INTRODUCTION

Iris dataset is a multivariate dataset introduced by British statistician and biologist Ronald Fisher in his 1936 paper. It includes three iris species (Iris setosa, Iris Virginica, Iris versicolor) with 50 samples each as well as some properties about each flower. One flower species is linearly separable from the other two, but the other two are not linearly separable from each other.

Attribute Information is given below:

1. Sepal length in cm
2. Sepal width in cm
3. Petal length in cm
4. Petal width in cm
5. Class: (Iris Setosa , Iris Virginica, Iris Versicolor)

OBJECTIVE

Download Iris dataset from the UCI repository and compare the results of at least two data analytics techniques. Here Decision tree ,Random Forest and Naïve Bayes techniques of Classification are used for comparison.

IMPORTING THE DATASET

```
iris<- read.csv("c:/iris/iris.csv")   #loading data
```

PREVIEW OF DATASET

I/P:

```
View(iris)          #view dataset
str(iris)           #view structure of dataset
summary(iris)       #view statistical summary of dataset
head(iris)          #view top 6 rows of dataset
```

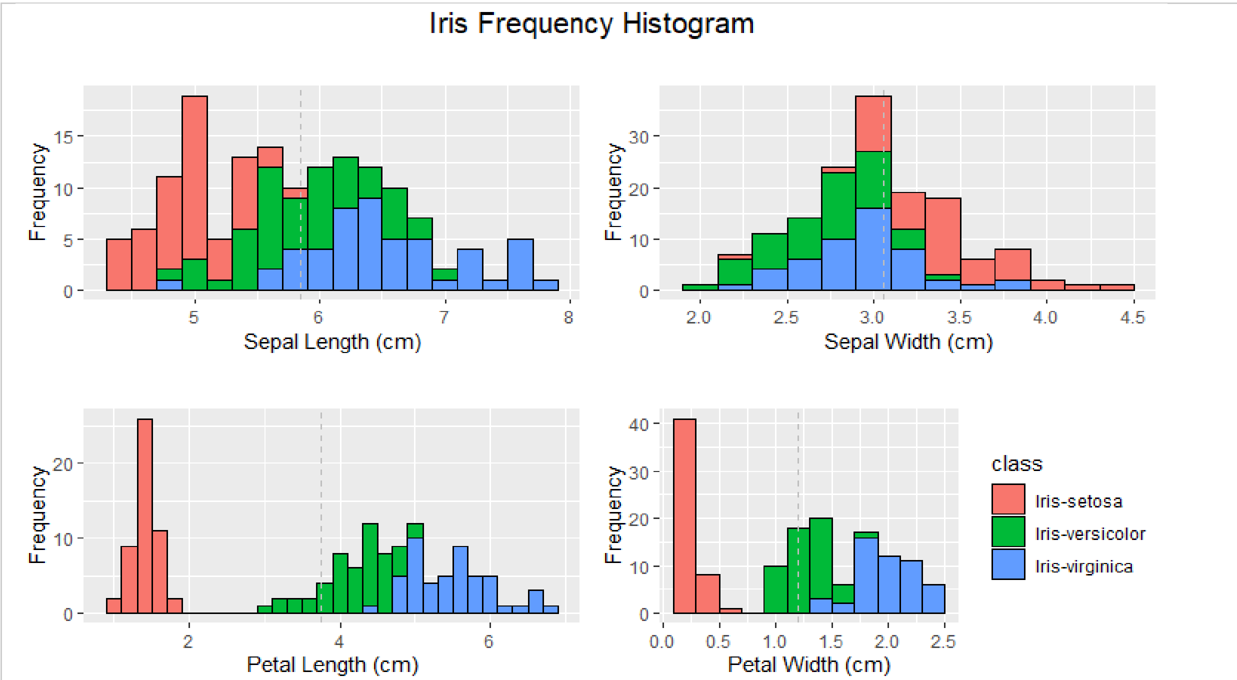
O/P:

```
> str(iris)
'data.frame':   150 obs. of  5 variables:
 $ sepal.length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
 $ sepal.width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
 $ petal.length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
 $ petal.width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
 $ class       : chr  "Iris-setosa" "Iris-setosa" "Iris-setosa" "Iris-setosa" ...
> summary(iris)
  sepal.length    sepal.width     petal.length    petal.width       class          
 Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100   Length:150        
 1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300   Class :character  
 Median :5.800   Median :3.000   Median :4.350   Median :1.300   Mode  :character  
 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                     
> head(iris)
  sepal.length sepal.width petal.length petal.width       class
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
6          5.4         3.9          1.7         0.4 Iris-setosa
```

DATA VISUALIZATION

A visual representation of how the data points are distributed with respect to the frequency.

Analysis with the histogram:



- The distribution of Iris-Setosa petal is completely different from other 2 species
- The species can’t be separated from one another using sepal features since the distribution is overlapping.
- Petal length and petal width can be used as a factor to identify 3 species

DATA ANALYTICS TECHNIQUES

- **DECISION TREE**

Decision tree is a type of supervised learning algorithm mostly used for classification problem. This algorithm split the data into two or more homogeneous sets based on the most significant attributes making the group as distinct as possible.

To increase the adaptability of the model, the entire data is divided into "train_data" and "test_data" sets. ‘caTools’ package is used for sample.split()

I/P:

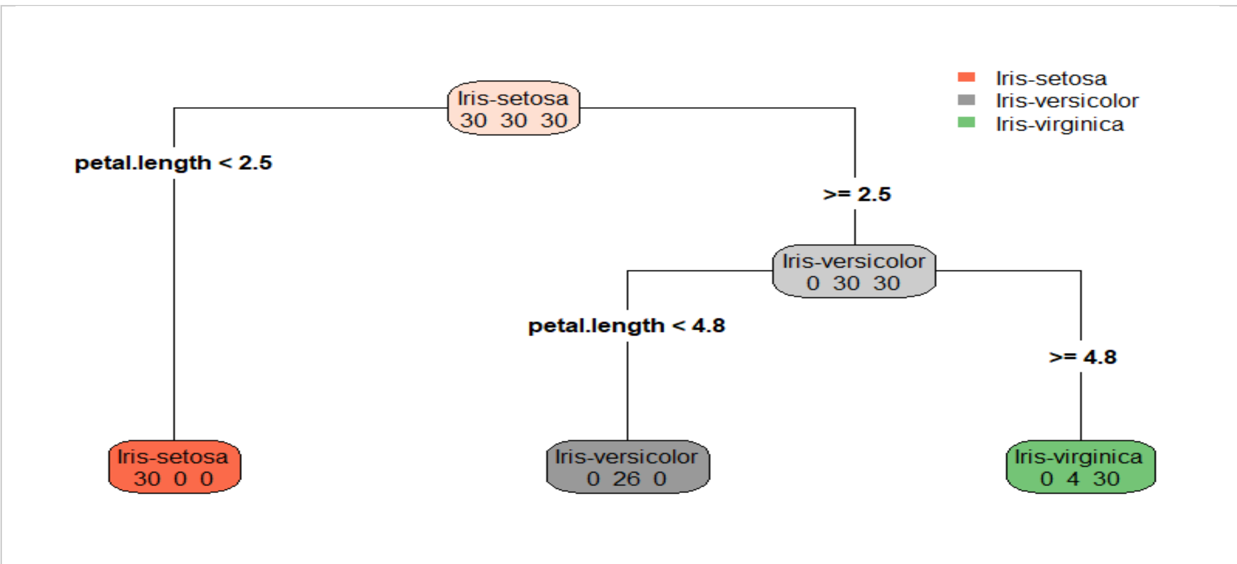
```
set.seed(123)                                          # Always generate same random numbers
sample_data = sample.split(iris, SplitRatio = 0.75)    # splits the data in the ratio mentioned in SplitRatio
train_data <- subset(iris, sample_data == TRUE)        # a training dataset  which are marked as TRUE
test_data <- subset(iris, sample_data == FALSE)        # a testing dataset  which are marked as FALSE
```

In R, rpart is for modeling decision trees and rpart.plot package enables the plotting of a tree. To predict which factors such as sepal length, sepal width , petal length, petal width determine the species of iris flower.

I/P:

```
fit<- rpart(class~ sepal.length +sepal.width + petal.length + petal.width, method = "class",
          data = train_data,
          control = rpart.control(cp = 0),
          parms = list(split="information"))
rpart.plot(fit,type= 4 , extra=1)
```

O/P:



Checking the accuracy using a confusion matrix by comparing predictions to actual classifications. 'caret' package is used for confusion matrix.

I/P:

```
iris_pred <- predict(object = fit,
                 newdata = test_data,
                 type = "class")                    #test data is used for prediction

confusionMatrix(data = as.factor(iris_pred),
             reference = as.factor(test_data$class))
```

O/P:

```
> confusionMatrix(data = as.factor(iris_pred),
+                 reference = as.factor(test_data$class))
Confusion Matrix and Statistics

                 Reference
Prediction        Iris-setosa Iris-versicolor Iris-virginica
  Iris-setosa              20               0              0
  Iris-versicolor           0              18              1
  Iris-virginica            0               2             19

Overall Statistics

               Accuracy : 0.95
                 95% CI : (0.8608, 0.9896)
    No Information Rate : 0.3333
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.925

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: Iris-setosa Class: Iris-versicolor Class: Iris-virginica
Sensitivity                      1.0000                 0.9000                0.9500
Specificity                      1.0000                 0.9750                0.9500
Pos Pred Value                   1.0000                 0.9474                0.9048
Neg Pred Value                   1.0000                 0.9512                0.9744
Prevalence                       0.3333                 0.3333                0.3333
Detection Rate                   0.3333                 0.3000                0.3167
Detection Prevalence             0.3333                 0.3167                0.3500
Balanced Accuracy                1.0000                 0.9375                0.9500
```

ACCURACY

In the above result accuracy is 0.95

i.e., our model has achieved 95% accuracy!

- **RANDOM FOREST**

Verifying performance using ‘randomForest’ package.

I/P:

```
iris_class <- factor(iris$class,
            levels = c ('Iris-setosa', 'Iris-versicolor', 'Iris-virginica'),
            labels = c (1, 2, 3))                                    #character to numeric
iris_random<- randomForest(iris_class~ sepal.length + sepal.width + petal.length + petal.width,
      data = iris)
print(iris_random)
print (importance(iris_random,type = 2))
```

O/P:

```
> print(iris_random)

Call:
 randomForest(formula = iris_class ~ sepal.length + sepal.width +      petal.length + petal.width,
 data = iris)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 2

        OOB estimate of  error rate: 4%
Confusion matrix:
   1  2  3 class.error
1 50  0  0        0.00
2  0 47  3        0.06
3  0  3 47        0.06
> print (importance(iris_random,type = 2))
             MeanDecreaseGini
sepal.length         9.589300
sepal.width          2.360683
petal.length        43.775235
petal.width         43.493235
```

GINI is a measure of node impurity. From the above details it is clear that Petal features are more important compared to sepal features since the values are too small for sepal features (9.59 and 2.36) and the error rate is 4%. So, we can eliminate sepal feature and check the accuracy again.

I/P:

```
iris_random1<- randomForest(iris_class~ petal.length + petal.width, data = iris )
print(iris_random1)
print(importance(iris_random1,type = 2))
```

O/P:

```
> print(iris_random1)

Call:
 randomForest(formula = iris_class ~ petal.length + petal.width,      data = iris)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 1

        OOB estimate of  error rate: 3.33%
Confusion matrix:
   1  2  3 class.error
1 50  0  0        0.00
2  0 47  3        0.06
3  0  2 48        0.04
> print(importance(iris_random1,type = 2))
             MeanDecreaseGini
petal.length         48.24790
petal.width          48.91101
```

In above table the error rate is 3.33%.

Checking the accuracy using a confusion matrix by comparing predictions to actual classifications. ‘caret’ package is used for confusion matrix.

```
iris_dataset<- iris                                  #copying the dataset to another variable
sample = sample.split(iris_dataset, SplitRatio = 0.75)   #splitting data for matching levels
train <- subset(iris_dataset, sample_data == TRUE)
test <- subset(iris_dataset, sample_data == FALSE)
iris_pred_rand <- predict(object = iris_random1,
             newdata = test, type = "class")
confusionMatrix(data = as.factor(iris_pred_rand),  reference = as.factor(test$class))
```

O/P:

```
> confusionMatrix(data = as.factor(iris_pred_rand),
+                 reference = as.factor(test$class))
Confusion Matrix and Statistics

          Reference
Prediction  1  2  3
         1 20  0  0
         2  0 19  0
         3  0  1 20

Overall Statistics

               Accuracy : 0.9833
                 95% CI : (0.9106, 0.9996)
    No Information Rate : 0.3333
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.975

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: 1 Class: 2 Class: 3
Sensitivity            1.0000   0.9500   1.0000
Specificity            1.0000   1.0000   0.9750
Pos Pred Value         1.0000   1.0000   0.9524
Neg Pred Value         1.0000   0.9756   1.0000
Prevalence             0.3333   0.3333   0.3333
Detection Rate         0.3333   0.3167   0.3333
Detection Prevalence   0.3333   0.3167   0.3500
Balanced Accuracy      1.0000   0.9750   0.9875
```

ACCURACY

In above result, the accuracy is 0.9833
So, the accuracy for this model is (0.9833 * 100)% =98.33%

- ## NAIVE BAYES MODEL

Naive Bayes is a classification technique based on Bayes' Theorem with an assumption of independence among predictors. For Naïve Bayes model , 'e1071' package is used.

I/P:

```
classifier_cl <- naiveBayes(class ~ ., data = train_data)
y_pred <- predict(classifier_cl, newdata = test_data)          #predicting on test data
cm <- table(test_data$class, y_pred)                           #for confusion matrix
confusionMatrix(cm)                                            #model evaluation
```

O/P:

```
> confusionMatrix(cm)
Confusion Matrix and Statistics

                 y_pred
                  Iris-setosa Iris-versicolor Iris-virginica
  Iris-setosa              20               0              0
  Iris-versicolor           0              15              5
  Iris-virginica            0               1             19

Overall Statistics

               Accuracy : 0.9
                 95% CI : (0.7949, 0.9624)
    No Information Rate : 0.4
    P-Value [Acc > NIR] : 8.166e-16

                  Kappa : 0.85

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: Iris-setosa Class: Iris-versicolor Class: Iris-virginica
Sensitivity                      1.0000                 0.9375                0.7917
Specificity                      1.0000                 0.8864                0.9722
Pos Pred Value                   1.0000                 0.7500                0.9500
Neg Pred Value                   1.0000                 0.9750                0.8750
Prevalence                       0.3333                 0.2667                0.4000
Detection Rate                   0.3333                 0.2500                0.3167
Detection Prevalence             0.3333                 0.3333                0.3333
Balanced Accuracy                1.0000                 0.9119                0.8819
```

ACCURACY

In above result the accuracy is 0.9
Accuracy of model is 0.9 *100 = 90%

INFERENCE

Accuracy:

Decision tree - 95%
Random Forest - 98.33%
Naïve Bayes - 90%

From above result it is evident that Random Forest is more accurate!