Breast Cancer Prediction Using Machine Learning

The Data Set

```
cancer = read.csv("DataSet/breastcancer.csv")
```

Loading the libraries and setting the random number generator seed to 23

```
library(class)
library(tidyverse)
library(caTools)
library(rpart)
set.seed(23)
```

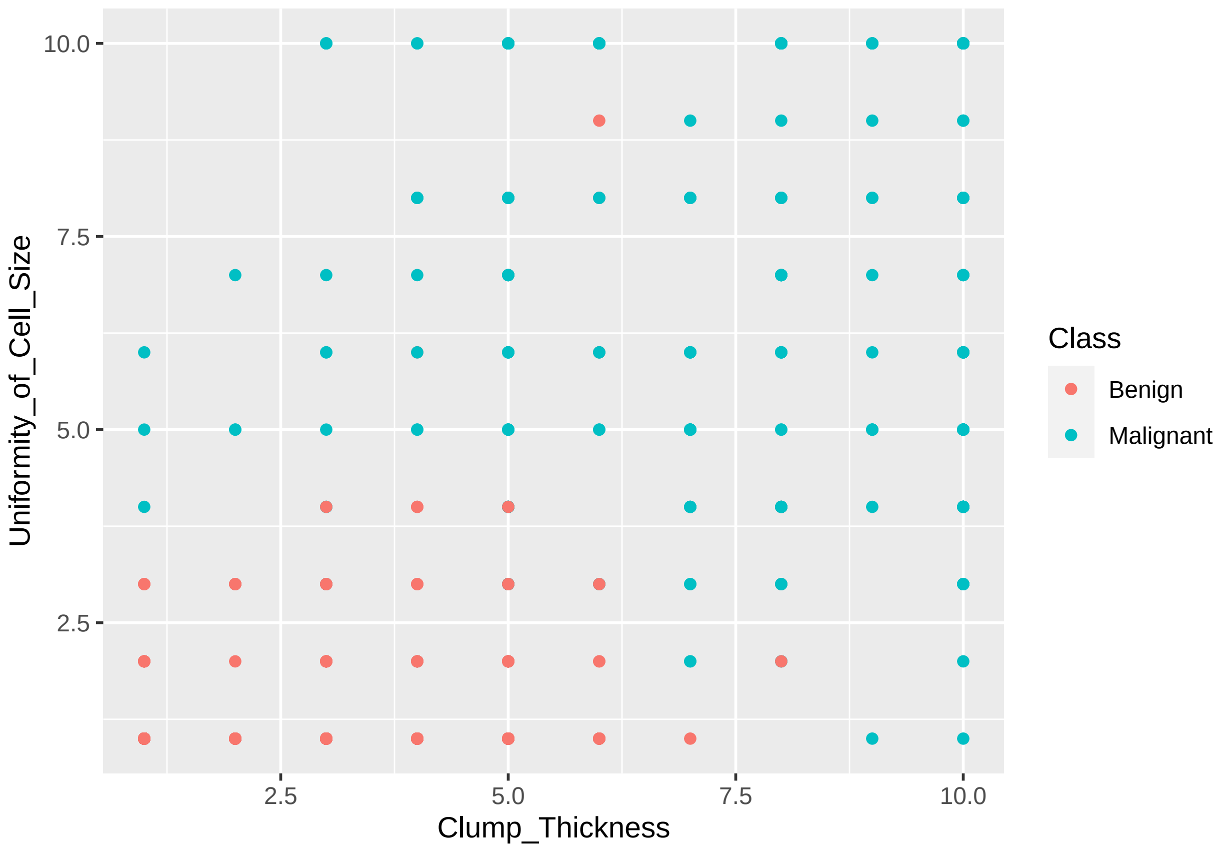
Cleaning and preparing the data

```
names(cancer) = c("Sample_ID",
                  "Clump_Thickness",
                  "Uniformity_of_Cell_Size",
                  "Uniformity_of_Cell_Shape",
                  "Marginal_Adhesion",
                  "Single_Epithelial_Cell_Size",
                  "Bare_Nuclei",
                  "Bland_Chromatin",
                  "Normal_Nucleoli",
                  "Mitoses",
                  "Class")
cancer = na.omit(cancer)
cancer$Class = factor(cancer$Class,
                      levels=c(2,4),
                      labels=c("Benign","Malignant"))
```

Adding meaningful column names, eliminating all NA values from the dataset and factoring the Class column values as Benign(2) and Malignant(4)

Question 1 - Create a scatter plot of Clump_Thickness vs. Uniformity_of_Cell_Size, color coded by the benign or malignant nature of the cell

```
ggplot(cancer) +
  geom_point(aes(x = Clump_Thickness, y = Uniformity_of_Cell_Size, color = Class))
```



Question 2 - Create functions that, given a confusion matrix, calculate sensitivity, specificity, accuracy and precision.

```
sensitivity = function(cm)
{
  return(cm[1,1]/(cm[1,1]+cm[1,2]))
}
specificity = function(cm)
{
  return(cm[2,2]/(cm[2,1]+cm[2,2]))
}
accuracy = function(cm)
{
  return((cm[1,1]+cm[2,2])/(cm[1,1]+cm[1,2]+cm[2,1]+cm[2,2]))
}
precision = function(cm)
{
  return(cm[1,1]/(cm[1,1]+cm[2,1]))
}
```

Functions created using the confusion matrix and its values TP, FP, TN, FN

Question 3: Separate Training and Test Data Sets

```
ind = sample(2, nrow(cancer), replace=TRUE, prob=c(0.67, 0.33))
```

Created training data set from the original cancer data set to correspond to 67% of the original available data.

#Question 4: Applying KNN

```
cancer.training = cancer[ind==1, 2:6]
cancer.test = cancer[ind==2, 2:6]
cancer.trainLabels = cancer[ind==1, 11]
cancer.testLabels = cancer[ind==2, 11]

prediction = knn(train = cancer.training,
                 test = cancer.test,
                 cl = cancer.trainLabels,
                 k=3)

(confusionMatrix = table(Actual_Value = cancer.testLabels,
                         Predicted_Value = prediction))
```

```
##              Predicted_Value
## Actual_Value Benign Malignant
##    Benign       158         7
##    Malignant      8        64
```

```
sensitivity(confusionMatrix)
```

```
## [1] 0.9575758
```

```
specificity(confusionMatrix)
```

```
## [1] 0.8888889
```

```
accuracy(confusionMatrix)
```

```
## [1] 0.9367089
```

```
precision(confusionMatrix)
```

```
## [1] 0.9518072
```

For KNN model the measures are:

Measure | Percentage

Sensitivity | 95.76%

Specificity | 88.89%

Accuracy | 93.67%

Precision | 95.18%

Question 5: Logistic Regression

```
(logisticModel = glm(cancer.trainLabels ~ Clump_Thickness +
                       Uniformity_of_Cell_Size +
                       Uniformity_of_Cell_Shape +
                       Marginal_Adhesion +
                       Single_Epithelial_Cell_Size,
                     data=cancer.training,
                     family='binomial'))
```

```
##
## Call:  glm(formula = cancer.trainLabels ~ Clump_Thickness + Uniformity_of_Cell_Size +
##     Uniformity_of_Cell_Shape + Marginal_Adhesion + Single_Epithelial_Cell_Size,
##     family = "binomial", data = cancer.training)
##
## Coefficients:
##                 (Intercept)              Clump_Thickness
##                    -10.4969                       0.9477
##     Uniformity_of_Cell_Size     Uniformity_of_Cell_Shape
##                      0.3193                       0.6698
##           Marginal_Adhesion  Single_Epithelial_Cell_Size
##                      0.2904                       0.3246
##
## Degrees of Freedom: 461 Total (i.e. Null);  456 Residual
## Null Deviance:       606.8
## Residual Deviance: 86.3  AIC: 98.3
```

```
prediction = predict(logisticModel, cancer.test, type='response')
cancer.test$predicted = ifelse(prediction>0.7, TRUE, FALSE)

(confusionMatrix = table(Actual_Value = cancer.testLabels,
                         Predicted_Value = prediction>0.7))
```

```
##              Predicted_Value
## Actual_Value FALSE TRUE
##    Benign      159    6
##    Malignant    11   61
```

```
sensitivity(confusionMatrix)
```

```
## [1] 0.9636364
```

```
specificity(confusionMatrix)
```

```
## [1] 0.8472222
```

```
accuracy(confusionMatrix)
```

```
## [1] 0.92827
```

```r
precision(confusionMatrix)
```

```
## [1] 0.9352941
```

For Logistic Regression model the measures are:

Measure | Percentage

Sensitivity | 96.36%

Specificity | 84.72%

Accuracy | 92.83%

Precision | 93.53%

Question 6: Decision Tree

```r
model = rpart(cancer.trainLabels ~
                Clump_Thickness +
                Uniformity_of_Cell_Size +
                Uniformity_of_Cell_Shape +
                Marginal_Adhesion +
                Single_Epithelial_Cell_Size,
              data=cancer.training,
              control=rpart.control(maxdepth=3),
              method='class')

prediction = predict(model, cancer.test, type='class')

(confusionMatrix = table(Actual_Value = cancer.testLabels,
                         Predicted_Value = prediction))
```

```
##              Predicted_Value
## Actual_Value Benign Malignant
##    Benign       155        10
##    Malignant      6        66
```

```r
sensitivity(confusionMatrix)
```

```
## [1] 0.9393939
```

```r
specificity(confusionMatrix)
```

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

```r
accuracy(confusionMatrix)
```

```
## [1] 0.9324895
```

```
precision(confusionMatrix)
```

```
## [1] 0.9627329
```

For Logistic Regression model the measures are:

Measure | Percentage

Sensitivity | 93.94%

Specificity | 91.67%

Accuracy | 93.25%

Precision | 96.27%

Question 7 - Is there one method that us best than the others, and why?

Decision tree algorithm works better compared to the other two models.

We can make this inference by looking at the specificity values from the three models.

Getting higher TypeI Errors (false positive) i.e, ‘Cancerous/Malignant’ predicted as ‘Healthy/Beningn’ is dangerous pertaining to this data set.

So, we must look at the specificity values to determine which model predictions are suitable.

Decision tree algorithm has specificity value of 91.6% which is the highest among the three models and thus, makes a better choice compared to others.