Report_10657769_10651875

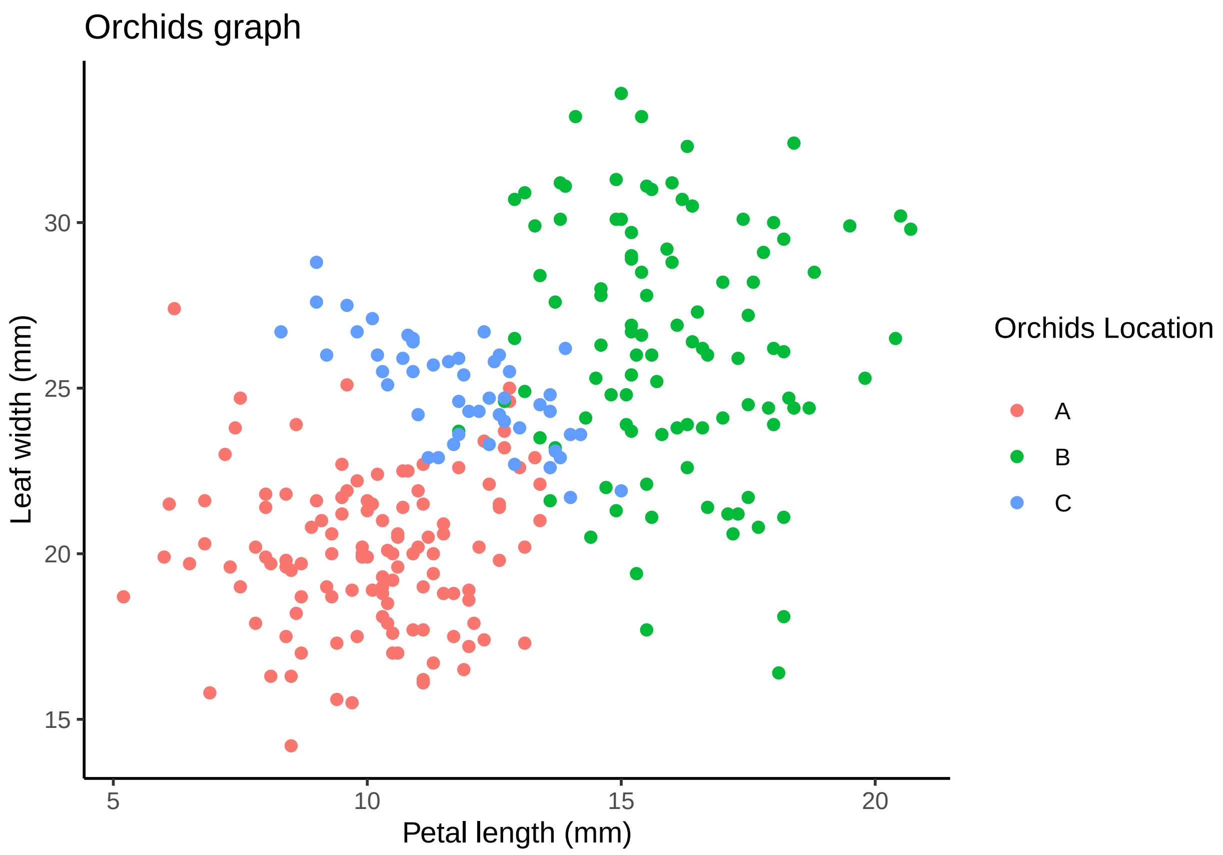
By: Aderibigbe Olugbenga & Paul Hazell

22/04/2020

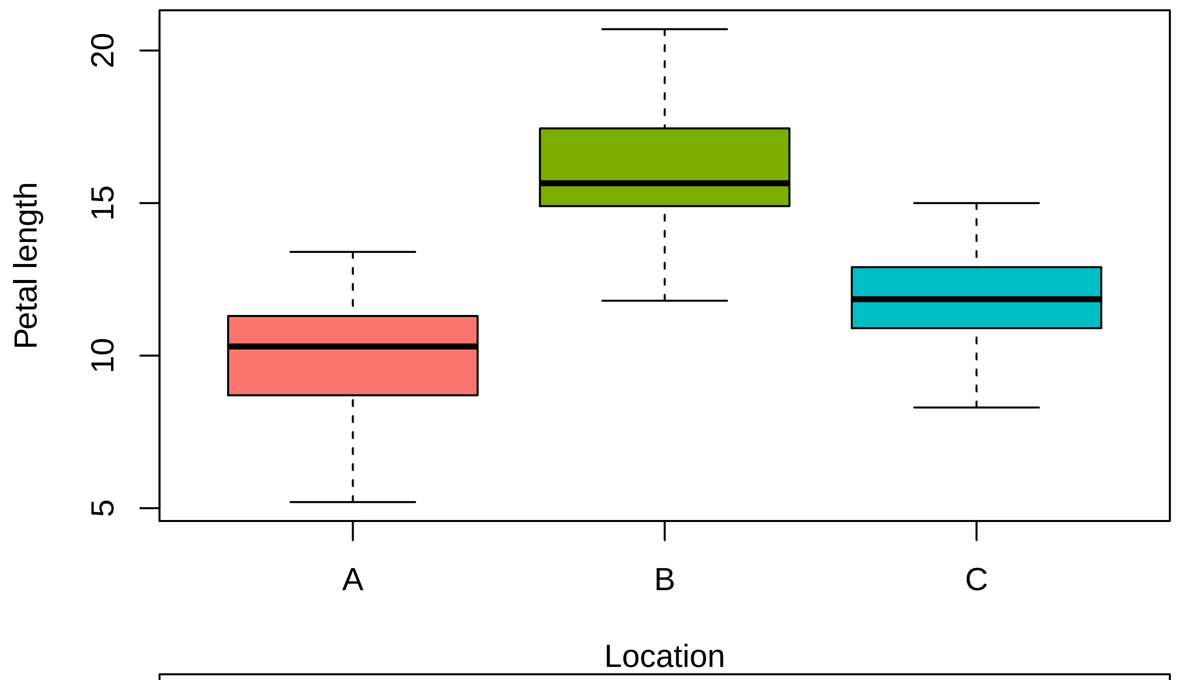
3.1 Machine Learning Task

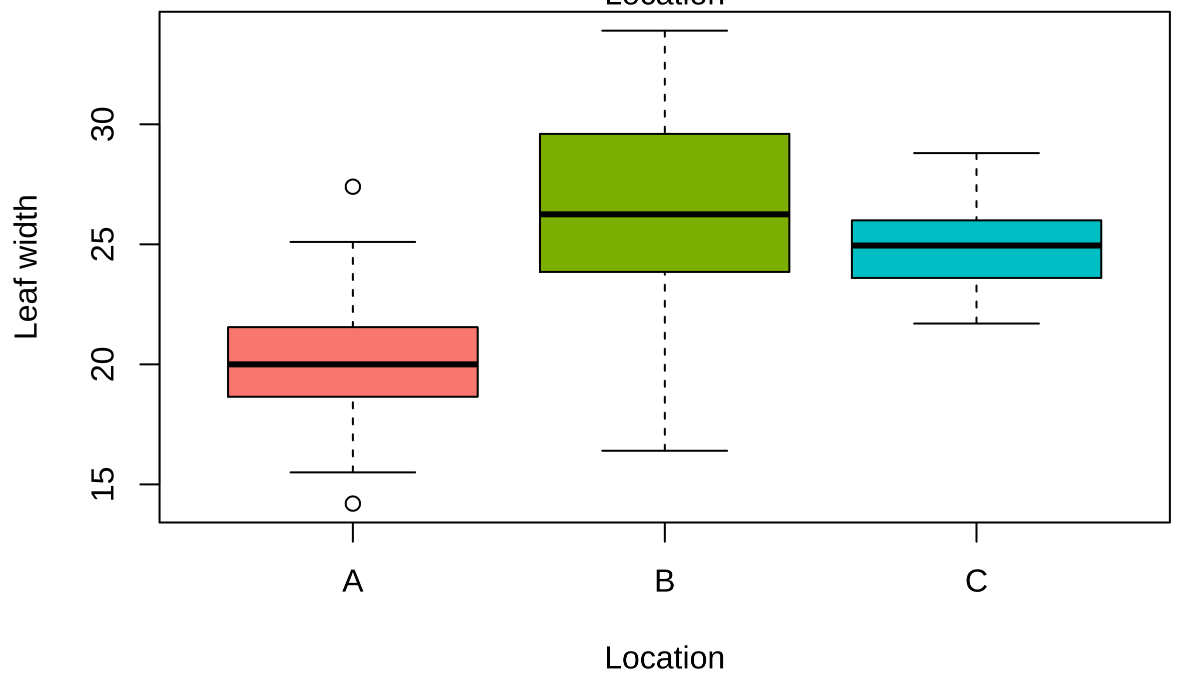
Part (a) Selecting parameters

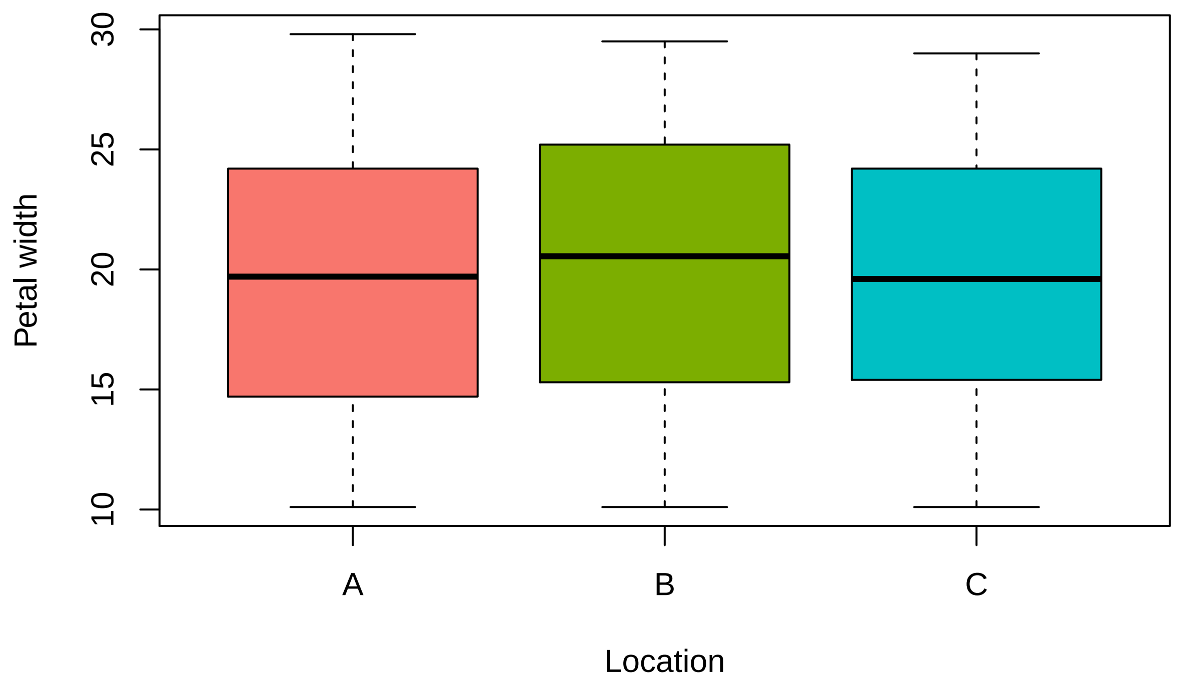
Graph of bivariate scatter plots to distinguish between the three locations of Orchids



Boxplots of the data to choose two characteristcs that should be used as predictors for orchids' locations.


Petal length
5
10
15
20
A
B
C
Location


Leaf width
15
20
25
30
A
B
C
Location



From the graph it can be seen that there is a considerable difference between the mean values of the Petal length (X1) and Leaf width (X2).

```
##   loc       X1
## 1   A 10.05917
## 2   B 16.01400
## 3   C 11.84800
```

```
##   loc       X2
## 1   A 20.01917
## 2   B 26.30500
## 3   C 24.95600
```

Whereas the difference between the mean Petal width (X3) with different locations is not much.

```
##   loc       X3
## 1   A 19.48333
## 2   B 20.04700
## 3   C 19.71400
```

So the Petal length and Leaf width data will be used as predictors for the orchids' locations.

Part (b) Training data

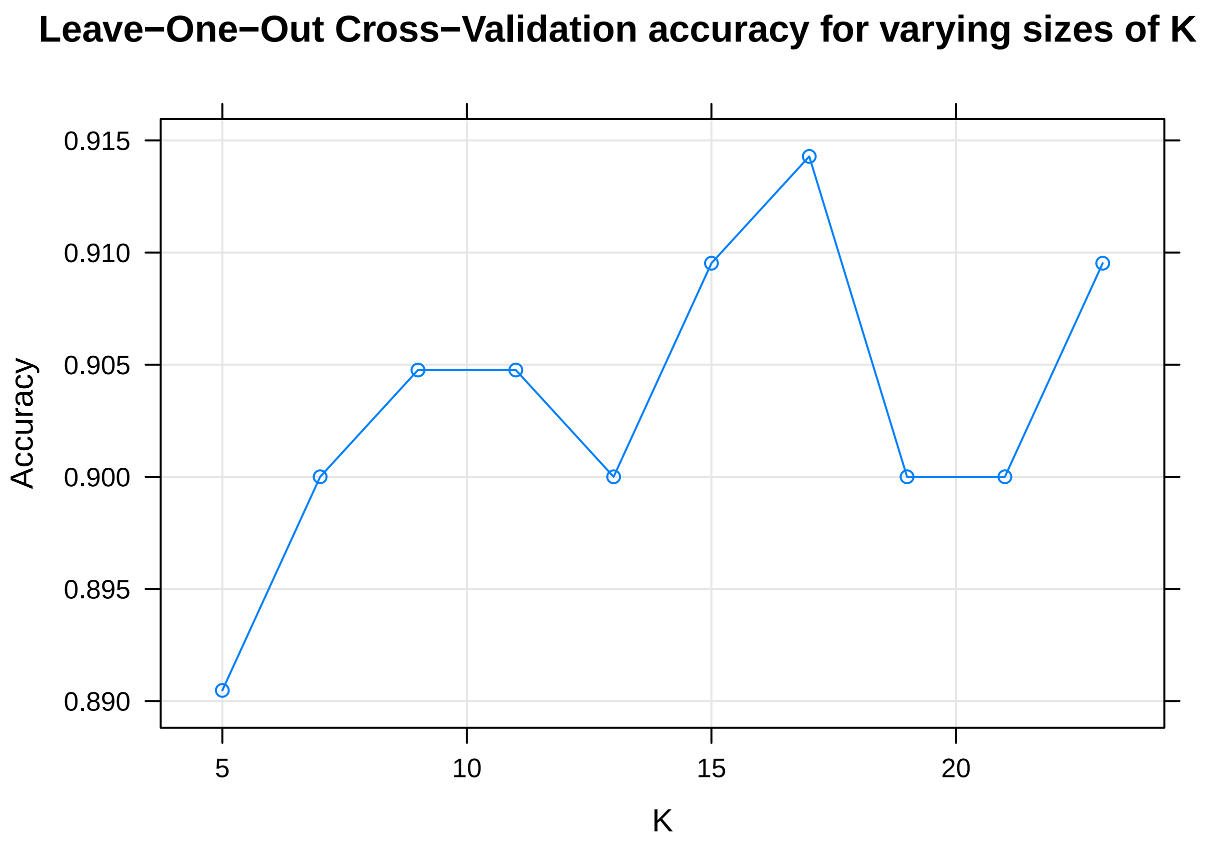
Creating a training set 210 randomly chosen data points and a test set of 60 data points.

```
set.seed(1)
data.subset <- sample(270, 210)
model.train <- orchid[data.subset,]
model.test <- orchid[-data.subset,]
```

Part (c) KNN Method

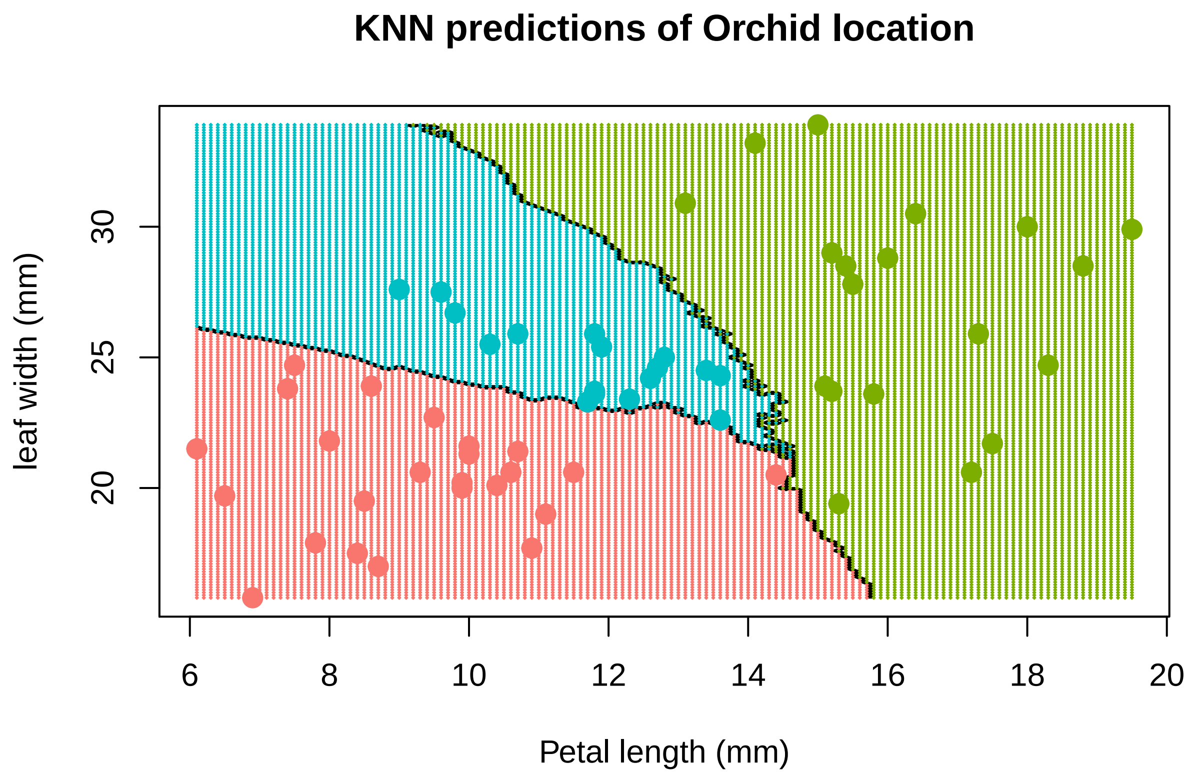
```r
# Create knn model
set.seed(1)
model.knn <- train(loc~.-X3,
                   data = model.train,
                   method = "knn",
                   trControl = trainControl(method  = "LOOCV"),
                   preProcess = c("center", "scale"), # Normalize the data
                   tuneLength = 10) # Number of possible K values to evaluate
```

Graph showing the accuracy of K



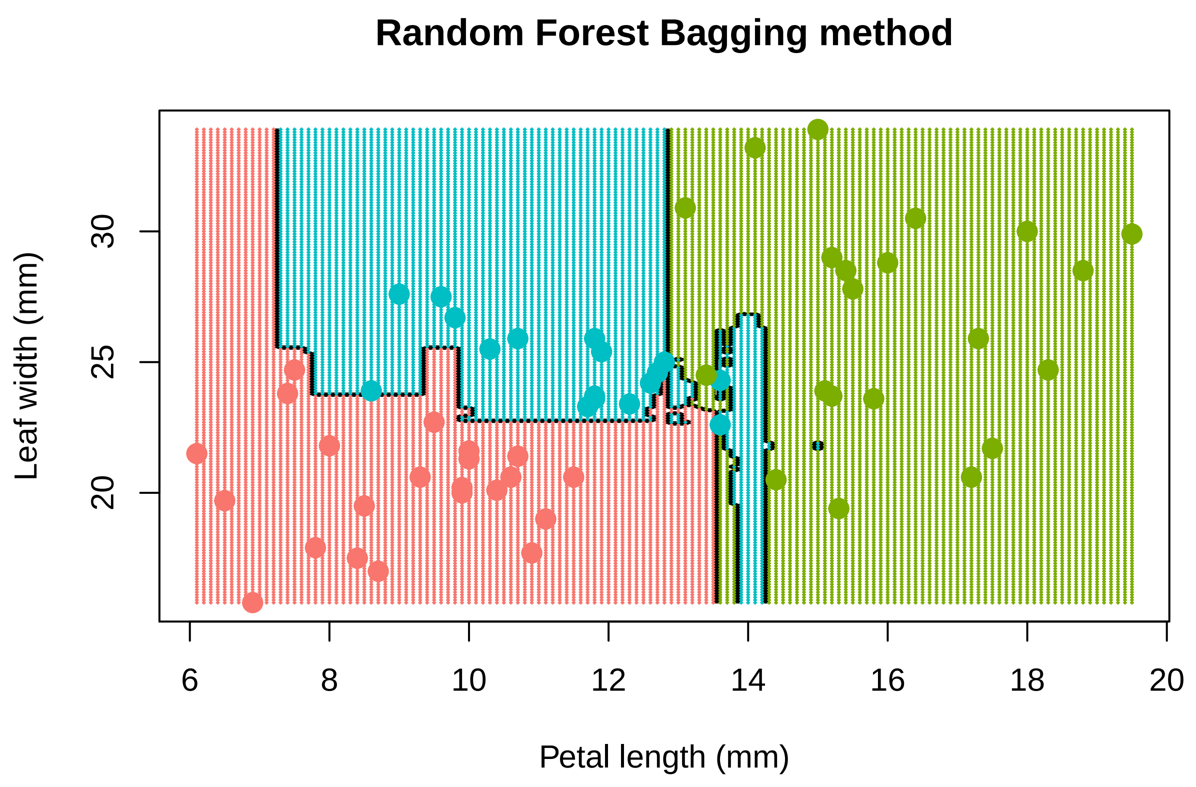
The best optimal k value is 17

```r
# Predicting the test model
predict.knn <- model.knn %>% predict(model.test)
```



Part (d) Random Forest Bagging method

```
##        A     B     C MeanDecreaseAccuracy MeanDecreaseGini
## X1 85.75 94.55 36.21               115.84            83.09
## X2 91.83 15.35 48.46               101.67            47.72
```



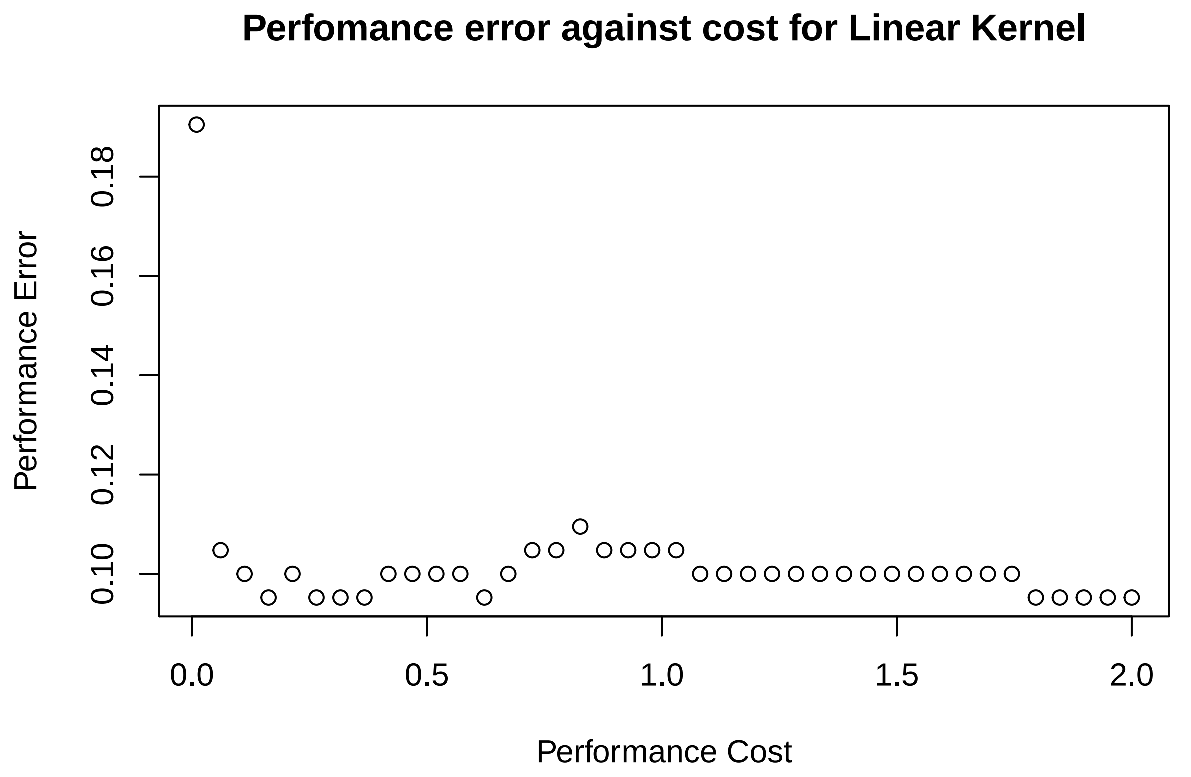
Part (e) Support Vector Machines

Linear kernel

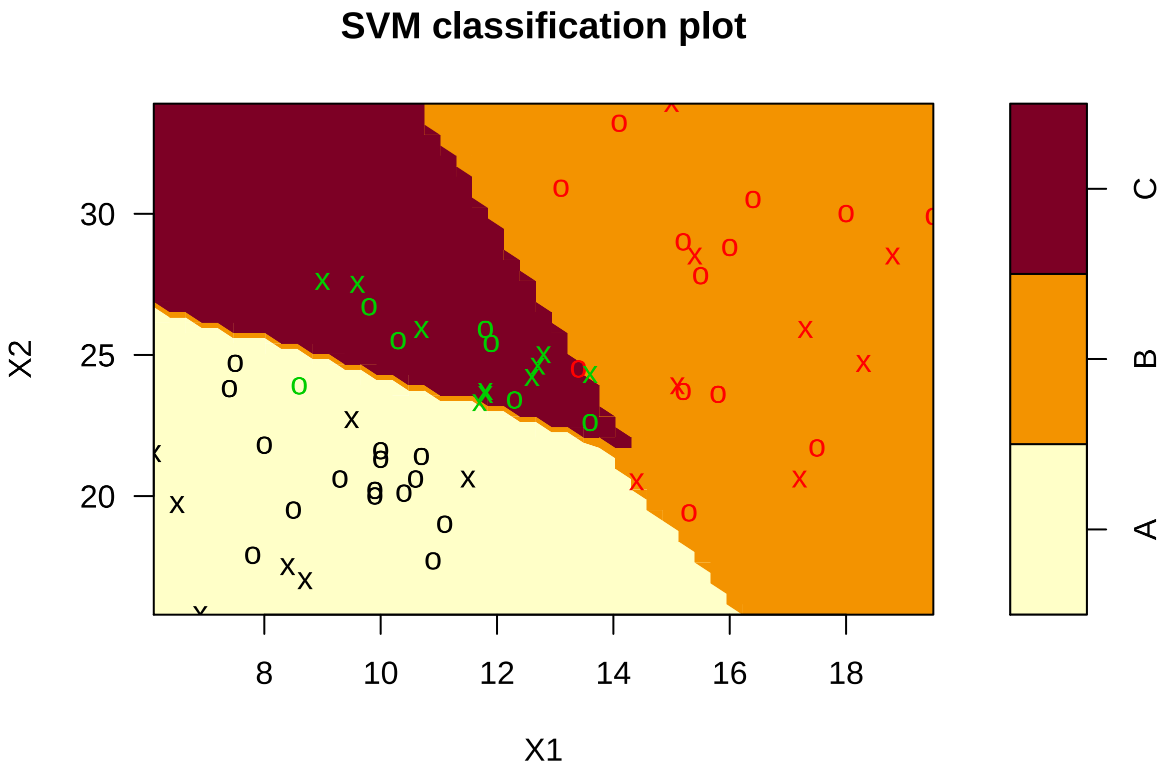
```
set.seed(1)
tune.out = tune(svm, loc ~ X1 + X2, data = orchid[data.subset,],
                kernel ="linear",
                ranges = list(cost = seq(from = 0.01,to = 2, length = 40) ))
```

Choosing the best cost parameter

```
plot(tune.out$performances$cost, tune.out$performances$error,
     main = "Perfomance error against cost for Linear Kernel",
     xlab = "Performance Cost",ylab = "Performance Error")
```



The best cost parameter for the linear kernel is 0.1630769



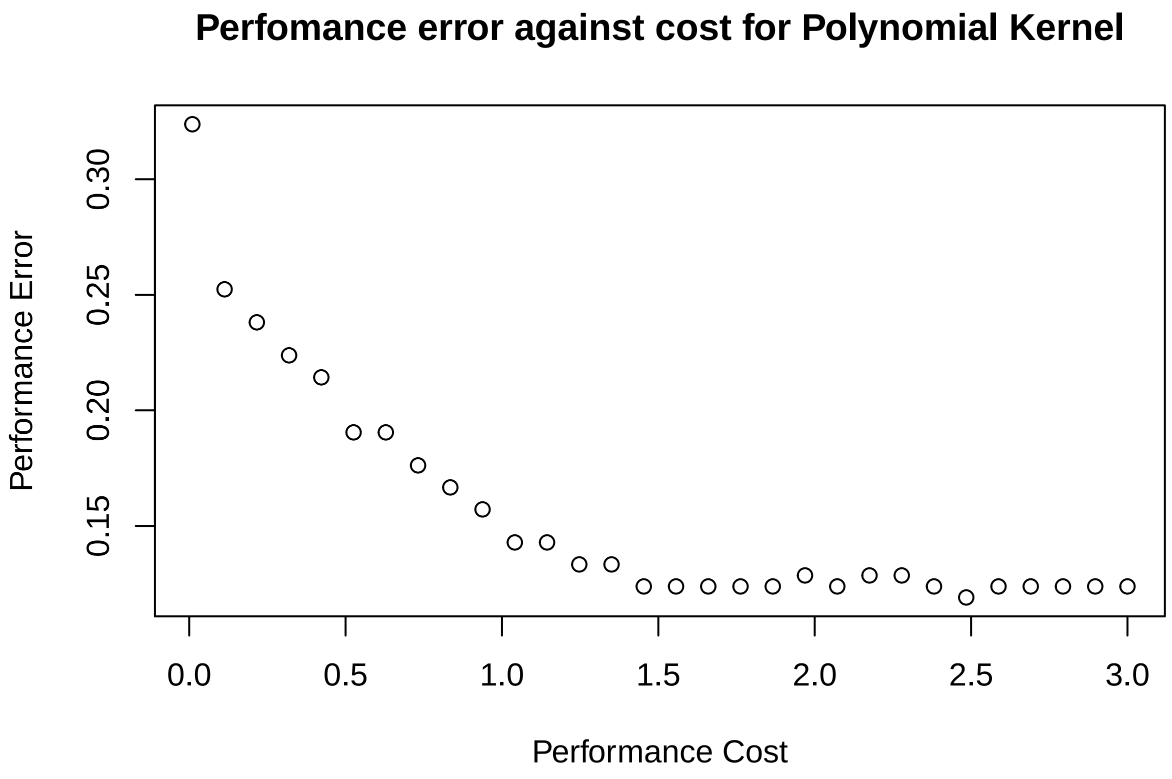
```
# Predicting test result linear
ypred_linear = predict(bestmod, model.test)
```

Polynomial Kernel

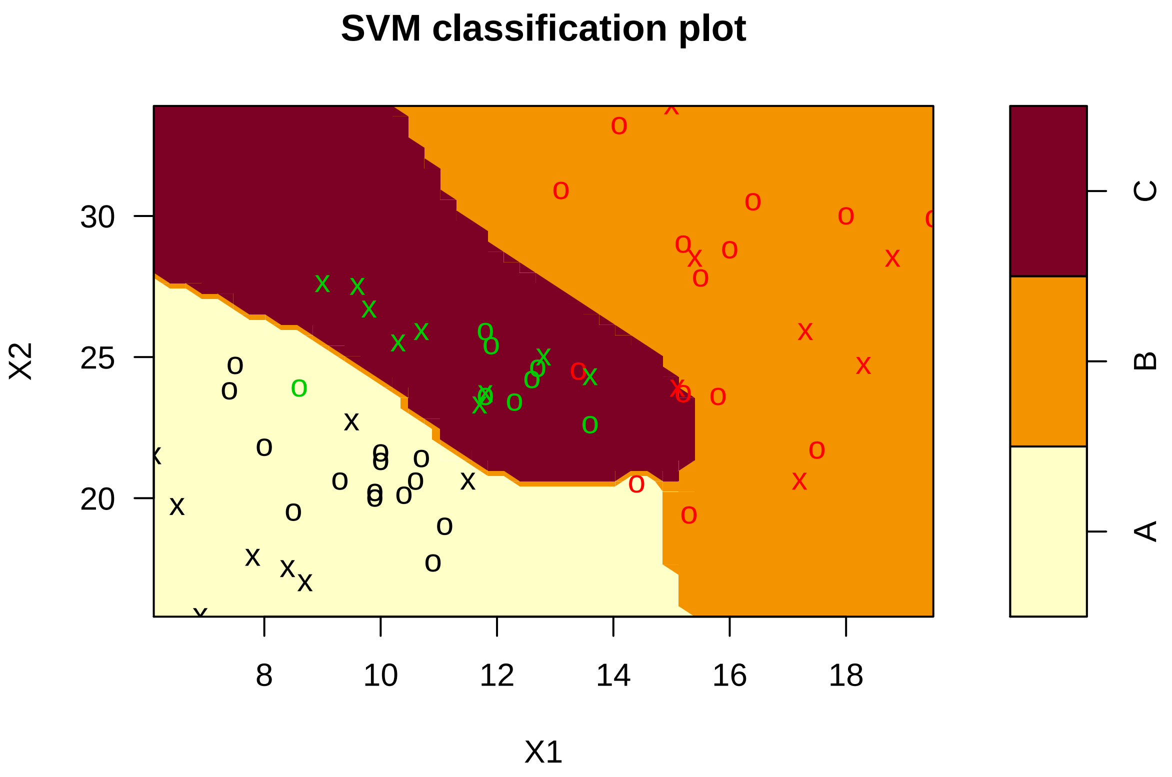
```r
# Create Polynomial Kernel
set.seed(1)
tune.out_poly = tune(svm, loc ~ X1 + X2, data = orchid[data.subset,],
                    kernel ="polynomial",
                    ranges = list(cost = seq(from = 0.01,to = 3, length = 30)))
```

Choosing the best cost parameter

```r
plot(tune.out_poly$performances$cost, tune.out_poly$performances$error,
     main = "Perfomance error against cost for Polynomial Kernel",
     xlab = "Performance Cost",ylab = "Performance Error")
```



The best cost parameter for the Polynomial kernel is 2.4844828



Computing the accuracy from the confusion matrix it can be seen that the accuracy from the Linear Kernel is higher than that of the Polynomial Kernel. It would be best to choose the kernel with the higher accuracy which is the Linear Kernel

---

Part (f) Comparison

Test Accuracy for KNN Method

```
##
## predict.knn  1  2  3
##           1 22  1  1
##           2  0 19  0
##           3  0  1 16
```

```
## [1] 0.95
```

Test Accuracy for Random Forest Bagging Method

```
##
## bag_predict  1  2  3
##           A 22  0  0
##           B  0 21  0
##           C  0  0 17
```

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

Test Accuracy for Linear Kernel Support Vector

```
##
## predict.knn  1  2  3
##           1 22  1  1
##           2  0 19  0
##           3  0  1 16
## [1] 0.9666667
```

Test Accuracy for Polynomial Kernel Support Vector

```
##
## ypred  1  2  3
##     A 22  1  1
##     B  0 17  0
##     C  0  3 16
```

```
## [1] 0.9166667
```

From our results it can be seen that the model with the best accuracy is the Random Forest Bagging method. This method came out as the best because the Bagging method helps in increasing the accuracy of the Random Forest method and also helps reduce variance of the model to help in predicting accurate results.
