Edx Data Project

June 6, 2019

Introduction

This project explores the Iris data set from the UCI Machine Learning Repository. (https://archive.ics.uci.edu/ml/datasets/Iris) The objective is to create a model that can predict the species of iris based on physical characteristics. The key steps were performed:

Import and review the data for problems/issues/outliers

Create train and test sets

Train the models

Use test set to predict model performance

Analysis

We first imported the data and assigned the proper column names.

```
#import data(https://github.com/jrowl/edxiris)
data <- read.csv("iris.data", header=FALSE)

#set comlum names
colnames(data) <- c("sepal_length", "sepal_width", "petal_length" , "petal_width" , "class")
```

We then reviewed the data for any problems/issues/outliers.

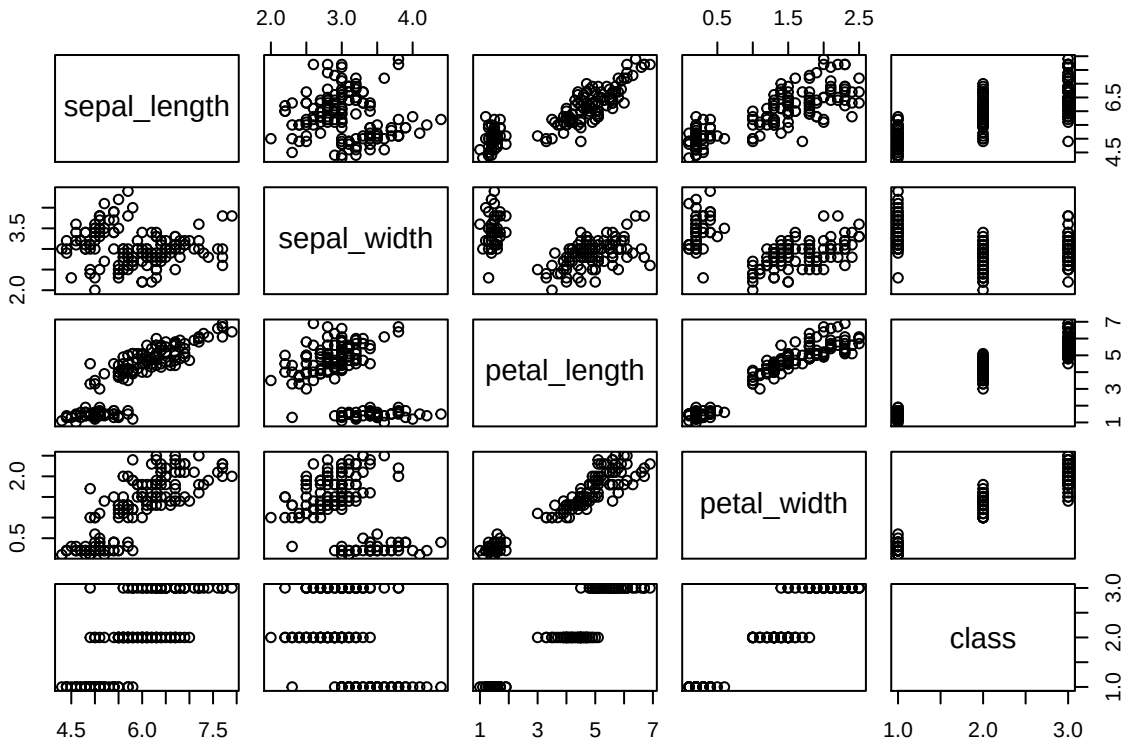
```
summary(data)
```

```
##   sepal_length    sepal_width     petal_length    petal_width
##  Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100
##  1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300
##  Median :5.800   Median :3.000   Median :4.350   Median :1.300
##  Mean   :5.843   Mean   :3.054   Mean   :3.759   Mean   :1.199
##  3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800
##  Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500
##              class
##  Iris-setosa    :50
##  Iris-versicolor:50
##  Iris-virginica :50
##
##
##
```

The summary statistics show that there are no missing values or outliers in the data sets. The means and ranges seem reasonable and there are no negative values. The observations of each class are equally split.

We then plotted the data to visually inspect for correlations or any patterns in the data.

```
#visually inspect data correlations
plot(data)
```



The data shows several correlations among the predictors. There also seems to be clumps of data that could represent parts that could be linearly separable. This is promising because we may be able to use a simple linear modeling technique.

Next we created a train and test set using 75/25.

```
#create train and test index
set.seed(918)
index <- sample(1:nrow(data), nrow(data)*.75, replace = F)
```

We then fit a model using Linear Discriminant Analysis(LDA) based on our earlier observation of linearly separable groups in the the data plots.

```
#fit LDA model
library(MASS)
fit.lda <- lda(class~. , data[index,])

#build a confusion matrix of lda model results on train data
predict.lda <- predict(fit.lda, newdata = data[index,], type = "class")
table(predict.lda$class, data$class[index])
```

```
##
```

```
##                  Iris-setosa Iris-versicolor Iris-virginica
##   Iris-setosa              39              0              0
##   Iris-versicolor           0             37              1
##   Iris-virginica            0              2             33
```

```
#calculate accuracy of model on train data
error.train.lda <- 1-(sum(predict.lda$class == data$class[index])/length(data$class[index]))
```

The LDA model successfully classified all but 3 of the observations correctly. There may be non-linearities in the model. Next we will fit a Quadratic Discriminant Analysis(QDA) model to see if it performs better.

```
#fit qda model
fit.qda <- qda(class~. , data[index,])

#build a confusion matrix of qda model results on trianing data
predict.qda <- predict(fit.qda, newdata = data[index,], type = "class")
table(predict.qda$class, data$class[index])
```

```
##
##                  Iris-setosa Iris-versicolor Iris-virginica
##   Iris-setosa              39              0              0
##   Iris-versicolor           0             37              1
##   Iris-virginica            0              2             33
```

```
#calculate accuracy of model on training data
error.train.qda <- 1-(sum(predict.qda$class == data$class[index])/length(data$class[index]))
```

The QDA model did not seem to perform any better than the LDA model. There may be interactions in the model. We will use a Decision Tree and Random Forest(RF) model next.

```
#fit and prune tree model
library(rpart)
fit.tree <- rpart(class~. , data[index,])
fit.tree <- prune(fit.tree, cp = fit.tree$cptable[min(fit.tree$cptable[,3]) == fit.tree$cptable[,3],1])

#build a confusion matrix of tree model results on trianing data
predict.tree <- predict(fit.tree, newdata = data[index,], type = "class" , cp = fit.tree$cptable[min(fi
table(predict.tree, data$class[index])
```

```
##
## predict.tree      Iris-setosa Iris-versicolor Iris-virginica
##   Iris-setosa              39              0              0
##   Iris-versicolor           0             38              2
##   Iris-virginica            0              1             32
```

```
#calculate accuracy of model on training data
error.train.tree <- 1-(sum(predict.tree == data$class[index])/length(data$class[index]))

#fit rf model
library(randomForest)
```

```
## randomForest 4.6-14
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
fit.rf <- randomForest(class~. , data[index,])

#build a confusion matrix of rf model results on trianing data
predict.rf <- predict(fit.rf, newdata = data[index,], type = "class")
table(predict.rf, data$class[index])
```

```
##
## predict.rf        Iris-setosa Iris-versicolor Iris-virginica
##   Iris-setosa              39               0              0
##   Iris-versicolor           0              39              0
##   Iris-virginica            0               0             34
```

```
#calculate accuracy of model on training data
error.train.rf <- 1-(sum(predict.rf == data$class[index])/length(data$class[index]))
```

The RF correctly classified all observations of the train data while the decision tree miss-classified 3. There may have been interactions between the variables that the RF model was able to find or it may have overtrained.

#Results

Finally we will test the models with the 25% of the data we held back from the training set to get an estimate of model performance.

```
#lda test set performance

predict.lda <- predict(fit.lda, newdata = data[-index,], type = "class")
error.test.lda <- 1-(sum(predict.lda$class == data$class[-index])/length(data$class[-index]))

#qda test set performance

predict.qda <- predict(fit.qda, newdata = data[-index,], type = "class")
error.test.qda <- 1-(sum(predict.qda$class == data$class[-index])/length(data$class[-index]))

#tree test set petreeormance

predict.tree <- predict(fit.tree, newdata = data[-index,], type = "class")
error.test.tree <- 1-(sum(predict.tree == data$class[-index])/length(data$class[-index]))

#rf test set performance

predict.rf <- predict(fit.rf, newdata = data[-index,], type = "class")
error.test.rf <- 1-(sum(predict.rf == data$class[-index])/length(data$class[-index]))

#build comparison table
comparison <- data.frame(matrix(ncol=2, nrow=4))
colnames(comparison) <- c("train", "test")
rownames(comparison) <- c("lda", "qda", "tree" ,"rf")

comparison[1,1] <- error.train.lda
comparison[1,2] <- error.test.lda
comparison[2,1] <- error.train.qda
```

```
comparison[2,2] <- error.test.qda
comparison[3,1] <- error.train.tree
comparison[3,2] <- error.test.tree
comparison[4,1] <- error.train.rf
comparison[4,2] <- error.test.rf

round(comparison, 3)
```

```
##      train  test
## lda  0.027 0.000
## qda  0.027 0.026
## tree 0.027 0.079
## rf   0.000 0.079
```

#Conclusion

While the Random Forest model performed the best on the train set, the LDA model performed best on the left out test set. As a result the LDA model is the best overall model for classifying Iris species in this data set from this experiment.