Aim: To conduct classification analysis using ANY ONE of the classification techniques.

KNN is a supervised learning technique. It has wide applications. we are given a labelled dataset consisting of training observations (x, y) and would like to capture the relationship between x and y. More formally, our goal is to learn a function h: X→Y so that given an unseen observation x, h(x) can confidently predict the corresponding output y.

About the data

Oncologist collected data on 10 parameters for patients suspected to breast cancer. dataset is complete, and we do not have any missing values. The diagnosis results are also available.

I am using this data to predict model for forthcoming patients to identify malignant or bening using KNN classification

Data Preparation

Load the dataset. The first column is "id" of patient which is nominal data thus eliminating row no 1.

```
#create object bc and import file
bc <- read.csv("Breast_Cancer.csv")
head(bc)
#here id is nominal data, which is a label
bc=bc[,-1] #removing first column
```

```
> table(bc$diagnosis)

  B   M
357 212
```

Frequency distribution table shows, 357 patients do not have cancer while 212 patients have cancer.

The parameters in dataset have different kind of scales, thus normalizing data on scale range 0 to 1 and creating a normalized data frame (bc_n)

```
normalize = function(x) {
        return((x-min(x)) / (max(x) - min(x)))
}

bc_n = as.data.frame(lapply(bc[2:31], normalize))
summary(bc_n) # <---- normalized
```

```
> summary(bc_n) # <---- normalized
  radius_mean      texture_mean     perimeter_mean     area_mean       smoothness_mean   compactness_mean
 Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000
 1st Qu.:0.2233   1st Qu.:0.2185   1st Qu.:0.2168   1st Qu.:0.1174   1st Qu.:0.3046   1st Qu.:0.1397
 Median :0.3024   Median :0.3088   Median :0.2933   Median :0.1729   Median :0.3904   Median :0.2247
 Mean   :0.3382   Mean   :0.3240   Mean   :0.3329   Mean   :0.2169   Mean   :0.3948   Mean   :0.2606
 3rd Qu.:0.4164   3rd Qu.:0.4089   3rd Qu.:0.4168   3rd Qu.:0.2711   3rd Qu.:0.4755   3rd Qu.:0.3405
 Max.   :1.0000   Max.   :1.0000   Max.   :1.0000   Max.   :1.0000   Max.   :1.0000   Max.   :1.0000
```

Creating Training and Testing data

Before creating training and test data, mix data for sampling and set seed. I have used 80-20 ratio to split data into training data set thus, 455 observations in training and 114 for testing. As we need to check if model is efficient thus need for creating 2 objects bc_train and bc_test.

```
set.seed(1000)

#randomly mixing data
bcrandom <- sample(1:nrow(bc_n), size = nrow(bc_n)*.8, replace = FALSE )

#creating training and test data set

bc_train = bc_n[bcrandom, ]
bc_test = bc_n[-bcrandom, ]
```

Creating labels to assign Bening and malignant to map actual findings

```
> bc_test_labels = bc[-bcrandom, 1]
> bc_test_labels
  [1] M M M B M M M B B M B B M B B B B M B B M B B B M B B B M M B M M B M B M B
 [39] B M B B B B B M M M B M M B M M B B B B B M B B B M B M B B B B M B B M B B
 [77] M B B B B B B B M B B B B B M B B B B B B B M B M M B M B B B B B B B B B B
Levels: B M
```

```
> bc_train_labels = bc[bcrandom, 1]
> bc_train_labels
  [1] M B M B M M B M B B B M B B B M B M B M M B B B B B B B B B M B B B B M B B
 [39] M B B B B B M M B B B M M M B M B M B B M B B B M M B B B M M M B M M B M M
 [77] M B B M B B M B M B B M M M B M B B B M B M M M M B B M B B B B M B M M M B
[115] M B B M M M B B B B M B M M M B M B M M M B B B M B M B M B B M M B B B M B
[153] M M B B B B B M B M B B B B M M M B B M M B M B M B B M B B M B B B M B M M
[191] B B B B B B B B B B B B B M B B B M B B B B M M B B B B B M B M B B B B M M
[229] M M B B M M B B B M B B B B M B M B B B B B B B B M B B M M M B M M M B M M
[267] M B B B B B B B B B B M M M B B B B B M B B B B M M M B B B M B B M B M M B
[305] M B B B M B B B B M B B B B B M B B B B B B B B B B M B B M B M M B B M M M
[343] B B M B B B B B M B M B B M M M M B B M B M B B M M B M B M B M B B B B M B
[381] M B M M B B M M M B M B B M M M B M M B M B B B B M M M M B M M B B M B B B
[419] M M B B M M M B B B B B M B M B B M B B B M B M B B B B B B B B B B B B M
Levels: B M
```

Training Model

K = Square root of number of observations. It should be an odd number to break the draw thus k = 23

```
> k=sqrt(nrow(bc))
> k
[1] 23.85372
```

Predicting Bening or Malignant using knn

```
library(class)
bc_test_pred = knn(train = bc_train, test = bc_test, cl = bc_train_labels, k=23)
```

```
> bc_test_pred
  [1] M M M B M M M B B M B M M B B B B M B B M B B B M B B B M M B B M B M B M B B M B B B B B
 [46] M M B B M M B B M B B B B B M B B B M B M B B B B M B B M B B M B B B B B B B M B B B B B
 [91] M B B B B B B B M B M M B M B B B B B B B B B B
Levels: B M
```

Accuracy

```
Confusion Matrix and Statistics

          Reference
Prediction  B  M
         B 73  2
         M  0 39

               Accuracy : 0.9825
                 95% CI : (0.9381, 0.9979)
    No Information Rate : 0.6404
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.9615
 Mcnemar's Test P-Value : 0.4795

            Sensitivity : 1.0000
            Specificity : 0.9512
         Pos Pred Value : 0.9733
         Neg Pred Value : 1.0000
             Prevalence : 0.6404
         Detection Rate : 0.6404
   Detection Prevalence : 0.6579
      Balanced Accuracy : 0.9756

       'Positive' Class : B
```

For the original data, we have 73 non-cancerous (Bening) cases while 41 Malignant cases. Now the KNN model predicted 75 non-cancerous (Bening) cases while 39 Malignant cases. Thus, it predicted 2 cases where Patient was suffering from cancer and model predicted that they do not have cancer.

The accuracy of model is **98.25%** thus saving time and accuracy.

Interpretation

KNN uses historical data to predict if value classifies in category. KNN is helpful in implementation of Breast cancer prediction with 98.25% accuracy. Out of which 73 cases have been accurately predicted as Benign (B) in nature which constitutes 64%. Also, 39 out of 114 observations were accurately predicted as Malignant (M) in nature which constitutes 34.2%. Thus, a total of 39 out of 114 predictions where TP i.e., True Positive in nature. There were 2 cases of False Negatives (FN) meaning 2 cases were malignant in nature but got predicted as Bening.

Reference

Available at: https://www.youtube.com/watch?v=xccONoz2zns

Available at: https://www.kaggle.com/junkal/breast-cancer-prediction-using-machine-learning/data