**AMERICAN INTERNATIONAL UNIVERSITY–BANGLADESH(AIUB)**

**FACULTY OF SCIENCE & TECHNOLOGY**
**DEPARTMENT OF CSE**

**Introduction to Data Science**
**Fall 2022-2023**

# Section: D

**FINAL Term Project**

**Submitted to**
**Tohedul Islam**
**Assistant Professor, CSE**

**Submitted By**

| Name | ID |
|---|---|
| MD. ALI AHNAF | 20-42378-1 |

# Dataset name & Description

The dataset is selected from Kaggle which was the original dataset of the National Institute of Diabetes and Digestive and Kidney Diseases. The objective of the dataset is to diagnostically predict whether a patient has diabetes, based on certain diagnostic measurements included in the dataset. All patients here are females at least 21 years old of Pima Indian heritage.
The source-link of the dataset: [https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database](https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database)

```
data <- read.csv("C:\\diabetes.csv")
str(data)
summary(data)
```

```
Console   Terminal   Background Jobs
R 4.2.2 · ~/
> data <- read.csv("C:\\diabetes.csv")
> str(data)
'data.frame':   768 obs. of  9 variables:
 $ Pregnancies             : int  6 1 8 1 0 5 3 10 2 8 ...
 $ Glucose                 : int  148 85 183 89 137 116 78 115 197 125 ...
 $ BloodPressure           : int  72 66 64 66 40 74 50 0 70 96 ...
 $ SkinThickness           : int  35 29 0 23 35 0 32 0 45 0 ...
 $ Insulin                 : int  0 0 0 94 168 0 88 0 543 0 ...
 $ BMI                     : num  33.6 26.6 23.3 28.1 43.1 25.6 31 35.3 30.5 0 ...
 $ DiabetesPedigreeFunction: num  0.627 0.351 0.672 0.167 2.288 ...
 $ Age                     : int  50 31 32 21 33 30 26 29 53 54 ...
 $ Outcome                 : int  1 0 1 0 1 0 1 0 1 1 ...
> summary(data)
  Pregnancies        Glucose      BloodPressure    SkinThickness      Insulin           BMI
 Min.   : 0.000   Min.   :  0.0   Min.   :  0.00   Min.   : 0.00   Min.   :  0.0   Min.   : 0.00
 1st Qu.: 1.000   1st Qu.: 99.0   1st Qu.: 62.00   1st Qu.: 0.00   1st Qu.:  0.0   1st Qu.:27.30
 Median : 3.000   Median :117.0   Median : 72.00   Median :23.00   Median : 30.5   Median :32.00
 Mean   : 3.845   Mean   :120.9   Mean   : 69.11   Mean   :20.54   Mean   : 79.8   Mean   :31.99
 3rd Qu.: 6.000   3rd Qu.:140.2   3rd Qu.: 80.00   3rd Qu.:32.00   3rd Qu.:127.2   3rd Qu.:36.60
 Max.   :17.000   Max.   :199.0   Max.   :122.00   Max.   :99.00   Max.   :846.0   Max.   :67.10
 DiabetesPedigreeFunction      Age          Outcome
 Min.   :0.0780           Min.   :21.00   Min.   :0.000
 1st Qu.:0.2437           1st Qu.:24.00   1st Qu.:0.000
 Median :0.3725           Median :29.00   Median :0.000
 Mean   :0.4719           Mean   :33.24   Mean   :0.349
```

The dataset has 9 attributes/variables where the class variable is an integer. There are total of 768 observations hence the dataset has 768 of rows.

Factoring the class variable to generate the Confusion Matrix because the data and the reference value must have to be factors and have the same no. of levels.

```
data['Outcome']=factor(data['Outcome'])
str(data)
```

Here, Outcome is the class variable of the dataset.

```
> data['Outcome']=factor(data['Outcome'])
> str(data)
'data.frame':   768 obs. of  9 variables:
 $ Pregnancies             : int  6 1 8 1 0 5 3 10 2 8 ...
 $ Glucose                 : int  148 85 183 89 137 116 78 115 197 125 ...
 $ BloodPressure           : int  72 66 64 66 40 74 50 0 70 96 ...
 $ SkinThickness           : int  35 29 0 23 35 0 32 0 45 0 ...
 $ Insulin                 : int  0 0 0 94 168 0 88 0 543 0 ...
 $ BMI                     : num  33.6 26.6 23.3 28.1 43.1 25.6 31 35.3 30.5 0 ...
 $ DiabetesPedigreeFunction: num  0.627 0.351 0.672 0.167 2.288 ...
 $ Age                     : int  50 31 32 21 33 30 26 29 53 54 ...
 $ Outcome                 : Factor w/ 2 levels "0","1": 2 1 2 1 2 1 2 1 2 2 ...
>
```

Normalizing the dataset where the values of each instance are between 0 to 1 and excluding the class variable (Outcome).

```r

data.subset <- data[c('Pregnancies','Glucose','BloodPressure','SkinThickness','Insulin','BMI','DiabetesPedigreeFunction','Age','Outcome')]
head(data.subset)

data_norm_func <- function(x){

  ((x - min(x))/ (max(x)-min(x)))

}

data.subset.n <- as.data.frame(lapply(data.subset[,1:8],data_norm_func))
str(data.subset.n)
summary(data.subset.n)
```

```
> data_norm_func <- function(x){
+
+   ((x - min(x))/ (max(x)-min(x)))
+
+ }
>
> data.subset.n <- as.data.frame(lapply(data.subset[,1:8],data_norm_func))
> str(data.subset.n)
'data.frame':   768 obs. of  8 variables:
 $ Pregnancies             : num  0.3529 0.0588 0.4706 0.0588 0 ...
 $ Glucose                 : num  0.744 0.427 0.92 0.447 0.688 ...
 $ BloodPressure           : num  0.59 0.541 0.525 0.541 0.328 ...
 $ SkinThickness           : num  0.354 0.293 0 0.232 0.354 ...
 $ Insulin                 : num  0 0 0 0.111 0.199 ...
 $ BMI                     : num  0.501 0.396 0.347 0.419 0.642 ...
 $ DiabetesPedigreeFunction: num  0.234 0.117 0.254 0.038 0.944 ...
 $ Age                     : num  0.483 0.167 0.183 0 0.2 ...
> summary(data.subset.n)
  Pregnancies         Glucose       BloodPressure    SkinThickness       Insulin             BMI         DiabetesPedigreeFunction
 Min.   :0.00000   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.00000   Min.   :0.0000   Min.   :0.00000
 1st Qu.:0.05882   1st Qu.:0.4975   1st Qu.:0.5082   1st Qu.:0.0000   1st Qu.:0.00000   1st Qu.:0.4069   1st Qu.:0.07077
 Median :0.17647   Median :0.5879   Median :0.5902   Median :0.2323   Median :0.03605   Median :0.4769   Median :0.12575
 Mean   :0.22618   Mean   :0.6075   Mean   :0.5664   Mean   :0.2074   Mean   :0.09433   Mean   :0.4768   Mean   :0.16818
 3rd Qu.:0.35294   3rd Qu.:0.7048   3rd Qu.:0.6557   3rd Qu.:0.3232   3rd Qu.:0.15041   3rd Qu.:0.5455   3rd Qu.:0.23409
 Max.   :1.00000   Max.   :1.0000   Max.   :1.0000   Max.   :1.0000   Max.   :1.00000   Max.   :1.0000   Max.   :1.00000
      Age
 Min.   :0.0000
 1st Qu.:0.0500
 Median :0.1333
 Mean   :0.2040
 3rd Qu.:0.3333
 Max.   :1.0000
>
```

The dataset has been split into two parts, Training and Validation/test set where 80 percent of the data were selected to train the classification model and the rest for validating the performance of the model. Here pseudorandom samples were generated and selected for both the training and rest for test set therefore the seed was initialized early for reproducibility.

```r
set.seed(146)
p <- 0.8
train <- sample(nrow(data.subset.n),nrow(data.subset.n)*p)
str(train)
summary(train)
updated_data.train <- data.subset[train,]
updated_data.val <- data.subset[-train,]
str(updated_data.train)
str(updated_data.val)

updated_data.train_labels <- data.subset[train,9]
updated_data.val_labels <- data.subset[-train,9]
str(updated_data.train_labels)
str(updated_data.val_labels)
NROW(updated_data.train_labels)
NROW(updated_data.val_labels)
```

```
> set.seed(146)
> p <- 0.8
> train <- sample(nrow(data.subset.n),nrow(data.subset.n)*p)
> str(train)
 int [1:614] 664 222 95 750 468 216 732 536 690 105 ...
> summary(train)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
    1.0   198.5   379.5   382.2   570.8   768.0
> updated_data.train <- data.subset[train,]
> updated_data.val <- data.subset[-train,]
> str(updated_data.train)
'data.frame':   614 obs. of  9 variables:
 $ Pregnancies             : int  9 2 2 6 0 12 8 4 1 2 ...
 $ Glucose                 : int  145 158 142 162 97 151 120 132 144 85 ...
 $ BloodPressure           : int  80 90 82 62 64 70 86 0 82 65 ...
 $ SkinThickness           : int  46 0 18 0 36 40 0 0 46 0 ...
 $ Insulin                 : int  130 0 64 0 100 271 0 0 180 0 ...
 $ BMI                     : num  37.9 31.6 24.7 24.3 36.8 41.8 28.4 32.9 46.1 39.6 ...
 $ DiabetesPedigreeFunction: num  0.637 0.805 0.761 0.178 0.6 0.742 0.259 0.302 0.335 0.93 ...
 $ Age                     : int  40 66 21 50 25 38 22 23 46 27 ...
 $ Outcome                 : Factor w/ 2 levels "0","1": 2 2 1 2 1 2 2 2 2 1 ...
> str(updated_data.val)
'data.frame':   154 obs. of  9 variables:
 $ Pregnancies             : int  1 2 10 0 11 13 5 4 3 7 ...
 $ Glucose                 : int  89 197 139 118 143 145 109 103 180 133 ...
 $ BloodPressure           : int  66 70 80 84 94 82 75 60 64 84 ...
 $ SkinThickness           : int  23 45 0 47 33 19 26 33 25 0 ...
 $ Insulin                 : int  94 543 0 230 146 110 0 192 70 0 ...
 $ BMI                     : num  28.1 30.5 27.1 45.8 36.6 22.2 36 24 34 40.2 ...
 $ DiabetesPedigreeFunction: num  0.167 0.158 1.441 0.551 0.254 ...
 $ Age                     : int  21 53 57 31 51 57 60 33 26 37 ...
 $ Outcome                 : Factor w/ 2 levels "0","1": 1 2 1 2 2 1 1 1 1 1 ...
>
> updated_data.train_labels <- data.subset[train,9]
> updated_data.val_labels <- data.subset[-train,9]
> str(updated_data.train_labels)
 Factor w/ 2 levels "0","1": 2 2 1 2 1 2 2 2 2 1 ...
> str(updated_data.val_labels)
 Factor w/ 2 levels "0","1": 1 2 1 2 2 1 1 1 1 1 ...
> NROW(updated_data.train_labels)
[1] 614
> NROW(updated_data.val_labels)
[1] 154
>
```

The labels for the class (class labels) were stored in two variables for applying those for classification models.

KNN algorithm was applied, and data were predicted for the validation set where K value was chosen by calculating square root of the total number of samples/observations in the training dataset. (Square root method)

## Confusion Matrix

```
library(class)
Ypred_knn=knn(updated_data.train,
              updated_data.val,
              cl=updated_data.train_labels,
              k=25)

confusion=table(Ypred_knn,updated_data.val_labels)
confusion
sum(diag(confusion))/nrow(updated_data.val)
```

```
> library(class)
> Ypred_knn=knn(updated_data.train,
+               updated_data.val,
+               cl=updated_data.train_labels,
+               k=25)
> confusion=table(Ypred_knn,updated_data.val_labels)
> confusion
         updated_data.val_labels
Ypred_knn  0  1
        0 99 17
        1 11 27
> sum(diag(confusion))/nrow(updated_data.val)
[1] 0.8181818
>
```

Here the confusion matrix was generated. In this table for predicted class variable values for test dataset, it was compared with the reference / reference datasets selected test data.



Here in the first quadrant in the confusion matrix table, true positive value is 99 which means 99 of the patients was actually non-diabetic from true class (reference data) and the model also predicted it

correctly as non-diabetic.

Moving into the 2nd quadrant, the false positive value is 17 therefore 17 patients did had diabetes, but they were incorrectly predicted as non-diabetic.

Moving into the 3rd quadrant, the false negative value is 11 therefore 11 patients did not had diabetes, but they were incorrectly predicted as diabetic.

Moving into the 4th quadrant, the true negative value is 27 therefore 27 patients were diabetic, and they were correctly predicted as diabetic.

The accuracy of the classifier was measured by the total sum of the diagonal value mainly the True Positive (TP) and the True Negative(TN) value divided by the number of observed test data's.
Hence, we get around 81.82% of accuracy from the KNN classification model by validating it with the unseen test/validation set.

```
library('caret')
confusionMatrix(Ypred_knn,updated_data.val_labels)
```

```
> sum(diag(confusion))/nrow(updated_data.val)
[1] 0.8181818
> library('caret')
> confusionMatrix(Ypred_knn,updated_data.val_labels)
Confusion Matrix and Statistics

          Reference
Prediction  0  1
         0 99 17
         1 11 27

               Accuracy : 0.8182
                 95% CI : (0.7481, 0.8757)
    No Information Rate : 0.7143
    P-Value [Acc > NIR] : 0.002041

                  Kappa : 0.5355

 Mcnemar's Test P-Value : 0.344704

            Sensitivity : 0.9000
            Specificity : 0.6136
         Pos Pred Value : 0.8534
         Neg Pred Value : 0.7105
             Prevalence : 0.7143
         Detection Rate : 0.6429
   Detection Prevalence : 0.7532
      Balanced Accuracy : 0.7568

       'Positive' Class : 0
```