**American International University – Bangladesh**

**Department of Computer Science & Engineering**



**Project Title: K-Nearest Neighbor (KNN) classification algorithm and its relevant tasks to a supervised Dataset.**

**Course: Introduction to Data Science**

|  |  |
| --- | --- |
| **Submitted by-** | **Submitted to-** |
|  |  |
| **Name: MD. SHOHAIBUR RAHMAN** | **Name: DR. ABDUS SALAM** |
| **ID: 20-42424-1** |  |
| **Section: C** |  |
| **Summer 2022-2023** |  |
| **B.Sc. CSE** |  |
|  |  |

**Dataset Description:**

The Heart Attack Analysis and Prediction Dataset, available at the is a comprehensive collection of data designed for exploring and predicting the occurrence of heart attacks. This dataset contains valuable information about various factors that could contribute to heart attacks.

Source: Kaggle

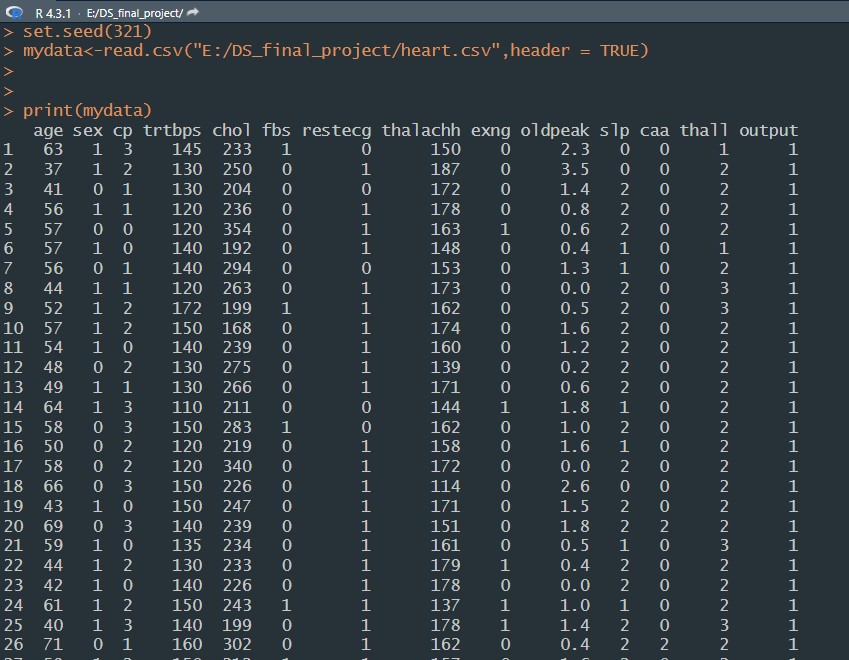
URL: *https://www.kaggle.com/datasets/rashikrahmanpritom/heart-attack-analysis-prediction-dataset*

**Attributes:**

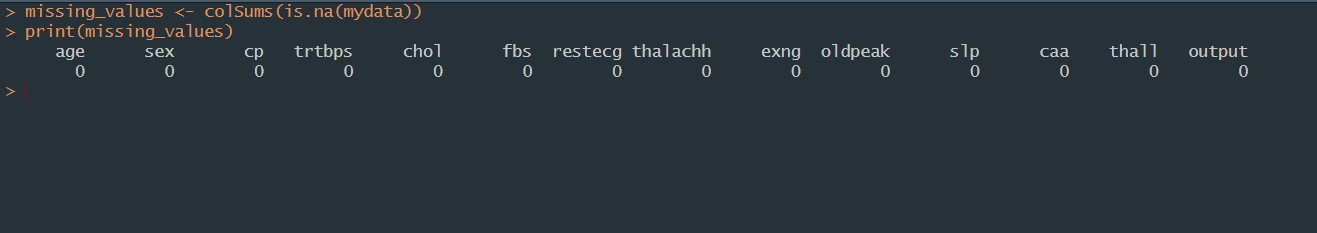
Attributes in the dataset include:

1. age,
2. sex,
3. chest pain type,
4. resting blood pressure,
5. cholesterol levels,
6. fasting blood sugar,
7. electrocardiographic results,
8. maximum heart rate achieved,
9. exercise-induced angina,
10. ST depression induced by exercise,
11. slope of the peak exercise ST segment,
12. number of major vessels colored by fluoroscopy,
13. thalassemia type and
14. the target variable indicating whether a person had a heart attack or not.

* **Importing the Dataset:** The dataset is located in a file called heart.csv in the current working directory. To begin data pre-processing using R, the first step is to import the dataset. Once imported, the heart.csv file is transformed into an R data frame and stored in a variable named "mydata". After printing the dataset, it looks like this-

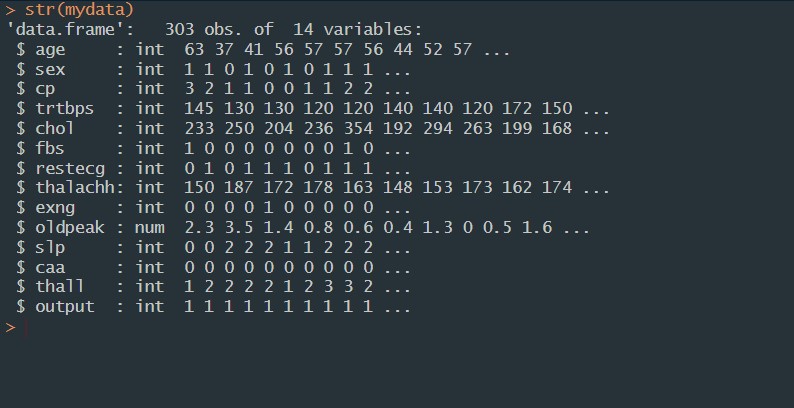


* **Finding Missing Values:**



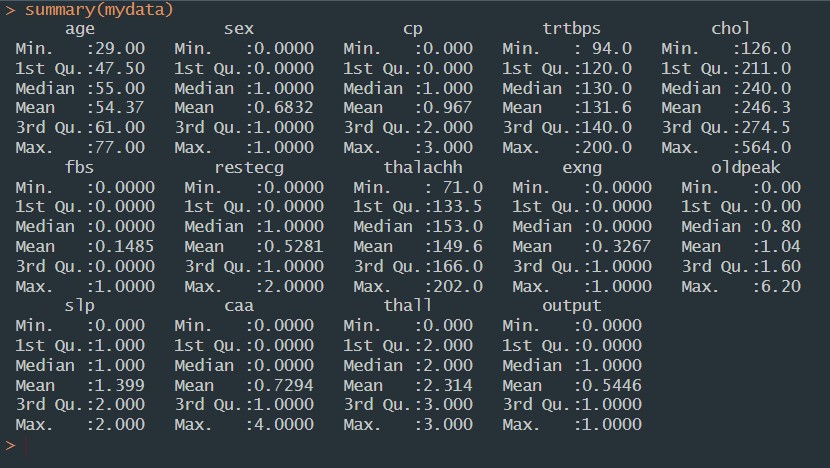
No missing values were found across the dataset.

* **Data structure:**

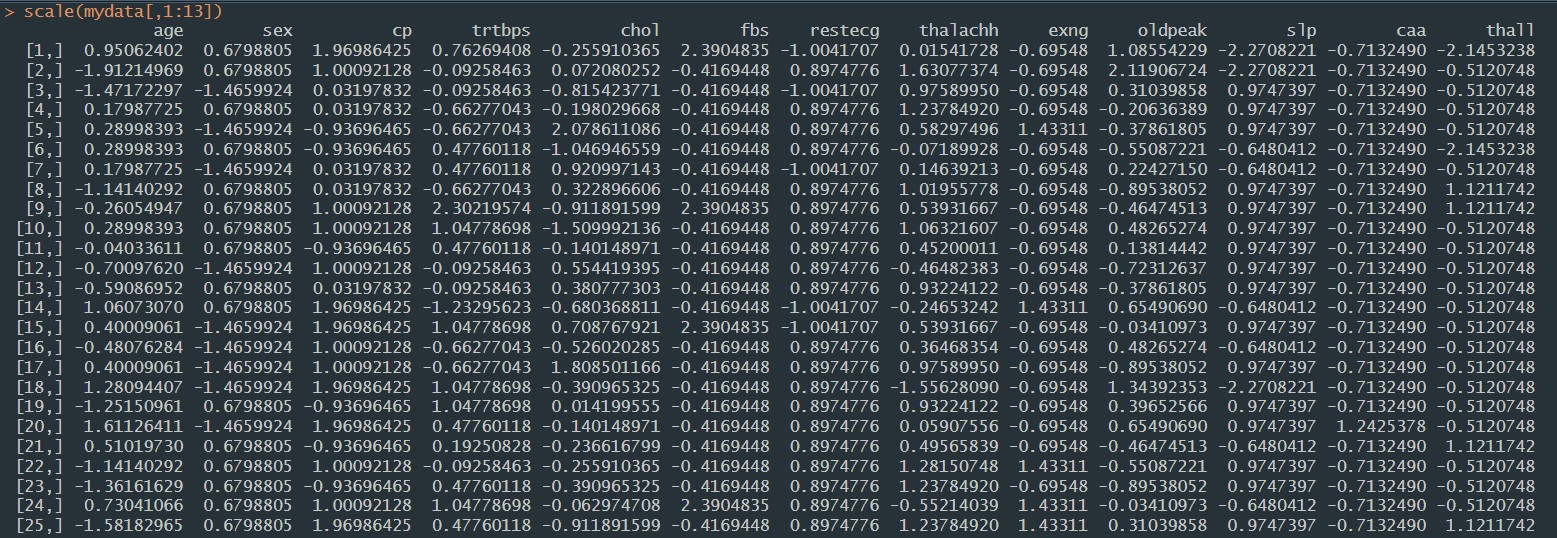


The dataset consists of all integer / numeric values. So, no conversion was needed since KNN uses only numeric value.

* **Data Summary:**

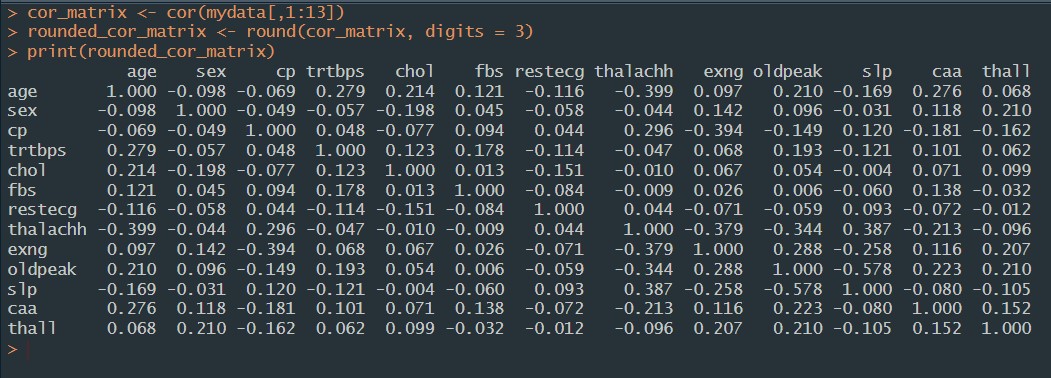


The summary() function was used to get an overview of key statistics and characteristics of the dataset, such as mean, median, minimum, maximum, and quartiles, enabling a quick understanding of the central tendencies and distribution of the data.

* **Scalling:** without target data

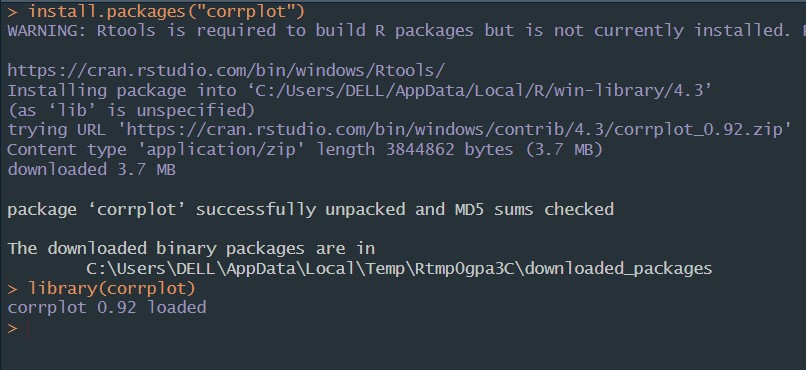
The scale() function was applied to standardize the numerical features of the dataset without including the target column (column 14 “Output”), ensuring that all feature values have a mean of zero and a standard deviation of one.

* **Round and Correlation Matrix:**



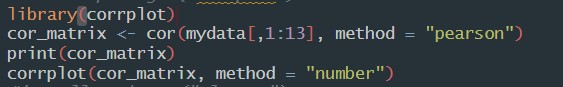
Pearson correlation matrix was used to quantify the linear relationship between pairs of variables in a dataset. It helps to identify how strongly and in what direction variables are related, which is valuable for understanding potential patterns and dependencies in the data.

* **Install and load corrplot:**

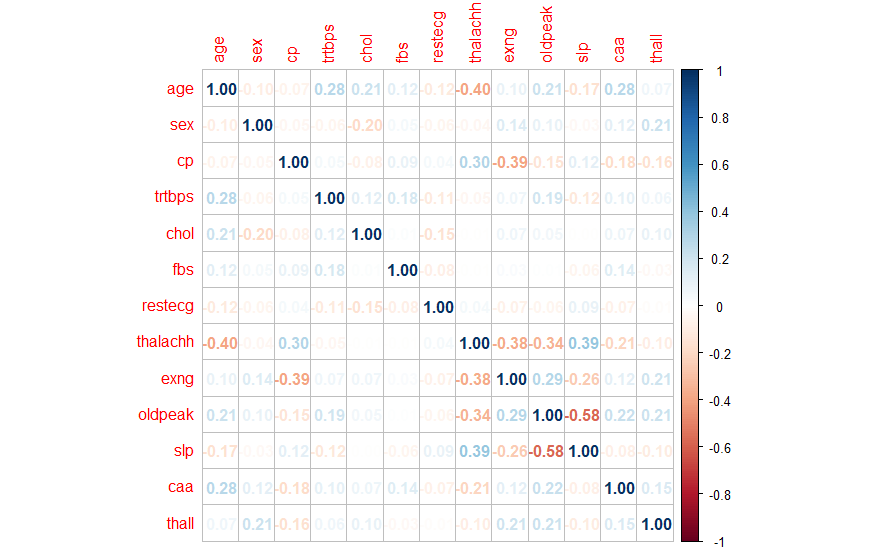


corrplot library is used to create visual representations of correlation matrices, such as heatmaps or clustered correlation plots.

* **Code:**

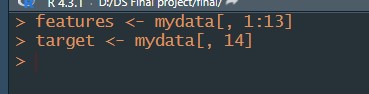


* **Visualize correlation:**



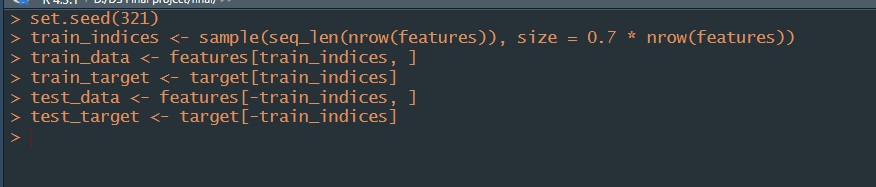
Pearson correlation method was used in matrix between the first 13 features in the dataset. It displays the correlation values, and creates a visualization using corrplot() with the correlation coefficients shown as numbers in a grid.

* **Define feature and target variable:**

****

Dataset was separated into features (independent variables) and the target variable (dependent variable).

* **Training and testing**

****

Random seed set.seed(321) was used to randomly split data and set 70% of data to train and 30% of data set to test.

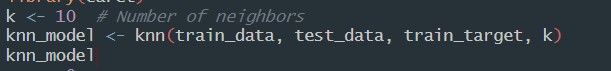
Created the train\_data (features for training) and train\_target (target values for training) datasets.

* **Install and load Caret library**

****

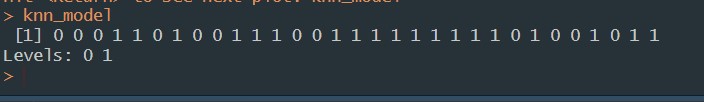
library(caret) and library(class) is required to accomplish KNN algorithm.

* **KNN model matrix:**

****

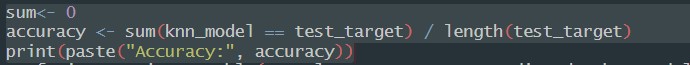
‘K’ is set to 10 which indicates nearest neighbour. knn\_model is created by applying the kNN algorithm to the ‘train\_data’ and ‘test\_data’ using the ‘train\_target’ for training and then predicting the classes for the ‘test\_data’. knn\_model holds the predicted classes (groups) for the test data based on the given value of k.

* **KNN model output:**

****

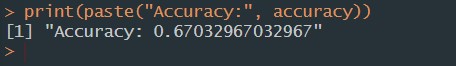
The levels 0 and 1 indicate the possible classes or outcomes that the algorithm can predict. In this case, 0 and 1 are the class labels. The Levels line specifies the possible class labels in dataset 0 (no heart disease) and 1 (heart disease).

* **KNN accuracy code:**

****

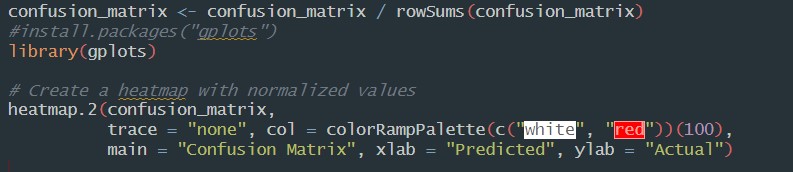
Summed up the number of correct predictions (where knn\_model matches test\_target) and dividing it by the total number of test instances which represents the ratio of correct predictions. It also indicates how well model performs on test data.

* **KNN accuracy determine:**

****

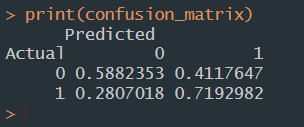
Accuracy was found “Accuracy: 0.67032967032967”. So about 67% of the predictions made by the model match the actual outcomes of dataset.

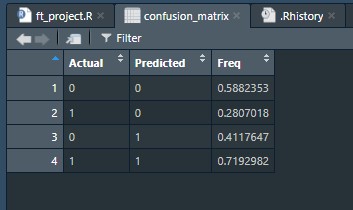
* **Confusion matrix code:**

****

Confusion matrix values are divided by the sum of each corresponding row's values. The confusion\_matrix summarizes the comparison between the actual test\_target values and the predicted knn\_model values. It shows how many instances were correctly predicted and how many were predicted inaccurately.

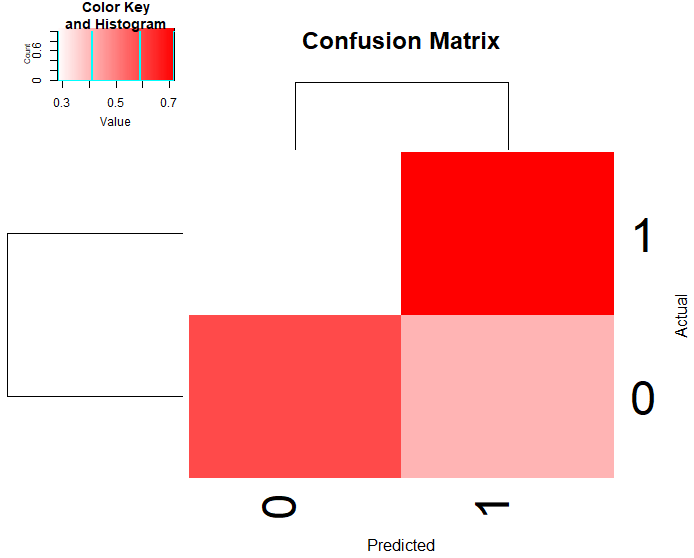
* **Visualize confusion matrix:**

****

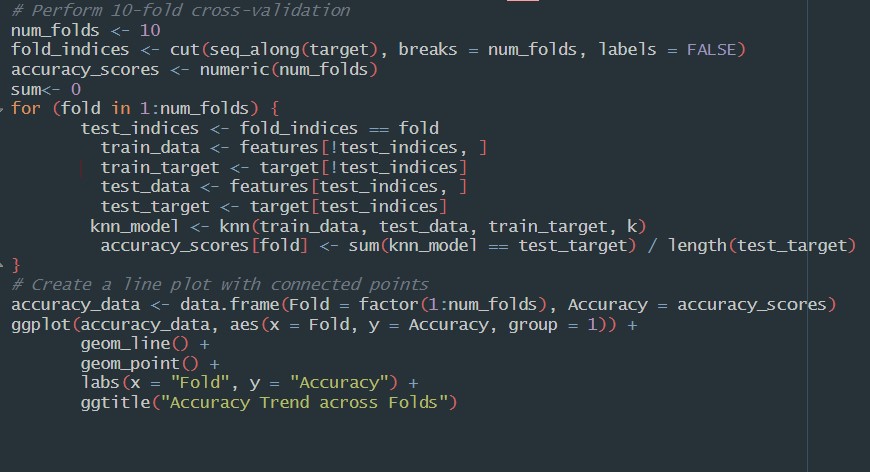
****

Here, frequency of 0.5882353 in the confusion matrix represents that 58% of the instances from a particular class were correctly classified by the model and so on for others as well. In a confusion matrix, each row represents the actual or true class labels of the instances, while each column represents the predicted class labels by the model.

* **Heatmap:**

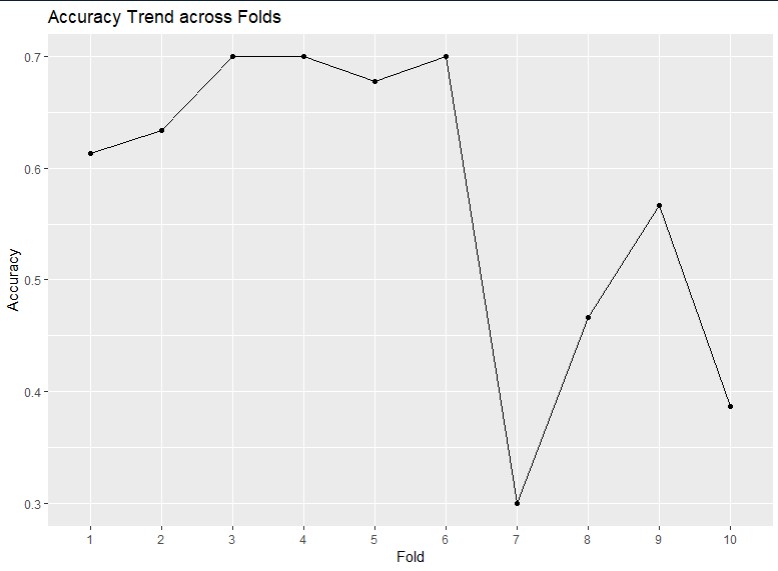
****

* **10-fold cross validation code :**

****

10-fold cross-validation were done by splitting the data into subsets, trains and tests a kNN model for each fold, calculates accuracy scores, and then creates a line plot showing how accuracy changes across the folds.

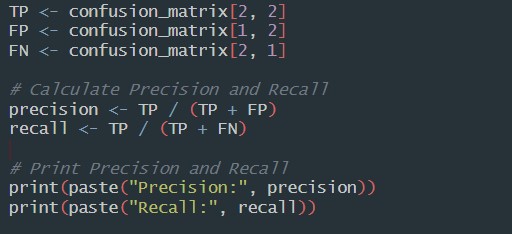
* **10-fold accuracy plot:**

****

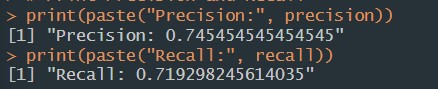
As 10 folds were selected for cross validation , the plot represents variation in model performance where 1 to 6 model performs correct accuracy with high value (over 60%) and 7th model with lowest (30%).

* **Calculate precision and recall:**

Code :



Result:



Precision and recall serve to assess how well a classification model performs on data. Precision emphasizes accurate positive predictions, vital when mistakes are expensive. Recall prioritizes the model's capability to identify all positives, especially when missing them is undesirable.

* **Conclusion:**

In conclusion, the analysis of the heart dataset provided valuable insights into predicting heart disease. By utilizing methods like k-nearest neighbors, cross-validation, and performance metrics like accuracy, precision, and recall we gained a deeper understanding of the dataset's patterns and predictive capabilities. These findings can contribute to better decision-making in healthcare and aid in identifying potential risk factors for heart disease.

Thank you.