IRIS Data Capstone Project

Introduction

For the IRIS project, we use the iris dataset from Kaggle. We aim to analyse the data and predict species on the basis of sepal and petal size by classification model. Accuracy will be used to evaluate how close our predictions. There are 150 observations, Model will use 80% records to train the model and remaining 20% data will be used to test its performance. It includes three iris species with 50 samples each as well as some properties about each flower. This is a calssification problem. samples belong to two or more classes and we want to learn from already labeled data how to predict the class of unlabeled data.

During this exercise we will perform the following: -Load and Check Data -Split the data -Summarize data -Visualize Data -Build Models -Make Prediction

1. Load and Check Data

```
#Load libraries
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 3.4.4
```

```
## -- Attaching packages ---------------------------------------------------------------------------------
--------------------------- tidyverse 1.2.1 --
```

```
## v ggplot2 3.2.0       v purrr   0.3.2
## v tibble  2.1.1       v dplyr   0.8.0.1
## v tidyr   0.8.3       v stringr 1.4.0
## v readr   1.3.1       v forcats 0.4.0
```

```
## Warning: package 'tibble' was built under R version 3.4.4
```

```
## Warning: package 'tidyr' was built under R version 3.4.4
```

```
## Warning: package 'readr' was built under R version 3.4.4
```

```
## Warning: package 'purrr' was built under R version 3.4.4
```

```
## Warning: package 'dplyr' was built under R version 3.4.4
```

```
## Warning: package 'stringr' was built under R version 3.4.4
```

```
## Warning: package 'forcats' was built under R version 3.4.4
```

```
## -- Conflicts ------------------------------------------------------------------------------------------
--------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(kableExtra)
```

```
## Warning: package 'kableExtra' was built under R version 3.4.4
```

```
##
## Attaching package: 'kableExtra'
```

```
## The following object is masked from 'package:dplyr':
##
##     group_rows
```

```
library(gridExtra)
```

```
## Warning: package 'gridExtra' was built under R version 3.4.4
```

```
##
## Attaching package: 'gridExtra'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
library(caret)
```

```
## Warning: package 'caret' was built under R version 3.4.4
```

```
## Loading required package: lattice
```

```
##
## Attaching package: 'caret'
```

```
## The following object is masked from 'package:purrr':
##
##     lift
```

```
#Load dataset

iris <- read.csv('Iris.csv')
```

```
#Checking dimensions,
dim(iris)
```

```
## [1] 150   6
```

iris data have 150 observations & 6 features

```
#checking count of each species
table(iris$Species)
```

```
##
##     Iris-setosa Iris-versicolor  Iris-virginica
##              50              50              50
```

All the three species have equal number of observation.

2. Split the Data We will split the loaded dataset into two, 80% of which we will use to train our models and 20% will be used to test our model.

```
# Split the data
iris <- iris[,2:6]
sample <- createDataPartition(iris$Species, p=0.80, list=FALSE)

# Create training data
iris_train <- iris[sample,]
str(iris_train)
```

```
## 'data.frame':    120 obs. of  5 variables:
##  $ SepalLengthCm: num  5.1 4.9 4.7 4.6 5 4.6 4.4 4.9 5.4 4.8 ...
##  $ SepalWidthCm : num  3.5 3 3.2 3.1 3.6 3.4 2.9 3.1 3.7 3.4 ...
##  $ PetalLengthCm: num  1.4 1.4 1.3 1.5 1.4 1.4 1.4 1.5 1.5 1.6 ...
##  $ PetalWidthCm : num  0.2 0.2 0.2 0.2 0.2 0.3 0.2 0.1 0.2 0.2 ...
##  $ Species      : Factor w/ 3 levels "Iris-setosa",..: 1 1 1 1 1 1 1 1 1 1 ...
```

```r
# Create test data
iris_test <- iris[-sample,]
str(iris_test)
```

```
## 'data.frame':    30 obs. of  5 variables:
##  $ SepalLengthCm: num  5.4 5 5.8 5.1 4.7 4.5 4.4 5.1 4.6 5.3 ...
##  $ SepalWidthCm : num  3.9 3.4 4 3.5 3.2 2.3 3.2 3.8 3.2 3.7 ...
##  $ PetalLengthCm: num  1.7 1.5 1.2 1.4 1.6 1.3 1.3 1.9 1.4 1.5 ...
##  $ PetalWidthCm : num  0.4 0.2 0.2 0.3 0.2 0.3 0.2 0.4 0.2 0.2 ...
##  $ Species      : Factor w/ 3 levels "Iris-setosa",..: 1 1 1 1 1 1 1 1 1 1 ...
```

The training dataset has 120 observations and testing data has 30.

3. Summarize Data

a. Instances by Class

```r
# Summarize the class distribution
percentage <- prop.table(table(iris_train$Species))*100
cbind(Freq = table(iris_train$Species), Percentage = percentage)
```

```
##                 Freq Percentage
## Iris-setosa       40   33.33333
## Iris-versicolor   40   33.33333
## Iris-virginica    40   33.33333
```

Observation- We can see that each class has the same number of instances (40 or 33% of the dataset).

b. Summarize the dataset

```r
#Checking iris datset summary
summary(iris_train)
```

```
##  SepalLengthCm    SepalWidthCm   PetalLengthCm    PetalWidthCm  
##  Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100  
##  1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.575   1st Qu.:0.300  
##  Median :5.700   Median :3.000   Median :4.250   Median :1.300  
##  Mean   :5.845   Mean   :3.060   Mean   :3.751   Mean   :1.202  
##  3rd Qu.:6.425   3rd Qu.:3.325   3rd Qu.:5.100   3rd Qu.:1.825  
##  Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500  
##             Species  
##  Iris-setosa    :40  
##  Iris-versicolor:40  
##  Iris-virginica :40  
##                      
##                      
## 
```

Observation- We can see that all of the numerical values have the same scale (centimeters) and similar ranges [0,8] centimeters.No missing value in the datset

```r
#Average length & width by species
temp_df <- iris_train %>% group_by(Species) %>% summarize(mean(SepalLengthCm),mean(SepalWidthCm),mean(PetalL
engthCm),mean(PetalWidthCm))
kable(temp_df,align = 'c',col.names = c('Species','Avg Sepal Length','Avg Sepal Width','Avg Petal Length','A
vg Petal Width'))
```

Species	Avg Sepal Length	Avg Sepal Width	Avg Petal Length	Avg Petal Width
Iris-setosa	5.0100	3.4175	1.4600	0.2400
Iris-versicolor	5.9075	2.7625	4.2225	1.3050
Iris-virginica	6.6175	3.0000	5.5700	2.0625

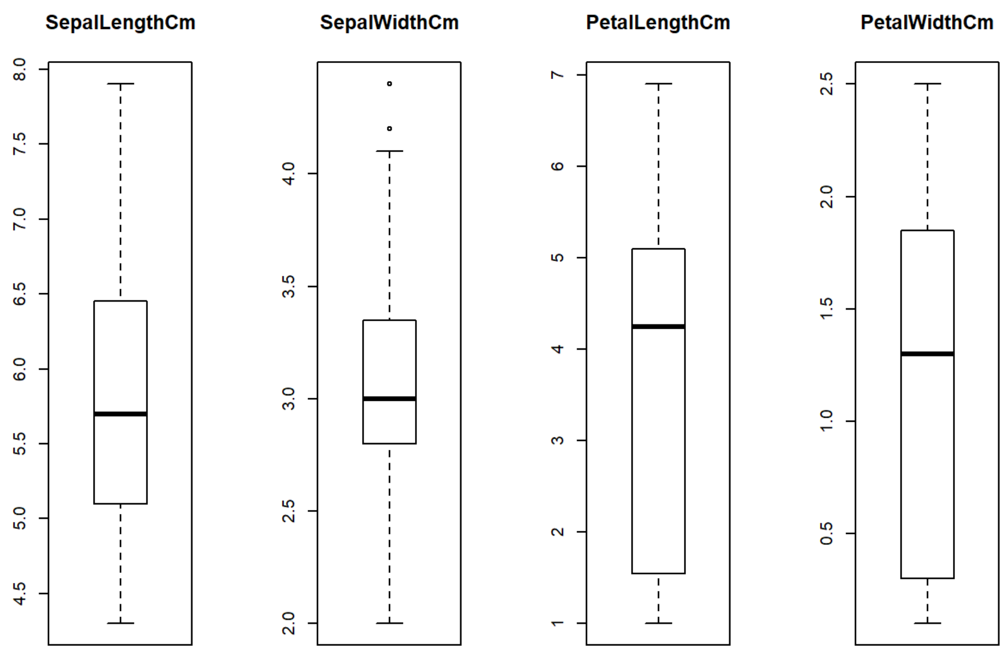
4. Visualize Data

a. Univariate plots Let's understand the input variable and output variable separately

```r
input_var <- iris_train[,1:4]
output_var <- iris_train[,5]
```

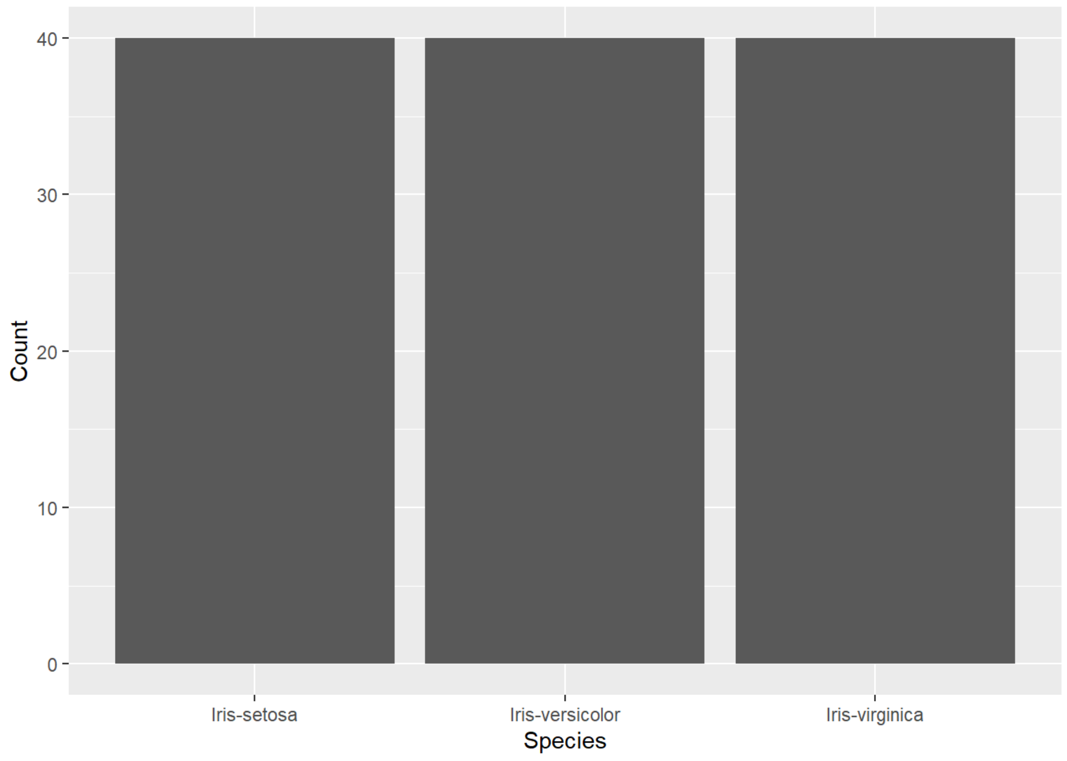
Input varibale plot:

```
par(mfrow=c(1,4))
for (i in 1:4) {
  boxplot(input_var[i], main=names(iris_train)[i])
}
```



Output variable plot

```
qplot(output_var, xlab='Species', ylab = 'Count')
```



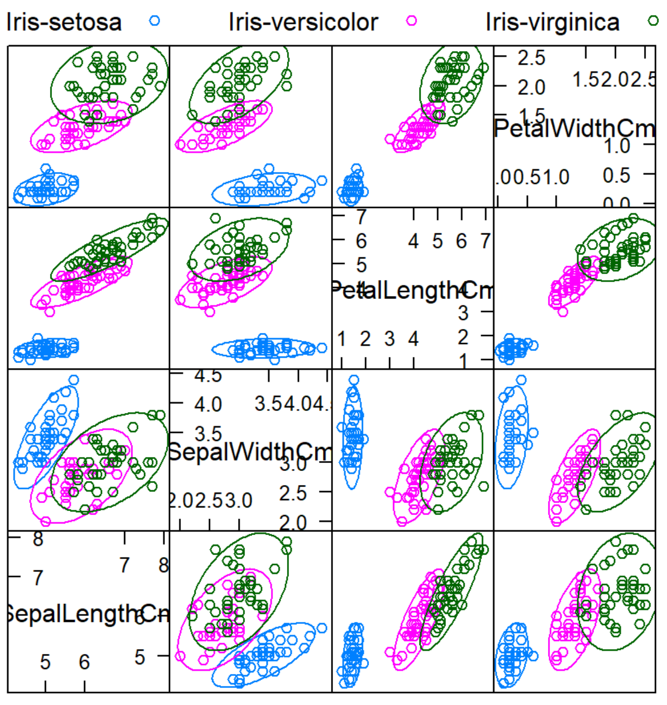
```
iris_train
```

```
##     SepalLengthCm SepalWidthCm PetalLengthCm PetalWidthCm         Species
## 1             5.1          3.5           1.4          0.2     Iris-setosa
## 2             4.9          3.0           1.4          0.2     Iris-setosa
## 3             4.7          3.2           1.3          0.2     Iris-setosa
## 4             4.6          3.1           1.5          0.2     Iris-setosa
## 5           5.0         3.6          1.4         0.2     Iris-setosa
## 7           4.6         3.4          1.4         0.3     Iris-setosa
## 9           4.4         2.9          1.4         0.2     Iris-setosa
## 10          4.9         3.1          1.5         0.1     Iris-setosa
## 11          5.4         3.7          1.5         0.2     Iris-setosa
## 12          4.8         3.4          1.6         0.2     Iris-setosa
## 13          4.8         3.0          1.4         0.1     Iris-setosa
## 14          4.3         3.0          1.1         0.1     Iris-setosa
## 16          5.7         4.4          1.5         0.4     Iris-setosa
## 17          5.4         3.9          1.3         0.4     Iris-setosa
## 19          5.7         3.8          1.7         0.3     Iris-setosa
## 20          5.1         3.8          1.5         0.3     Iris-setosa
## 21          5.4         3.4          1.7         0.2     Iris-setosa
## 22          5.1         3.7          1.5         0.4     Iris-setosa
## 23          4.6         3.6          1.0         0.2     Iris-setosa
## 24          5.1         3.3          1.7         0.5     Iris-setosa
## 25          4.8         3.4          1.9         0.2     Iris-setosa
## 26          5.0         3.0          1.6         0.2     Iris-setosa
## 27          5.0         3.4          1.6         0.4     Iris-setosa
## 28          5.2         3.5          1.5         0.2     Iris-setosa
## 29          5.2         3.4          1.4         0.2     Iris-setosa
## 31          4.8         3.1          1.6         0.2     Iris-setosa
## 32          5.4         3.4          1.5         0.4     Iris-setosa
## 33          5.2         4.1          1.5         0.1     Iris-setosa
## 34          5.5         4.2          1.4         0.2     Iris-setosa
## 35          4.9         3.1          1.5         0.1     Iris-setosa
## 36          5.0         3.2          1.2         0.2     Iris-setosa
## 37          5.5         3.5          1.3         0.2     Iris-setosa
## 38          4.9         3.1          1.5         0.1     Iris-setosa
## 39          4.4         3.0          1.3         0.2     Iris-setosa
## 40          5.1         3.4          1.5         0.2     Iris-setosa
## 41          5.0         3.5          1.3         0.3     Iris-setosa
## 44          5.0         3.5          1.6         0.6     Iris-setosa
## 46          4.8         3.0          1.4         0.3     Iris-setosa
## 47          5.1         3.8          1.6         0.2     Iris-setosa
## 50          5.0         3.3          1.4         0.2     Iris-setosa
## 51          7.0         3.2          4.7         1.4 Iris-versicolor
## 52          6.4         3.2          4.5         1.5 Iris-versicolor
## 53          6.9         3.1          4.9         1.5 Iris-versicolor
## 54          5.5         2.3          4.0         1.3 Iris-versicolor
## 56          5.7         2.8          4.5         1.3 Iris-versicolor
## 57          6.3         3.3          4.7         1.6 Iris-versicolor
## 58          4.9         2.4          3.3         1.0 Iris-versicolor
## 59          6.6         2.9          4.6         1.3 Iris-versicolor
## 60          5.2         2.7          3.9         1.4 Iris-versicolor
## 61          5.0         2.0          3.5         1.0 Iris-versicolor
## 63          6.0         2.2          4.0         1.0 Iris-versicolor
## 65          5.6         2.9          3.6         1.3 Iris-versicolor
## 66          6.7         3.1          4.4         1.4 Iris-versicolor
## 67          5.6         3.0          4.5         1.5 Iris-versicolor
## 68          5.8         2.7          4.1         1.0 Iris-versicolor
## 69          6.2         2.2          4.5         1.5 Iris-versicolor
## 70          5.6         2.5          3.9         1.1 Iris-versicolor
## 72          6.1         2.8          4.0         1.3 Iris-versicolor
## 74          6.1         2.8          4.7         1.2 Iris-versicolor
## 76          6.6         3.0          4.4         1.4 Iris-versicolor
## 77          6.8         2.8          4.8         1.4 Iris-versicolor
## 78          6.7         3.0          5.0         1.7 Iris-versicolor
## 79          6.0         2.9          4.5         1.5 Iris-versicolor
## 81          5.5         2.4          3.8         1.1 Iris-versicolor
## 82          5.5         2.4          3.7         1.0 Iris-versicolor
## 83          5.8         2.7          3.9         1.2 Iris-versicolor
## 85          5.4         3.0          4.5         1.5 Iris-versicolor
## 86          6.0         3.4          4.5         1.6 Iris-versicolor
## 88          6.3         2.3          4.4         1.3 Iris-versicolor
## 89          5.6         3.0          4.1         1.3 Iris-versicolor
## 90          5.5         2.5          4.0         1.3 Iris-versicolor
## 91          5.5         2.6          4.4         1.2 Iris-versicolor
## 92          6.1         3.0          4.6         1.4 Iris-versicolor
## 93          5.8         2.6          4.0         1.2 Iris-versicolor
## 95          5.6         2.7          4.2         1.3 Iris-versicolor
## 96          5.7         3.0          4.2         1.2 Iris-versicolor
## 97          5.7         2.9          4.2         1.3 Iris-versicolor
## 98             6.2          2.9           4.3          1.3 Iris-versicolor
## 99             5.1          2.5           3.0          1.1 Iris-versicolor
## 100            5.7          2.8           4.1          1.3 Iris-versicolor
## 101            6.3          3.3           6.0          2.5  Iris-virginica
## 102            5.8          2.7           5.1          1.9  Iris-virginica
## 105            6.5          3.0           5.8          2.2  Iris-virginica
## 106            7.6          3.0           6.6          2.1  Iris-virginica
## 108            7.3          2.9           6.3          1.8  Iris-virginica
## 109            6.7          2.5           5.8          1.8  Iris-virginica
## 110            7.2          3.6           6.1          2.5  Iris-virginica
## 111            6.5          3.2           5.1          2.0  Iris-virginica
## 113            6.8          3.0           5.5          2.1  Iris-virginica
## 114            5.7          2.5           5.0          2.0  Iris-virginica
## 115            5.8          2.8           5.1          2.4  Iris-virginica
## 116            6.4          3.2           5.3          2.3  Iris-virginica
## 117            6.5          3.0           5.5          1.8  Iris-virginica
## 118            7.7          3.8           6.7          2.2  Iris-virginica
## 119            7.7          2.6           6.9          2.3  Iris-virginica
## 120            6.0          2.2           5.0          1.5  Iris-virginica
## 121            6.9          3.2           5.7          2.3  Iris-virginica
## 122            5.6          2.8           4.9          2.0  Iris-virginica
## 125            6.7          3.3           5.7          2.1  Iris-virginica
## 127            6.2          2.8           4.8          1.8  Iris-virginica
## 128            6.1          3.0           4.9          1.8  Iris-virginica
## 130            7.2          3.0           5.8          1.6  Iris-virginica
## 131            7.4          2.8           6.1          1.9  Iris-virginica
## 132            7.9          3.8           6.4          2.0  Iris-virginica
## 133            6.4          2.8           5.6          2.2  Iris-virginica
## 134            6.3          2.8           5.1          1.5  Iris-virginica
## 135            6.1          2.6           5.6          1.4  Iris-virginica
## 136            7.7          3.0           6.1          2.3  Iris-virginica
## 137            6.3          3.4           5.6          2.4  Iris-virginica
## 140            6.9          3.1           5.4          2.1  Iris-virginica
## 141            6.7          3.1           5.6          2.4  Iris-virginica
## 142            6.9          3.1           5.1          2.3  Iris-virginica
## 143            5.8          2.7           5.1          1.9  Iris-virginica
## 144            6.8          3.2           5.9          2.3  Iris-virginica
## 145            6.7          3.3           5.7          2.5  Iris-virginica
## 146            6.7          3.0           5.2          2.3  Iris-virginica
## 147            6.3          2.5           5.0          1.9  Iris-virginica
## 148            6.5          3.0           5.2          2.0  Iris-virginica
## 149            6.2          3.4           5.4          2.3  Iris-virginica
## 150            5.9          3.0           5.1          1.8  Iris-virginica
```

b. Multivariate Plots

Scatter Plot: Let's look at scatter plots of all pairs of attributes and color the points by class.

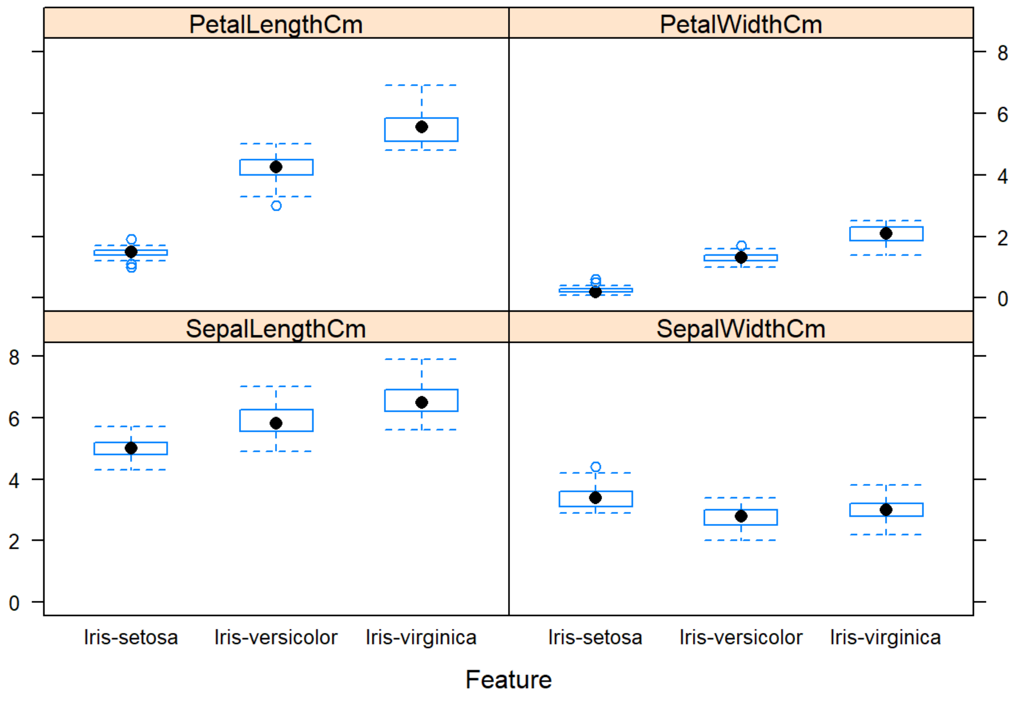
```
featurePlot(x=input_var, y=output_var, plot='ellipse', auto.key=list(columns=3))
```



Scatter Plot Matrix

Box Plots: Let’s look at the box plots of each attribute, broken down into separate plots for each class

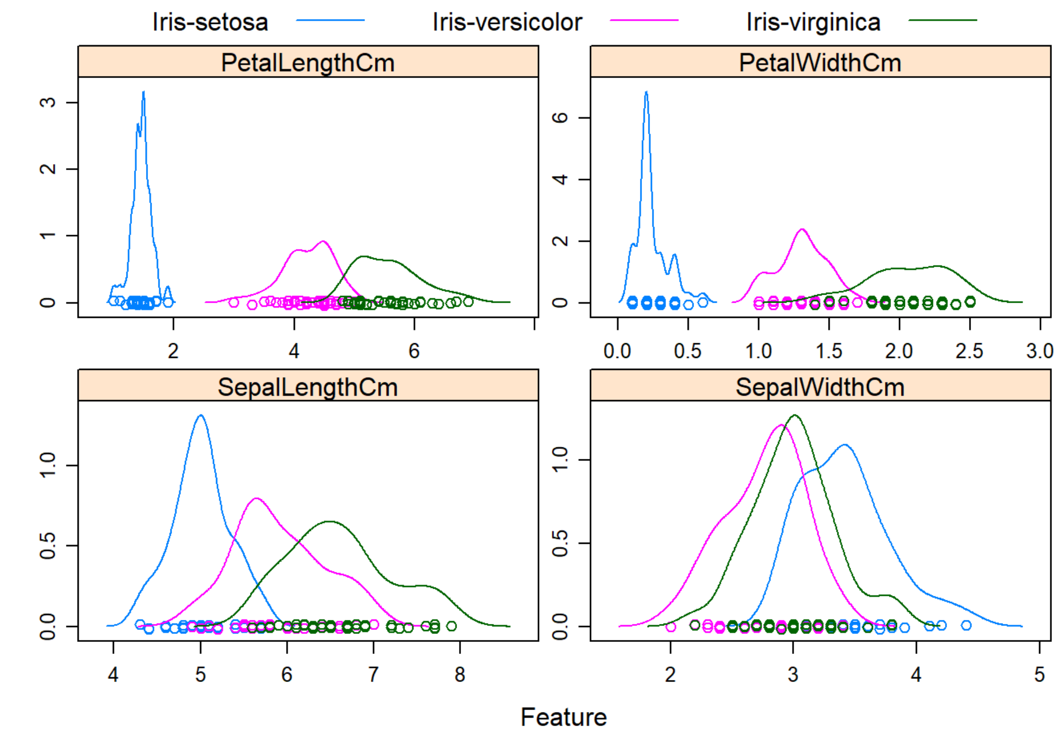
```
featurePlot(x=input_var, y=output_var, plot='box', auto.key=list(columns=3))
```



Density Plots;

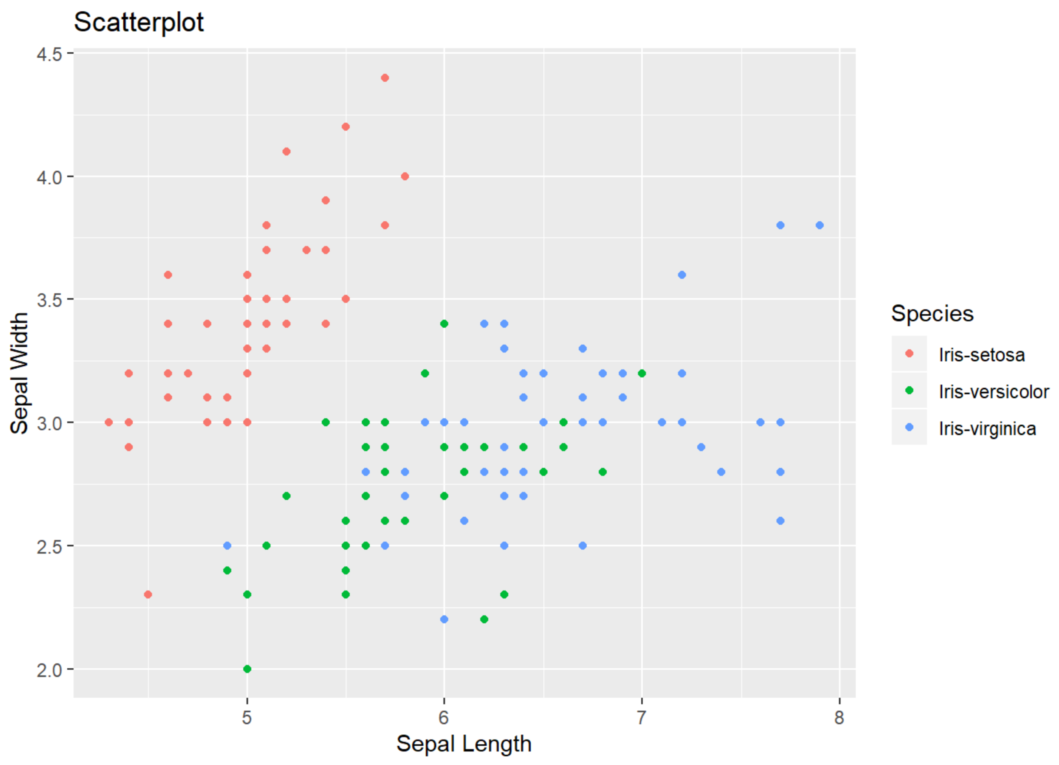
Let’s look at the distribution of each attribute, broken down into separate plots for each class using density plots

```
featurePlot(x=input_var, y=output_var,
            plot='density',
            scales = list(x = list(relation='free'),
                          y = list(relation='free')),
            auto.key=list(columns=3))
```



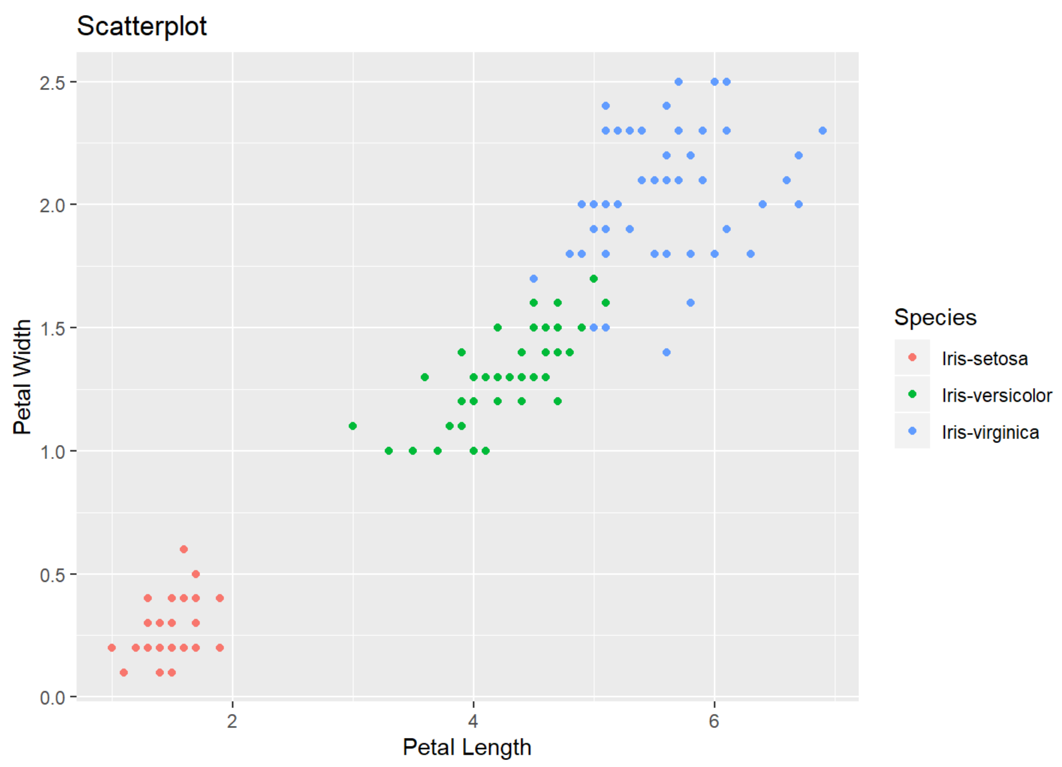
Observation- Like boxplots, we can see the difference in distribution of each attribute by class value. We can also see the bell curve of each attribute.

```
#Scatter plot between Sepel Length & Sepel Width:
ggplot(iris, aes(x=SepalLengthCm, y=SepalWidthCm, color=Species)) + geom_point() + labs(title="Scatterplot",
x="Sepal Length", y="Sepal Width")
```



The above graph shows relationship between the sepal length and width. Now we will check relationship between the petal length and width.

```
#Scatter plot between Petal Length & Petal Width:
ggplot(iris, aes(x=PetalLengthCm, y=PetalWidthCm, color=Species)) + geom_point() + labs(title="Scatterplot",
x="Petal Length", y="Petal Width")
```



As we can see that the Petal Features are giving a better cluster division compared to the Sepal features. This is an indication that the Petals can help in better and accurate Predictions over the Sepal. We will check that later.

5. Build Models

We will perform the below models to predict species and will compare the accuracy and finds the best model. - Linear Discriminant Analysis (LDA) - Classification and Regression Trees (CART) - k-nearest Neighbours (KNN) - Support Vector Machines (SVM) - Random FOrest (RF)

EValuate algorithms and pick the best model:

We will use 10-fold cross validation to estimate accuracy. This will split our dataset into 10 parts, train in 9 and test on 1 and repeat for all combinations of train-test splits.

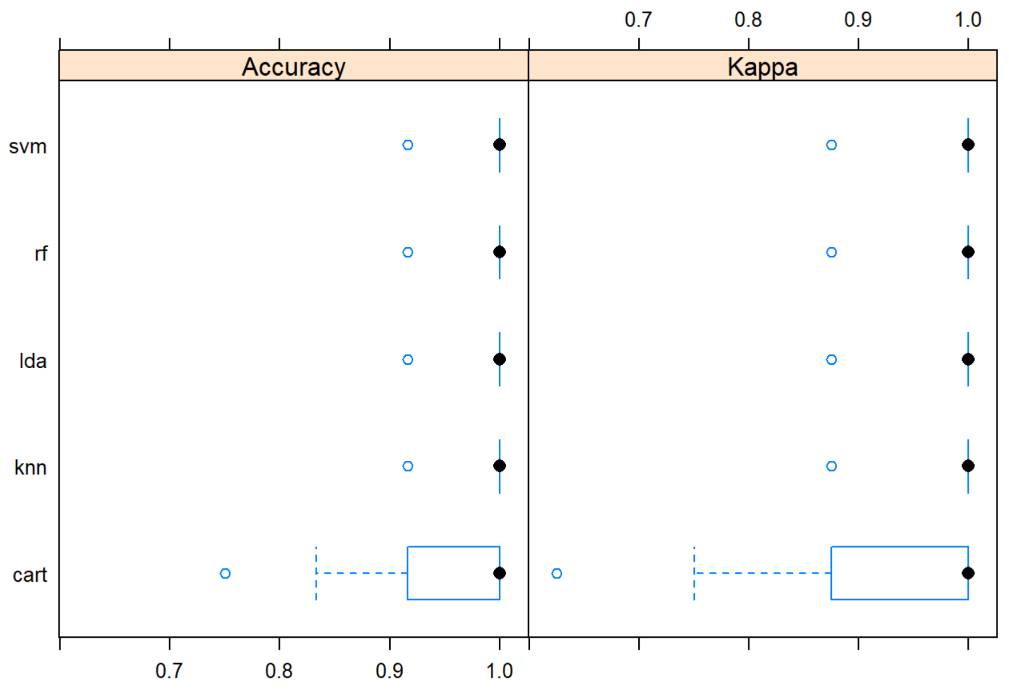
We will use a mixture of simple linear (LDA), nonlinear (CART, KNN) and complex nonlinear methods (SVM, RF).

```
control <- trainControl(method='cv', number=10)
metric <- 'Accuracy'

# Linear Discriminant Analysis (LDA)
set.seed(101)
fit.lda <- train(Species~., data=iris_train, method='lda',
                  trControl=control, metric=metric)


# Classification and Regression Trees (CART)
set.seed(101)
fit.cart <- train(Species~., data=iris_train, method='rpart',
                  trControl=control, metric=metric)


# k-Nearest Neighbors (KNN)
set.seed(101)
fit.knn <- train(Species~., data=iris_train, method='knn',
                  trControl=control, metric=metric)

# Support Vector Machines (SVM) with a radial kernel
set.seed(101)
fit.svm <- train(Species~., data=iris_train, method='svmRadial',
                  trControl=control, metric=metric)

# Random Forest (RF)
set.seed(101)
fit.rf <- train(Species~., data=iris_train, method='ranger',
                  trControl=control, metric=metric)
```

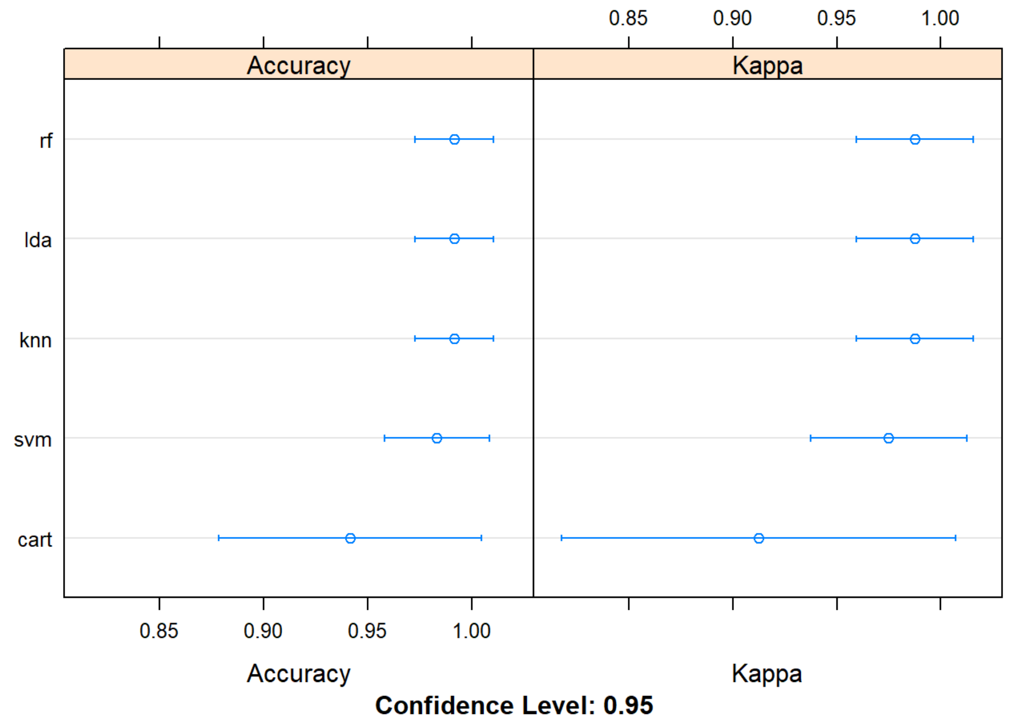
```
# Compare the results of these algorithms
iris.results <- resamples(list(lda=fit.lda, cart=fit.cart, knn=fit.knn, svm=fit.svm, rf=fit.rf))

# Table Comparison
summary(iris.results)
```

```
##
## Call:
## summary.resamples(object = iris.results)
##
## Models: lda, cart, knn, svm, rf
## Number of resamples: 10
##
## Accuracy
##           Min.   1st Qu. Median      Mean 3rd Qu. Max. NA's
## lda  0.9166667 1.0000000      1 0.9916667       1    1    0
## cart 0.7500000 0.9166667      1 0.9416667       1    1    0
## knn  0.9166667 1.0000000      1 0.9916667       1    1    0
## svm  0.9166667 1.0000000      1 0.9833333       1    1    0
## rf   0.9166667 1.0000000      1 0.9916667       1    1    0
##
## Kappa
##       Min. 1st Qu. Median   Mean 3rd Qu. Max. NA's
## lda  0.875   1.000      1 0.9875       1    1    0
## cart 0.625   0.875      1 0.9125       1    1    0
## knn  0.875   1.000      1 0.9875       1    1    0
## svm  0.875   1.000      1 0.9750       1    1    0
## rf   0.875   1.000      1 0.9875       1    1    0
```

```
# Let's plot the results of these algorithms:
bwplot(iris.results)
```



```
dotplot(iris.results)
```



Observation- Looking at the results and plots, we can say that mean accuracy of LDA and KNN model is better than other models. Lets use LDA model to make final predictions.

6. Make Prediction We will use the test data and LDA model to make final predictions.

a. Predict on test data

```
iris_prediction <- predict(fit.lda, iris_test)
confusionMatrix(iris_prediction, iris_test$Species)
```

```
## Confusion Matrix and Statistics
##
##                  Reference
## Prediction        Iris-setosa Iris-versicolor Iris-virginica
##   Iris-setosa              10               0              0
##   Iris-versicolor           0               8              0
##   Iris-virginica            0               2             10
##
## Overall Statistics
##
##                Accuracy : 0.9333
##                  95% CI : (0.7793, 0.9918)
##     No Information Rate : 0.3333
##     P-Value [Acc > NIR] : 8.747e-12
##
##                   Kappa : 0.9
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: Iris-setosa Class: Iris-versicolor
## Sensitivity                      1.0000                 0.8000
## Specificity                      1.0000                 1.0000
## Pos Pred Value                   1.0000                 1.0000
## Neg Pred Value                   1.0000                 0.9091
## Prevalence                       0.3333                 0.3333
## Detection Rate                   0.3333                 0.2667
## Detection Prevalence             0.3333                 0.2667
## Balanced Accuracy                1.0000                 0.9000
##                      Class: Iris-virginica
## Sensitivity                         1.0000
## Specificity                         0.9000
## Pos Pred Value                      0.8333
## Neg Pred Value                      1.0000
## Prevalence                          0.3333
## Detection Rate                      0.3333
## Detection Prevalence                0.4000
## Balanced Accuracy                   0.9500
```

We can see that the accuracy is 100%. It was a small validation dataset, but this result is within our expected margin of 97% +/-4% suggesting we may have an accurate and a reliably accurate model.