dat_val$classes, predRF)
accRF <- sum(diag(tabValRF))/sum(tabValRF)

# print accuracy
acc <- c(Mlogistic = accLog,RandomForest = accRF)
```

The accuracy of each model is stored in ‘acc’ variable. This variable is used to verify which model has the better accuracy.

```
> acc
  Mlogistic      RandomForest
  0.8225455    0.9120000
```

From the output, we see that the model of Random Forest has the highest accuracy and can be used for modelling the test data.

```
# use the method that did best on the validation data to predict the test data
best <- names( which.max(acc) )
switch(best,
      Mlogistic = {
        predTestLog <- predict(fitLog, type = "class", newdata = dat_test)
        tabTestLog <- table(dat_test$classes, predTestLog)
```

```
      accBest <- sum(diag(tabTestLog))/sum(tabTestLog)
     },
     RandomForest = {
      predTestRF<-predict(fitRF,type = "class",newdata = dat_test)
      tabTestRF <- table(dat_test$classes, predTestRF)
      accBest <- sum(diag(tabTestRF))/sum(tabTestRF)
      }
)
```

The test data is used in Random Forest model and the accuracy is 0.8887701. But this is just one iteration. Model might provide different values if we re-run it. So, for arriving at a best model, we will have to train the data several times and then conclude which model is better.

So, we will model this satellite data in a repeated loop and train the model. The model is then used for testing the test data that is stored in dat_test variable.

```
# replicate the process a number of times
R <- 100
out <- matrix(NA, R, 4)
colnames(out) <- c("val_logistic","val_RF", "best", "test")
out <- as.data.frame(out)

for ( r in 1:R ) {

 # split the data
 keep <- sample(1:D, 5500)
 test <- setdiff(1:D, keep)
 dat <- satellitedata[keep,]
 dat_test <- satellitedata[test,]

 #Splitting the training data into training & validation data
 N<-nrow(dat)
 train<-sample(1:N,size = 0.75*N)
 val<-setdiff(1:N,train)
 dat_train<-dat[train,]
 dat_val<-dat[val,]

 #Fitting the model
 # multinomial logistic regression
 fitLog<-multinom(classes~.,data = dat_train)

 #Random Forest
 fitRF <- randomForest(classes~., data = dat_train, importance=TRUE)




 #Prediction
```

```
 # Multinomial logistic regression
 predLogR <- predict(fitLog, type = "class", newdata = dat_val)
 tabValLog <- table(dat_val$classes, predLogR)
 tabValLog
 accLog <- sum(diag(tabValLog))/sum(tabValLog)

 # Random Forest Prediction
 predRF<-predict(fitRF,type = "class",newdata = dat_val)
 tabValRF <- table(dat_val$classes, predRF)
 tabValRF
 accRF <- sum(diag(tabValRF))/sum(tabValRF)

 # print accuracy
 acc <- c(Mlogistic = accLog,RandomForest = accRF)
 out[r,1] <- accLog
 out[r,2] <- accRF

 # use the method that did best on the validation data
 # to predict the test data
 best <- names( which.max(acc) )
 switch(best,
       Mlogistic = {
         predTestLog <- predict(fitLog, type = "class", newdata = dat_test)
         tabTestLog <- table(dat_test$classes, predTestLog)
         accBest <- sum(diag(tabTestLog))/sum(tabTestLog)
       },
       RandomForest = {
         predTestRF<-predict(fitRF,type = "class",newdata = dat_test)
         tabTestRF <- table(dat_test$classes, predTestRF)
         accBest <- sum(diag(tabTestRF))/sum(tabTestRF)

       }

 )
 out[r,3] <- best
 out[r,4] <- accBest
  print(r)  # 'r' is printed to keep tract of iterations
}
```

The same fitting, predicting & testing occurs for 100 iterations and the test accuracy values are stored in the out variable. This out variable shows that for all 100 iterations, "Random Forest" model produced the highest accuracy among the two model and the test data was tested using that model.

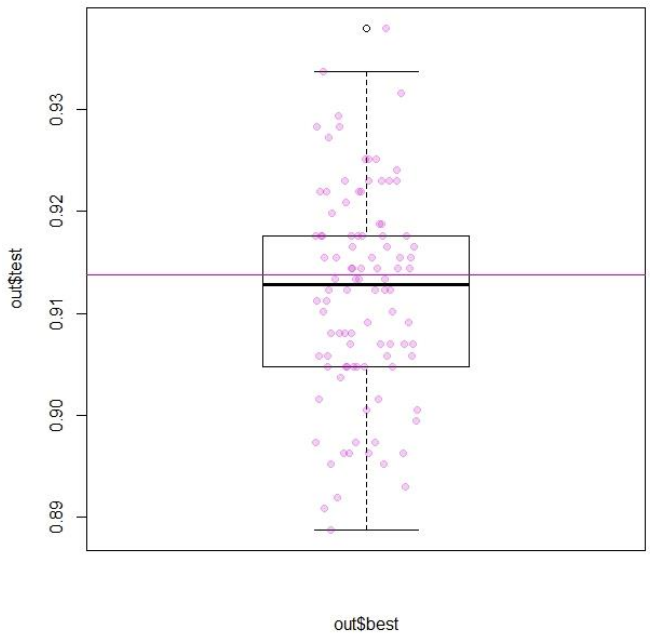
```
> head(out)
  val_logistic    val_RF       best             test
1    0.8094545 0.9192727 RandomForest 0.9133690
2    0.7760000 0.9010909 RandomForest 0.9122995
```

```
3   0.8145455 0.9134545 RandomForest 0.9058824
4   0.7614545 0.9134545 RandomForest 0.9048128
5   0.8516364 0.9134545 RandomForest 0.9101604
6   0.8043636 0.9090909 RandomForest 0.9069519
```

The values of this out variable is used for plotting the test accuracy results of the models. We see that, for all 100 iterations, Random Forest model was dominant and accuracy rate was in the range of 0.88 to 0.94.

```
# check out the error rate summary statistics
table(out[,3])
# table(out[,3])
#RandomForest
       100

boxplot(out$test ~ out$best)
stripchart(out$test ~ out$best, add = TRUE, vertical = TRUE,
        method = "jitter", pch = 19, col = adjustcolor("magenta3", 0.2))
```



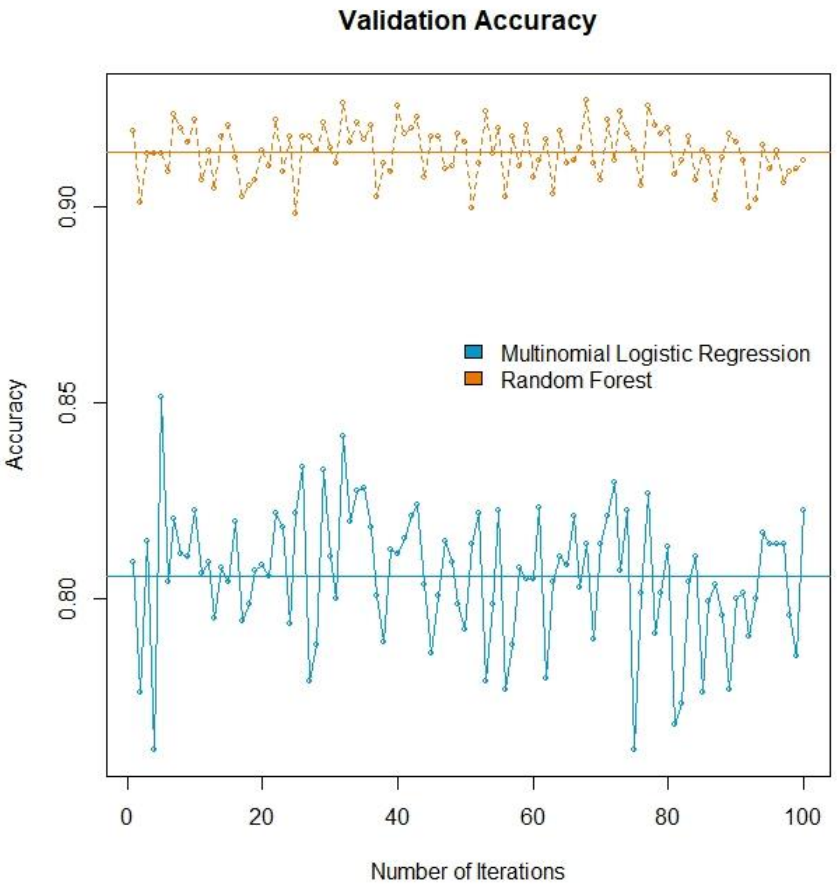

```
tapply(out[,4], out[,3], summary)

#$RandomForest
 #  Min.      1st Qu.   Median    Mean      3rd Qu.   Max.
 #0.8888     0.9048    0.9128    0.9119     0.9176    0.9380
```

From the summary, we can see that the mean accuracy of Random Forest model was 0.91 ie 91%. With minimum accuracy of 0.88 to maximum accuracy of 0.94.
So, from this we can see that, out of the two models, Random Forest model perform better.

For predictive performance of the best classifier, we can see from the summary that, the maximum accuracy is around 0.93 with a mean accuracy of 0.91.

```
matplot(out[,1:2], type = "o", pch= 1, cex= 0.5, lty= 2,
       col = c("deepskyblue3", "darkorange2"),ylab="Accuracy",
       xlab = "Number of Iterations",main="Validation Accuracy")
abline(h = c(mean(out[,1]), mean(out[,2])),
      col= c("deepskyblue3", "darkorange2"))
legend(47,0.87, fill = c("deepskyblue3", "darkorange2"),
      legend = c("Multinomial Logistic Regression","Random Forest"), bty = "n")
```



From this Mat Plot, we can see that there is no much of the variation in accuracy of Random Forest model when compared with the accuracy of Multinomial Logistical Regression. So, Random Forest is the better model and has the accuracy range of 0.88 to 0.94.

We can also re-run the Random forest model with the test data stored at the initial stage and I got the accuracy of 0.9090909 which is near to the mean accuracy of 0.91. It can vary based on sampling.