Assignment 4

#Importing Libraries

```
library(randomForest)

## randomForest 4.6-14

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

library(caret)

## Loading required package: lattice

## Loading required package: ggplot2

## 
## Attaching package: 'ggplot2'

## The following object is masked from 'package:randomForest':
## 
##     margin

library(ggplot2)
library(mlbench)
```

#Importing Data

```
data(BreastCancer)
head(BreastCancer)

##        Id Cl.thickness Cell.size Cell.shape Marg.adhesion Epith.c.size
## 1 1000025            5         1          1             1            2
## 2 1002945            5         4          4             5            7
## 3 1015425            3         1          1             1            2
## 4 1016277            6         8          8             1            3
## 5 1017023            4         1          1             3            2
## 6 1017122            8        10         10             8            7
##   Bare.nuclei Bl.cromatin Normal.nucleoli Mitoses     Class
## 1           1           3               1       1    benign
## 2          10           3               2       1    benign
## 3           2           3               1       1    benign
## 4           4           3               7       1    benign
## 5           1           3               1       1    benign
## 6          10           9               7       1 malignant

levels(BreastCancer$Class)

## [1] "benign"    "malignant"
```

#Finding missing values and removing them

```
sum(is.na(BreastCancer))

## [1] 16

BreastCancer <- na.omit(BreastCancer)[,c(2:11)]
```

#Splitting the data into 70% training and 30% testing.

```
set.seed(15)
intrain <- createDataPartition(y = BreastCancer$Class, p = 0.7, list = FALSE)
training <- BreastCancer[intrain,]
testing <- BreastCancer[-intrain,]
```

#SVM model without grid search.

```
set.seed(15)
svm.model <- train(Class~., data = training, method = 'svmLinear', scale =
FALSE)
svm.model

## Support Vector Machines with Linear Kernel
##
## 479 samples
##   9 predictor
##   2 classes: 'benign', 'malignant'
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 479, 479, 479, 479, 479, 479, ...
## Resampling results:
##
##   Accuracy   Kappa
##   0.9514326  0.8932109
##
## Tuning parameter 'C' was held constant at a value of 1
```

#Confusion Matrix:

```
results <- predict(svm.model, testing, type = "raw")
table(results, testing$Class)

##
## results     benign malignant
##   benign       129         7
##   malignant      4        64
```

#Examining the following values for the mtry variable: 1, 2.5, 6.25

```
set.seed(15)
Grid_Serach <- expand.grid(.C=c(1,2.5,6.25))
svm.model2 <- train(Class~., data = training, method='svmLinear', tuneGrid =
Grid_Serach, scale = FALSE)
svm.model2

## Support Vector Machines with Linear Kernel
##
## 479 samples
##   9 predictor
##   2 classes: 'benign', 'malignant'
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 479, 479, 479, 479, 479, 479, ...
## Resampling results across tuning parameters:
##
##   C     Accuracy   Kappa
##   1.00  0.9514326  0.8932109
##   2.50  0.9493463  0.8885054
##   6.25  0.9461395  0.8814763
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was C = 1.
```

#Confusion Matrix:

```
pred_class <- predict(svm.model2, testing, type="raw")
head(pred_class)

## [1] benign    malignant benign    malignant malignant malignant
## Levels: benign malignant

table(pred_class, testing$Class)

##
## pred_class  benign malignant
##   benign       129         7
##   malignant      4        64
```

#We have an accuracy of 95.1% for C = 1.