Assignment

Yatharth Malik

March 27, 2017

Comparing classification techniques on iris dataset

Many classification techniques are applied on iris dataset to compare the performance of each algorithm on testing dataset. Training dataset contains 100 rows which is used to train the model. Testing dataset contains 50 rows which is used to calculate the performance of the model on unseen dataset.

Loading dataset and libraries

```
library(ggplot2)
library(rpart)
library(rpart.plot)
library(gmodels)
library(e1071)
library(gridExtra)
library(randomForest)
data(iris)
summary(iris)
```

```
##   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   :3.057   Mean   :3.758   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
##        Species
##  setosa    :50
##  versicolor:50
##  virginica :50
##
##
##
```

Data Pre-processing

We will normalize the continuous variables before performing any analysis on the dataset

```
temp = as.data.frame(scale(iris[,1:4]))
temp$Species = iris$Species
summary(temp)
```

```
##   Sepal.Length       Sepal.Width       Petal.Length      Petal.Width
##  Min.   :-1.86378   Min.   :-2.4258   Min.   :-1.5623   Min.   :-1.4422
##  1st Qu.:-0.89767   1st Qu.:-0.5904   1st Qu.:-1.2225   1st Qu.:-1.1799
##  Median :-0.05233   Median :-0.1315   Median : 0.3354   Median : 0.1321
##  Mean   : 0.00000   Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000
##  3rd Qu.: 0.67225   3rd Qu.: 0.5567   3rd Qu.: 0.7602   3rd Qu.: 0.7880
##  Max.   : 2.48370   Max.   : 3.0805   Max.   : 1.7799   Max.   : 1.7064
##        Species
##  setosa    :50
##  versicolor:50
##  virginica :50
##
##
##
```

Exploratory data analysis

We will look at couple of plots, to capture the dependence of variables with each other.

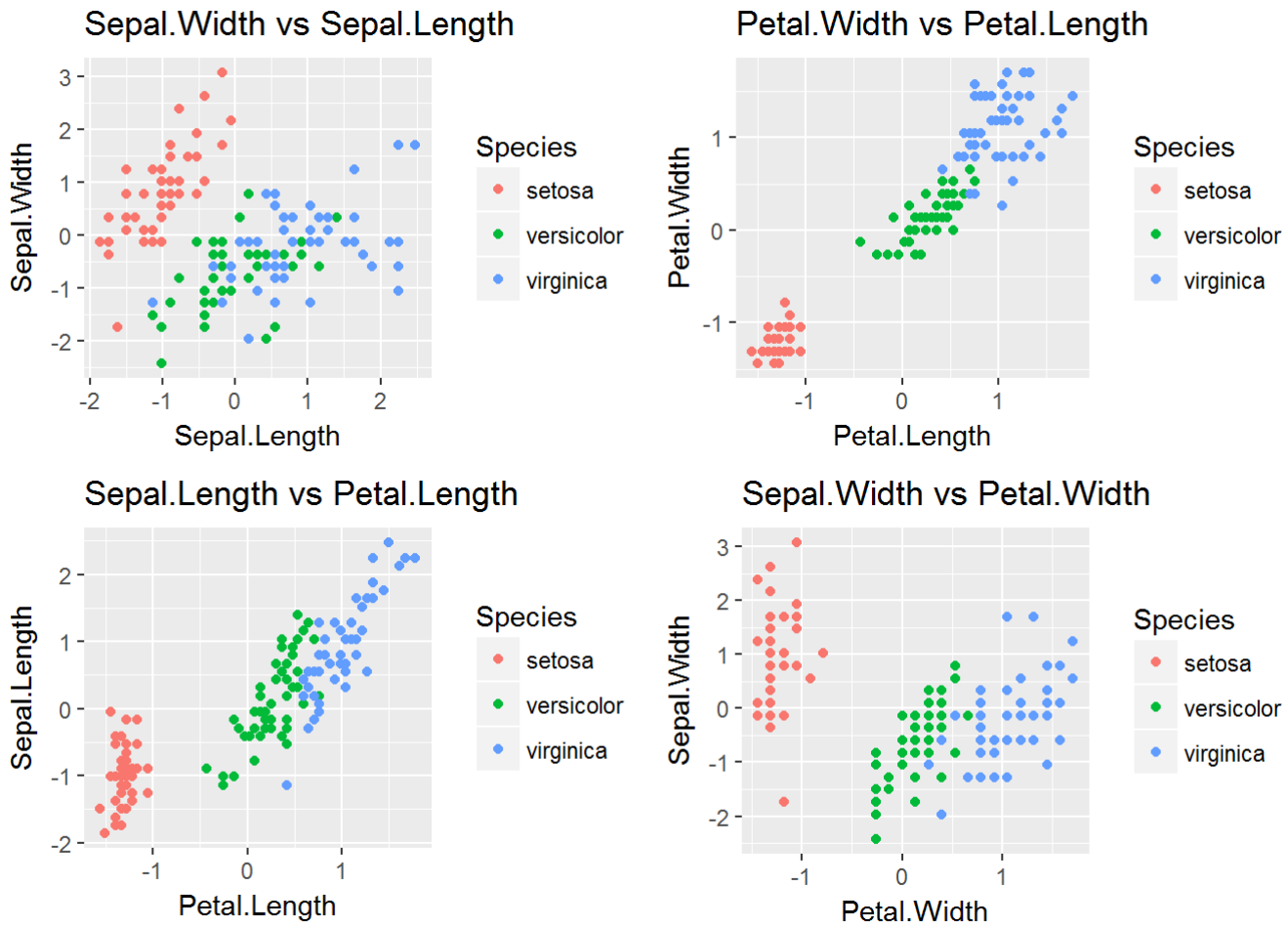
```
g1 = ggplot(temp,aes(x =Sepal.Length,y = Sepal.Width,color = Species)) + geom_point() +
ggtitle("Sepal.Width vs Sepal.Length")

g2 = ggplot(temp,aes(x =Petal.Length,y = Petal.Width,color = Species)) + geom_point() +
ggtitle("Petal.Width vs Petal.Length")

g3 = ggplot(temp,aes(x =Petal.Length,y = Sepal.Length,color = Species)) + geom_point() +
ggtitle("Sepal.Length vs Petal.Length")

g4 = ggplot(temp,aes(x =Petal.Width,y = Sepal.Width,color = Species)) + geom_point()  +
ggtitle("Sepal.Width vs Petal.Width")

grid.arrange(g1,g2,g3,g4,nrow = 2)
```



Creating training and testing dataset

```
smp_size =  100
set.seed(123)
train_ind = sample(seq_len(nrow(temp)), size = smp_size)
train = temp[train_ind, ]
test = temp[-train_ind, ]
```

Number of rows in “train”

```
nrow(train)
```

```
## [1] 100
```

Number of rows in “test”

```
nrow(test)
```

```
## [1] 50
```

Species distribution in “train”

```
table(train$Species)
```

```
##
##     setosa versicolor  virginica
##         36         31         33
```

Species distribution in “test”

```
table(test$Species)
```

```
##
##     setosa versicolor  virginica
##         14         19         17
```

Classification Techniques

1. Decision Trees

```
model.rpart = rpart(Species ~ . ,data =train)
preds.rpart = predict(model.rpart,newdata = test,type = "class")
CrossTable(test$Species,preds.rpart,chisq = F,prop.r = F,prop.c = F,prop.t = F,prop.chisq = F)
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |-------------------------|
##
##
## Total Observations in Table:  50
##
##
##              | preds.rpart
## test$Species |     setosa | versicolor |  virginica |  Row Total |
## -------------|------------|------------|------------|------------|
##       setosa |         14 |          0 |          0 |         14 |
## -------------|------------|------------|------------|------------|
##   versicolor |          0 |         16 |          3 |         19 |
## -------------|------------|------------|------------|------------|
##    virginica |          0 |          1 |         16 |         17 |
## -------------|------------|------------|------------|------------|
## Column Total |         14 |         17 |         19 |         50 |
## -------------|------------|------------|------------|------------|
##
##
```

Accuracy of Decision trees

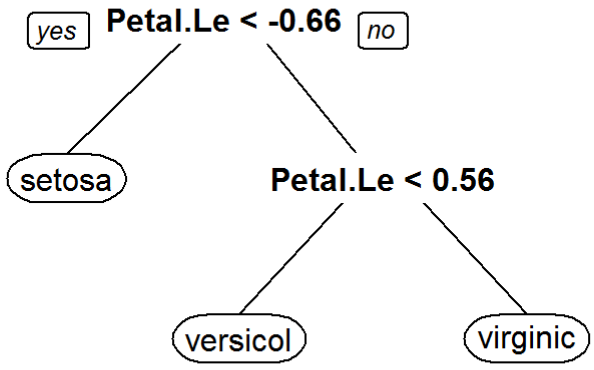
```
((14+16+16)/nrow(test))*100
```

```
## [1] 92
```

Explaination

Decision trees are supervised classification algorithm useful when input variables interact with the output in “if-then” kinds of ways. They are also suitable when inputs have an AND relationship to each other or when input variables are redundant or correlated.

By observing the plots from “Exploratory Data Analysis”, we can clearly see a positive relationship/correlation between the variables of Iris dataset. Thus making decision trees ideal for the classification of the species. Also the “if-then” relation between the variables of Iris dataset can be seen from the below plot.



2. k-Nearest Neighbours

```
library(class)
cl = train$Species
set.seed(1234)
preds.knn = knn(train[,1:4],test[,1:4],cl,k=3)
CrossTable(preds.knn,test$Species,chisq = F,prop.r = F,prop.c = F,prop.t = F,prop.chisq = F)
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |-------------------------|
##
##
## Total Observations in Table:  50
##
##
##              | test$Species
##    preds.knn |     setosa | versicolor |  virginica |  Row Total |
## -------------|------------|------------|------------|------------|
##       setosa |         14 |          0 |          0 |         14 |
## -------------|------------|------------|------------|------------|
##   versicolor |          0 |         17 |          2 |         19 |
## -------------|------------|------------|------------|------------|
##    virginica |          0 |          2 |         15 |         17 |
## -------------|------------|------------|------------|------------|
## Column Total |         14 |         19 |         17 |         50 |
## -------------|------------|------------|------------|------------|
##
##
```

Accuracy of kNN

```
((14+17+15)/nrow(test))*100
```

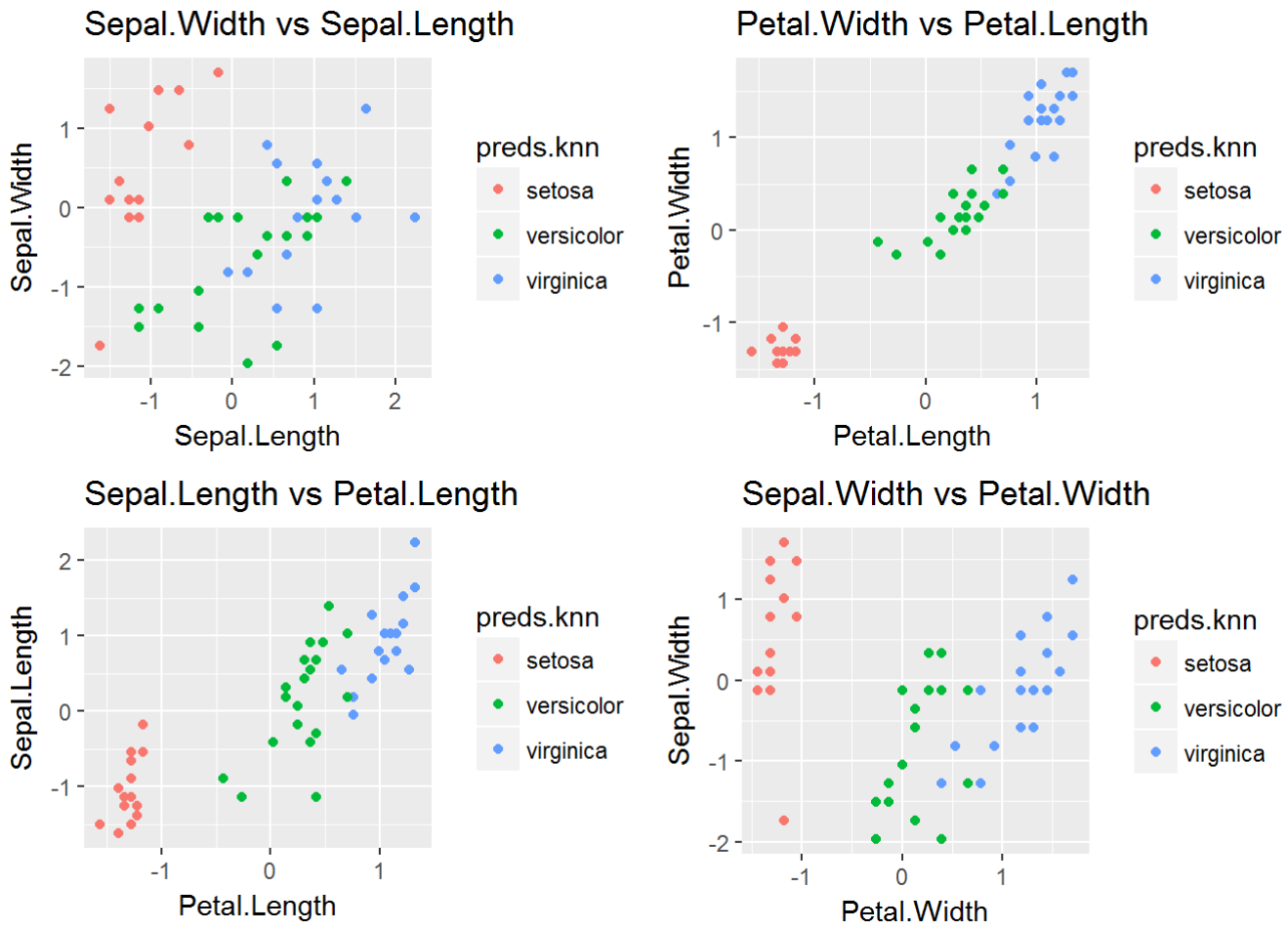
```
## [1] 92
```

Explaination

kNN can be used for both classification and regression problem. kNN considers the most similar other items defined in terms of their attributes, look at their labels, and give the unassigned item the majority vote.

By looking at the plots we can clearly see the grouping of species based on their charachterstics such as Sepal.Length, Sepal.Width, etc. When a new data point is introduced, its similarity (using euclidean distance in this case as all variables are continuous) is measured from each of the grouping and species of the test data point is assigned according to the nearest (distance-wise) grouping. Hence, kNN can be easily used for classification of testing data points where we can easily identify the clusters of training data points. Thus, making kNN suitable for Iris dataset.

Below plots show the classification of test data points based on the distance of the test data points from the training groups(clusters).



3. Support Vector Machine(SVM)

```
model.svm = svm(Species ~ . ,data = train)
preds.svm = predict(model.svm,newdata = test)
CrossTable(preds.svm,test$Species,chisq = F,prop.r = F,prop.c = F,prop.t = F,prop.chisq = F)
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |-------------------------|
##
##
## Total Observations in Table:  50
##
##
##              | test$Species
##    preds.svm |     setosa | versicolor |  virginica |  Row Total |
## -------------|------------|------------|------------|------------|
##       setosa |         14 |          0 |          0 |         14 |
## -------------|------------|------------|------------|------------|
##   versicolor |          0 |         16 |          2 |         18 |
## -------------|------------|------------|------------|------------|
##    virginica |          0 |          3 |         15 |         18 |
## -------------|------------|------------|------------|------------|
## Column Total |         14 |         19 |         17 |         50 |
## -------------|------------|------------|------------|------------|
##
##
```

Accuracy of SVM

```
((14+16+15)/nrow(test))*100
```

```
## [1] 90
```

Explaination

Support vector machines (SVMs) are useful when there are very many input variables or when input variables interact with the outcome or with each other in complicated (nonlinear) ways. By observing the plots we can clearly see that some variables are non-linearly related to each other. Hence, using SVM is a good option on the Iris dataset.

Since in SVM we plot each data item as a point in n-dimensional space (where n is number of features you have) with the value of each feature being the value of a particular coordinate and then find a line that splits the data between two differently classified groups of data such that the distances from the closest point in each of the two groups will be farthest away from this line drawn.Since our data is linearly seperable, SVM would be a good choice for classification purpose of Iris dataset.

4.Random Forest

```
set.seed(100)
model.rf = randomForest(Species ~ .,data = train)
preds.rf = predict(model.rf,newdata = test)
CrossTable(preds.rf,test$Species,chisq = F,prop.r = F,prop.c = F,prop.t = F,prop.chisq = F)
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |-------------------------|
##
##
## Total Observations in Table:  50
##
##
##              | test$Species
##     preds.rf |     setosa | versicolor |  virginica |  Row Total |
## -------------|------------|------------|------------|------------|
##       setosa |         14 |          0 |          0 |         14 |
## -------------|------------|------------|------------|------------|
##   versicolor |          0 |         16 |          1 |         17 |
## -------------|------------|------------|------------|------------|
##    virginica |          0 |          3 |         16 |         19 |
## -------------|------------|------------|------------|------------|
## Column Total |         14 |         19 |         17 |         50 |
## -------------|------------|------------|------------|------------|
##
##
```

Accuracy of Decision trees

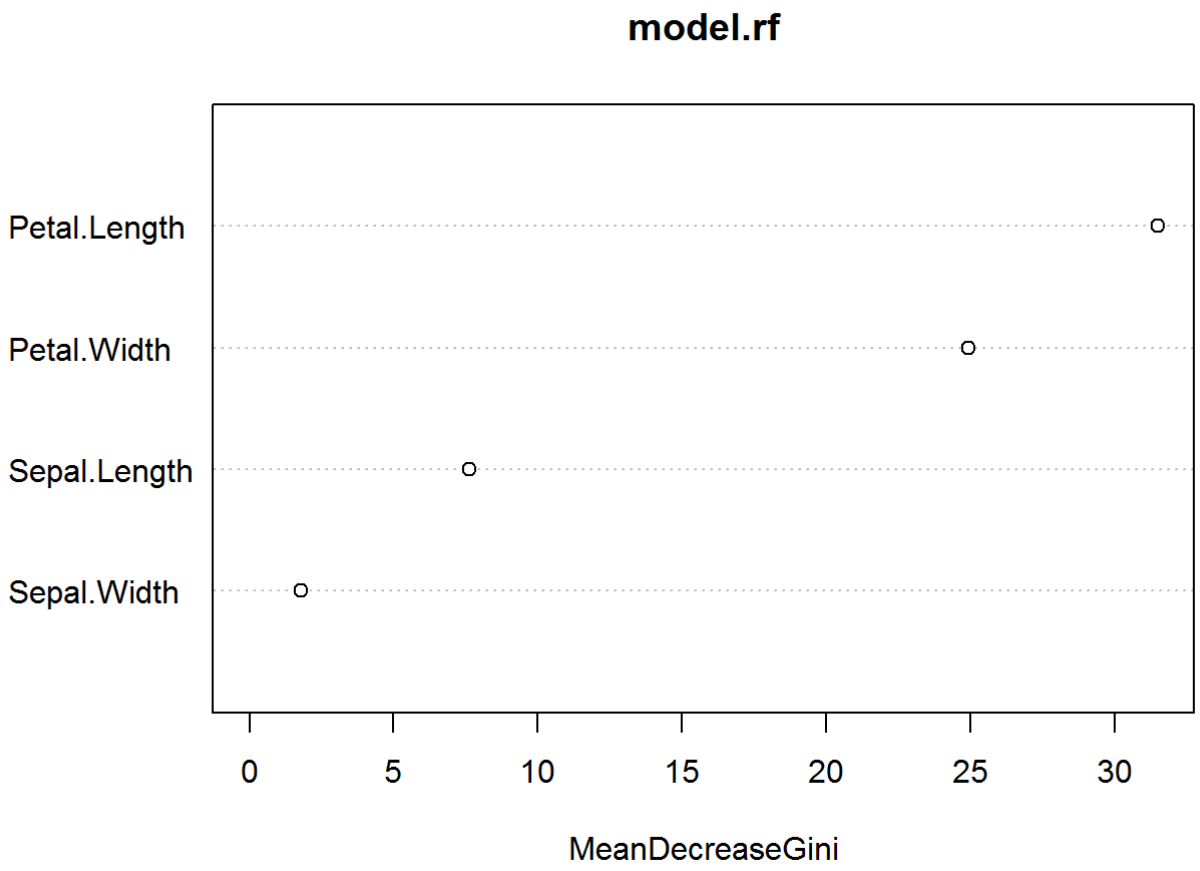
```
((14+16+16)/nrow(test))*100
```

```
## [1] 92
```

Explaination

Random forest is like bootstrapping algorithm with Decision tree (CART) model.Random forest builds multiple CART model with different sample and different initial variables.It repeats the process multiple times and then make final prediction on each observation.Final prediction is function of each prediction.

Random forest can be used in almost all cases and is frequently used to attain higher accuracy of model and to see importance of variables.Importance plot for variables of Iris data is shown below.



Petal.Length is the most important factor in classification of species of the flower.

Result comparison

We will now the compare the results of different models on iris dataset by looking at the predicted values that differ for each model.

Decision Tree vs kNN

```
which(preds.rpart != preds.knn)
```

```
## [1] 26 42
```

Decision Tree vs SVM

```
which(preds.rpart != preds.svm)
```

```
## [1] 42
```

Decision Tree vs Random Forest

```
which(preds.rpart != preds.rf)
```

```
## integer(0)
```

Both Random Forest and Decision trees gave us same prediction results.

kNN vs SVM

```
which(preds.knn != preds.svm)
```

```
## [1] 26
```

kNN vs Random Forest

```
which(preds.knn != preds.rf)
```

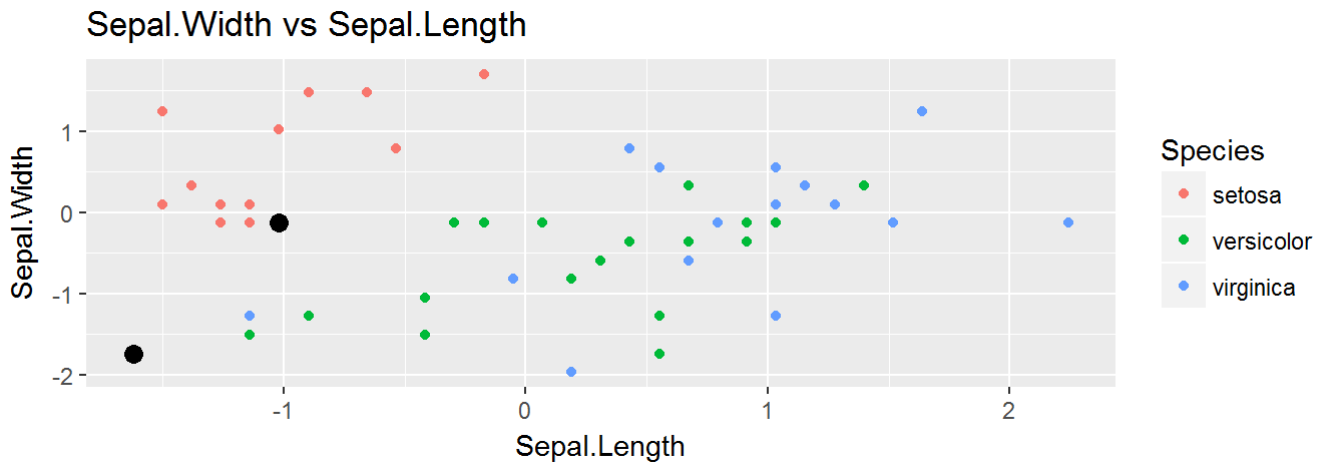
```
## [1] 26 42
```

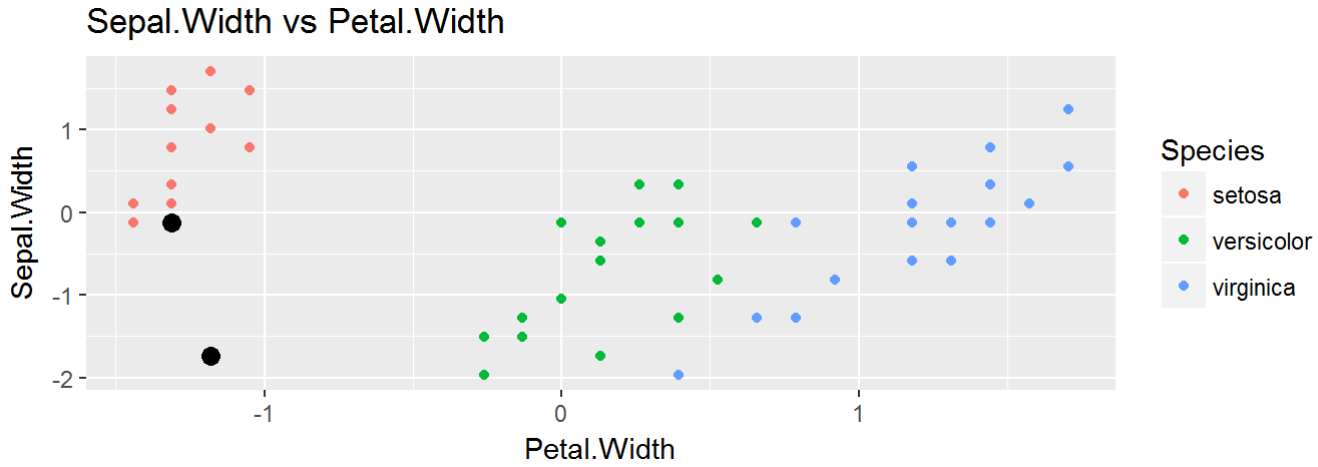
SVM vs Random Forest

```
which(preds.svm != preds.rf)
```

```
## [1] 42
```

Since the 26th and 42nd observation of testing dataset are classified wrongly in most of the cases, we will look at these outliers using the below plots.





Accuracy comparison

Comparison of the accuracy of different models on testing dataset.

```
models = data.frame(Technique = c("Decision Tree","kNN","SVM","Random Forest"),Accuracy_Percenta
ge = c(92,92,90,92))
models
```

```
##       Technique Accuracy_Percentage
## 1 Decision Tree                  92
## 2           kNN                  92
## 3           SVM                  90
## 4 Random Forest                  92
```

SVM performed poorer than other algorithms as the number of observations and variables in our dataset are small. Also not all variables of Iris data are non-linearly dependent.