Milestone 5: Final Project Paper

Predicting Risk of Developing Heart Disease

Dylan Dias, Chaitanya Patel, Thulasi Amaraneni, Mihir Manjrekar

June 2021

Milestone 1: Project Proposal

1 Description

Over the years there has been a large increase in the use of machine learning technologies in the field of health care. Health care generates huge amounts of data which consists of a variety of features of an individual patient which can be used for predictive analysis. This will help doctors to get a better insight of the patients health and the necessary medications can be provided to the patient to improve their health. For this project we are mainly focusing on predictive analysis of heart disease on the basis of patients various features.The term 'heart disease' refers to several unhealthy conditions of the heart. It is the most common reason for human death. Usually, women above the age of 45 and men above the age of 50 are most vulnerable to heart disease but there are high chances that it can affect anyone. Many risk factors like age, gender, obesity, unhealthy diet, hereditary, cigarette smoking, alcohol consumption, blood pressure and physical inactivity come into consideration when checking for cardiovascular diseases. The important issue about saving a patients life is the amount of time it takes for them to reach the hospital. It is a very challenging for doctors to diagnose this disease in its early stages. So in order to make their jobs more feasible we will be making use of various statistical and data mining tools to help predict the risk of developing heart disease. Our proposed solution predicts whether a patient has a risk of developing coronary heart disease in the near future by considering his/her risk factor attributes. This can help avoid incidents from being fatal.

2 Motivation

In this era to solve many medical problems many machine learning algorithm models are used. This helps in understanding the medical conditions of individual for efficient diagnosis of diseases and can suggest patients about a healthy living. Heart disease being one of the chronic diseases among individuals needs to be given necessary attention. This project's main motivation is to predict the occurrence of heart disease over a 10year period by analyzing several factors of an individual's habits like age, gender, obesity, unhealthy diet, hereditary, cigarette smoking, alcohol consumption, blood pressure and physical inactivity come into consideration when checking for cardiovascular diseases. To avoid any sort of delays because of any factors of a patient and diagnose the disease in early stages, more attention should be given to analyzing the data and identifying patterns that might be of useful insights.

In every hospital/clinic each transaction of a patient is a useful data. Instead of neglecting the data we can process the data and make it ready for analyzing through various techniques. So, it's very important to consider the chunks of all the patient's useful information and identify patterns which help in making efficient decisions.

3 Problem Definition

Many hospitals tend to collect patients' data to maintain records. That data can be used to answer simple queries about the patient's history and about their respective disease like the age, gender, past medications etc. These can help in answering simple questions like the average age of patients having heart disease, ratio of heart diseases among males and females, the gender which is most affected by heart disease. But its hard to respond to complex queries which involve in predicting the occurrence of heart diseases over a span of period, the treatment recommended to the patient with the given history of his condition.

All the decisions for the complex queries are taken by the doctor based on his experience and the recorded data is not taken into consideration most of the time. This practice leads to imperfect results which are not precise and often effects the quality of the diagnosing. So, to minimize these errors and to promote efficient consulting for patient's computer data support is necessary. We can implement data mining techniques and build machine learning models to extract knowledgeable insights from our medical data. This helps doctors to make effective decisions to treat the patients by improving decision making in the medical field.

4 Related Work and their limitations

There is fair enough amount of work done on heart disease risk prediction over the years. Some approaches are similar or are combination of different methods which we will be using. Hybrid models are used to get the better out of more than one approach.

Kim, Lee, and Lee (2015) uses an combination of decision trees and fuzzy logic. It uses decision tree to construct a rule base and the rule base is then inferred by a fuzzy rule bases system where these rules go through fuzzification to convert crisp data into fuzzy logic, an inference engine is then used to find the best possible match between data and the rules which are then defuzzified to get the predictions. Although using decision trees can be tricky sometimes, since a small change in the data can lead to large change in the overall tree structure and when handling big amount of data it can result in far more complex rules than needed. Also fuzzy logic is not always accurate since its results are based on assumptions and fuzzy based knowledge systems are inferior in terms of recognizing patterns compared to machine learning practices like neural networks.

Tasnim and Habiba (2021) uses a combination of random forest and decision trees, it uses random forest model for feature selection and then uses decision trees for classification. This approach very is simple, a RF model is used to extract important features, it takes in different subsets of features and generates n trees, the RF classifier calculates a support for each feature which states the importance of the feature. Once the features are selected it uses decision tree learning to generate a tree to classify the data. Since random forest consists of a large number of decision trees it comprises of the same problems that are face by decision trees i.e. due to high complexity of large datasets, any change in the data can influence many trees generated by the model. So as the data sample increases this approach may not be that useful.

5 Approach

We pre-processed the data to encode the feature vectors to numeric type if they are numeric and factor type if they are categorical.

We evaluated the data to measure the amount of missing values using the vis_miss function. Missing data creates imbalanced observations and causes biased estimates or could lead to invalid conclusions. Upon discovering a considerable amount of them, we decided to predict the missing values with the help of the mice package. This will help overcome issues like imbalanced observations and biased estimates that could lead to invalid conclusions.

For performing exploratory data analysis, we plotted our data for visualizations and a correlation heat map. This helped us learn more about the data set at hand and get an idea about which features could be more important for our analysis.

We selected four candidate algorithms to implement, Logistic Regression, Naive Bayes, K-Means and Support Vector Machine initially. We decided to divide them an among us after performing. Finally we compared the accuracies of each algorithm.

For our first model, we choose the logistic regression using glm function. We run the model on all values and based on the p-value we use the most significant variables . A summary of the model indicates that male, age, cigsPerDay, prevalentStroke, sysBP, and glucose to be the most significant. We split the data into training and testing data sets. We built a model based on only these variables using the training data and predicted the value of the TenYearCHD variable of the testing data. We use these predicted values determine the accuracy of the model using the confusion matrix.

For our second model, we implemented the naive bayes algorithm. Using the TenYearCHD as the target attribute we trained a naive bayes model with the training data. Then we used this classifier to predict the values for the testing data and compared them to the original target values. Then we obtained the accuracy of this classifier using the confusionMatrix function.

For our third model, we implemented K means clustering. K Means utilizes the Euclidean distance to measure the proximity of data points. We need to normalize the data for which we wrote a function that we applied to all the feature vectors. For the K means model, we use the number of centers as 2 because we know from our domain knowledge that there are only 2 clusters in the data. We are setting the nstart attribute to 20 as we wanted to limit the random restarts to a lower value as a higher value was not improving the results. Using the confusion matrix of the predicted values and the target values we get the accuracy for this model.

For our fourth model, we implemented Support vector machines. We split the data into training and testing data sets and feature scaling is carried out on both of them. We use the trainControl method to define 5 repetitions of 10 fold cross validation to be performed for training and the train function is used to train and tune a SVM model with a radial kernel. In the first training attempt we let the model use default sigma and arbitary C values. In the second training attempt we use the tuneGrid parameter to use a combination of the values of sigma of around 0.068 and C of around 1. This gives us the model with the best ROC for the parameters used. This model is used to predict on the test data and the results are compared with the target feature to get the accuracy from the confusion matrix.

For the last algorithm we chose Neural networks. It requires all our attributes to be numeric and we further we applied min-max normalization on our features for scaling. We split the data into training and testing data sets. For training, we use a feed-forward network having 15 input neurons, 2 output neurons and one hidden layer containing 8 neurons. We have used the back-propagation algorithm to update our weights with a learning rate of 0.005, cross-entropy loss is the error function with the sigmoid activation function used in the neurons. Since the package we used for neural networks has some open issues we weren't able set particular values as parameters.

6 Key issues and alternate ways to resolve those issues

Since heart disease is considered as one of the most impactful chronic diseases, diagnosing it needs to be taken care of in early stages. Attributes like age, gender, obesity, unhealthy diet, hereditary, cigarette smoking, alcohol consumption, blood pressure and physical inactivity are to be considered for the prediction. After pre-processing the data, based on the parameters in the dataset we will be training our model by building machine learning algorithms and with the help of test data we will be predicting the results of our model. By evaluating the accuracy, we can choose the best model. The machine learning algorithms that we implemented in the project include Naïve Bayes, Logistic Regression, K-means, SVM and neural networks. In all these algorithms we will be splitting our dataset into training and testing data, implement scaling, build the model, and evaluate the accuracy of all the models.

7 System Functions

Our system applies data mining techniques to extract useful insights and patterns from the data. Through supervised learning, we can train our model and then by applying classification algorithms, predictions can be made. By analyzing the performance of classification algorithms (Decision Tree, Naïve Bayes, etc.) on our data and taking the accuracy of algorithms into consideration we can evaluate each model with the help of the best-fit algorithm. We plan on examining various features from the data-set which will then be applied to 5 algorithms into order to see which one will be the best fit for the system. The main goal is to use this data to predict if a patient is at risk of developing a heart disease. We will be making use of the following algorithms Support Vector Machine, Neural Nets, Random Forest, Naive Bayes and Logistic Regression. We have decided to use supervised algorithms for now since the task is mostly classification, but we will progressively look for more algorithms to test like trying to use unsupervised algorithms for this case.

8 Uses

It is very important to have a tool that predicts (with high accuracy) the unhealthy heart as it can help the individuals as well as doctors in early diagnosis of the heart disease. It is also important to have a low false negative rate for our model as missing out on predicting any patients heart disease could cost a human life. Also, we will analyze various patterns in our data-set to check which people are more vulnerable (based on age and gender) to heart diseases so that they can plan a healthy living. Accurate predictions could lead the person to change their lifestyle and seek out medical help early which could reduce the death rates due to cardiovascular diseases.

9 Dataset

For this project, we opted for the Framingham Heart Study data-set which comprises of numerical and categorical data. It consists of 4240 records with 15 features. Each record contains categorical data (gender, age, education) of an individual as well as his/her health factors (like blood pressure, cigarette consumption, stroke rate, cholesterol, etc.). The features were selected over a wide range of study on individuals who have gone through a heart stroke and those who haven't. The description for each feature is given below:

Dataset link: Heart_Disease_Risk_Dataset

- Age : The study was conducted on individual with age ranging from 29 to 62.
- Sex : The gender of the induvidual is given as 1 = male, 2 = female.
- Systolic Blood Pressure(sysBP) : It is the measure of the force exerted by the heart on the walls of arteries each time it beats.
- Diastolic Blood Pressure(diaBP) : It is the measure of the force exerted by the heart on the walls of arteries in between beats.
- Total Cholesterol(chol) : It's the measure of cholesterol level in blood.
- Body Mass Index(bmi) : body mass index of the individual.
- currentSmoker : 1 = yes, 0 = no.
- cigsPerDay : Numerical value, Number of cigarettes smoked by the individual per day.
- Blood Pressure Medication(BPmeds) : number of blood pressure medications the individual had over 10 years.
- Prevalent Stroke : History of strokes.
- diabetes : 1 = yes, 0 = no.
- heart rate : heart rate of the individual.
- glucose : glucose in blood, numeric value.
- Prevalent Hypertension : Does the individual have high blood pressure, 1 = yes, 0 = no.
- TenYearCHD : Congenital heart disease, 1 = yes, 0 = no.

10 Algorithms

We are going to test the following Algorithm for this use case:

10.1 Logistic regression

The logistic regression model is a classification algorithm which is mostly used in predicting the probability of the binary classes. For example, predicting whether a patient has 10-year risk of developing coronary heart disease (CHD).

10.2 Support Vector Machine

It is a supervised machine learning algorithm that can be used for classification or regression problems. We treat each data item as a point in n-dimensional space where n is the number of features that distinctly classifies each data point. We then find the hyper plane that helps to distinguish between the data points with the largest amount of margin.

10.3 Naive Bayes

The Naive Bayes classifier is a probabilistic machine learning model based on Bayes Theorem which assumes that the predictors are independent i.e. each feature in a class is not related to other features present. For example, classify whether the day is suitable for playing golf, given the features of the day like humidity, windy and temperature.

10.4 Random Forest

Random forest is an ensemble learning method which consists of many decision trees. These individual decision trees are able to compute their own predictions. For the final prediction the random forest classifier aggregates the prediction of the individual decision trees.

10.5 Neural Network

Neural Network is an information processing model that has the ability to learn by analyzing training examples. They are able to identify complex patterns in the dataset using hidden layers and activation functions. Some neural network applications include image processing, character recognition etc.

Milestone 2: Data summary/visualization

This project is based on *"Predicting risk of heart disease"* available on National Heart Lung and Blood Institute.

URL :https://biolincc.nhlbi.nih.gov/studies/framcohort/

Imported Libraries

List of libraries that need to be installed to run this file

```
library(kableExtra)
library(plyr)
library(tidyverse)
library(gmodels)
library(ggmosaic)
library(corrplot)
library(mice)
library(caret)
library(rpart)
library(cluster)
library(fpc)
library(data.table)
library(knitr)
library(naniar)
library(visdat)
library(Hmisc)
library(shiny)
library(caTools)
library(corrplot)
library(e1071)
library(naivebayes)
library(neuralnet)
library(GGally)
library(pROC)
```

Dataset

The dataset we are using is from Framingham Heart Study data-set which comprises of numerical and categorical data. It consists of 4240 records with 15 features. Each record contains categorical data (gender, age, education) of an individual as well as his/her health factors (like blood pressure, cigarette consumption, stroke rate, cholesterol, etc.). The features were selected over a wide range of study on individuals who have gone through a heart stroke and those who haven't which is signified with the variable *TenYearCHD*.

The dataset structure is given below: The `read.csv()` command imports the dataset.

```
framingham <- read.csv("framingham.csv", header = T)
```

Checking the number of rows and columns in the dataset

```
ncol(framingham)
```

```
## [1] 16
```

```
nrow(framingham)
```

```
## [1] 4240
```

Data-set structure

```
str(framingham)
```

```
## 'data.frame':    4240 obs. of  16 variables:
##  $ male           : int  1 0 1 0 0 0 0 0 1 1 ...
##  $ age            : int  39 46 48 61 46 43 63 45 52 43 ...
##  $ education      : int  4 2 1 3 3 2 1 2 1 1 ...
##  $ currentSmoker  : int  0 0 1 1 1 0 0 1 0 1 ...
##  $ cigsPerDay     : int  0 0 20 30 23 0 0 20 0 30 ...
##  $ BPMeds         : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ prevalentStroke: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ prevalentHyp   : int  0 0 0 1 0 1 0 0 1 1 ...
##  $ diabetes       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ totChol        : int  195 250 245 225 285 228 205 313 260 225 ...
##  $ sysBP          : num  106 121 128 150 130 ...
##  $ diaBP          : num  70 81 80 95 84 110 71 71 89 107 ...
##  $ BMI            : num  27 28.7 25.3 28.6 23.1 ...
##  $ heartRate      : int  80 95 75 65 85 77 60 79 76 93 ...
##  $ glucose        : int  77 76 70 103 85 99 85 78 79 88 ...
##  $ TenYearCHD     : int  0 0 0 1 0 0 1 0 0 0 ...
```

```
summary(framingham)
```

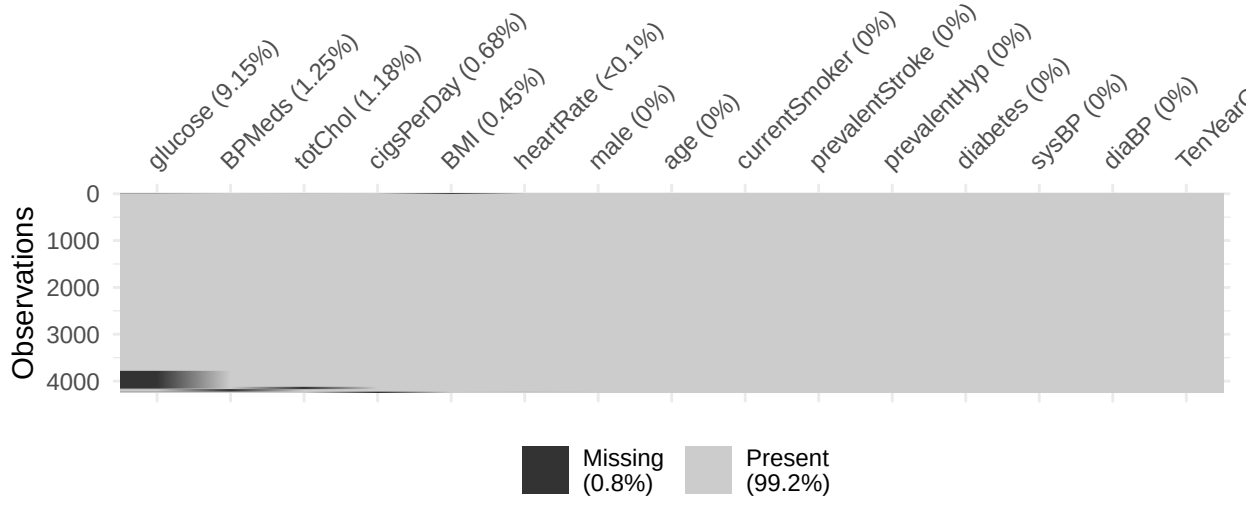
```
##       male             age          education     currentSmoker
##  Min.   :0.0000   Min.   :32.00   Min.   :1.000   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.:42.00   1st Qu.:1.000   1st Qu.:0.0000
##  Median :0.0000   Median :49.00   Median :2.000   Median :0.0000
##  Mean   :0.4292   Mean   :49.58   Mean   :1.979   Mean   :0.4941
##  3rd Qu.:1.0000   3rd Qu.:56.00   3rd Qu.:3.000   3rd Qu.:1.0000
##  Max.   :1.0000   Max.   :70.00   Max.   :4.000   Max.   :1.0000
##                                   NA's   :105
##    cigsPerDay         BPMeds        prevalentStroke     prevalentHyp
##  Min.   : 0.000   Min.   :0.00000   Min.   :0.000000   Min.   :0.0000
##  1st Qu.: 0.000   1st Qu.:0.00000   1st Qu.:0.000000   1st Qu.:0.0000
##  Median : 0.000   Median :0.00000   Median :0.000000   Median :0.0000
##  Mean   : 9.006   Mean   :0.02962   Mean   :0.005896   Mean   :0.3106
##  3rd Qu.:20.000   3rd Qu.:0.00000   3rd Qu.:0.000000   3rd Qu.:1.0000
##  Max.   :70.000   Max.   :1.00000   Max.   :1.000000   Max.   :1.0000
##  NA's   :29       NA's   :53
##     diabetes          totChol          sysBP           diaBP
##  Min.   :0.00000   Min.   :107.0   Min.   : 83.5   Min.   : 48.0
##  1st Qu.:0.00000   1st Qu.:206.0   1st Qu.:117.0   1st Qu.: 75.0
##  Median :0.00000   Median :234.0   Median :128.0   Median : 82.0
##  Mean   :0.02571   Mean   :236.7   Mean   :132.4   Mean   : 82.9
##  3rd Qu.:0.00000   3rd Qu.:263.0   3rd Qu.:144.0   3rd Qu.: 90.0
##  Max.   :1.00000   Max.   :696.0   Max.   :295.0   Max.   :142.5
```

```
##                   NA's   :50
##       BMI          heartRate         glucose         TenYearCHD
##  Min.   :15.54   Min.   : 44.00   Min.   : 40.00   Min.   :0.0000
##  1st Qu.:23.07   1st Qu.: 68.00   1st Qu.: 71.00   1st Qu.:0.0000
##  Median :25.40   Median : 75.00   Median : 78.00   Median :0.0000
##  Mean   :25.80   Mean   : 75.88   Mean   : 81.96   Mean   :0.1519
##  3rd Qu.:28.04   3rd Qu.: 83.00   3rd Qu.: 87.00   3rd Qu.:0.0000
##  Max.   :56.80   Max.   :143.00   Max.   :394.00   Max.   :1.0000
##  NA's   :19      NA's   :1        NA's   :388
```

In order to prepare the data for pre-processing we will convert the attributes to factors. We have also removed the education column from the data-set. As there is no relation between education and heart disease.

```
framingham$male = as.factor(framingham$male)
framingham$education = as.factor(framingham$education)
framingham$currentSmoker = as.factor(framingham$currentSmoker)
framingham$BPMeds = as.factor(framingham$BPMeds)
framingham$prevalentStroke = as.factor(framingham$prevalentStroke)
framingham$prevalentHyp = as.factor(framingham$prevalentHyp)
framingham$diabetes = as.factor(framingham$diabetes)
framingham$glucose = as.numeric(framingham$glucose)
framingham$heartRate = as.numeric(framingham$heartRate)
framingham$totChol = as.numeric(framingham$totChol)
framingham$cigsPerDay = as.numeric(framingham$cigsPerDay)
framingham$age = as.numeric(framingham$age)
framingham$TenYearCHD = as.factor(framingham$TenYearCHD)
framingham_new <- framingham[,-c(3)]
```

Checking for missing values present in the data-set

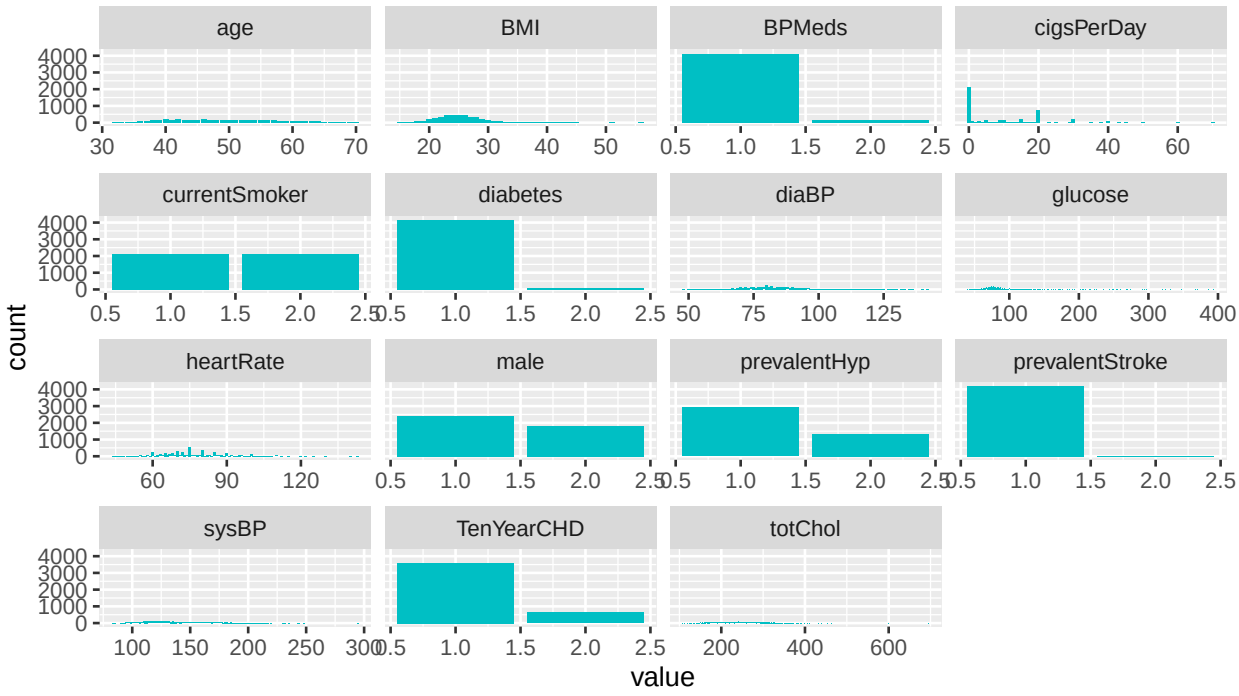


From the missing data plot, we can see that there is 0.8% missing data present in the data-set. In order to deal with missing values we will make use of the mice function to predict them.

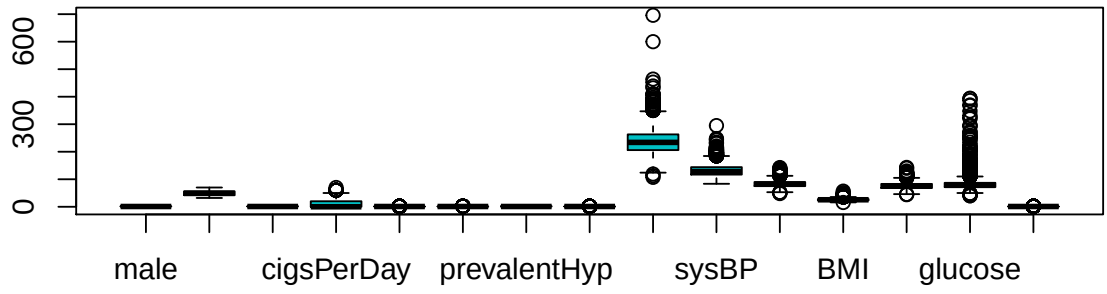
```
new_dataset <- mice(framingham_new, m = 5, method = c("", "", "", "pmm", "logreg", "", "", "", "pmm", "
final_cleaned <- mice::complete(new_dataset, 2)
```

Visualizations

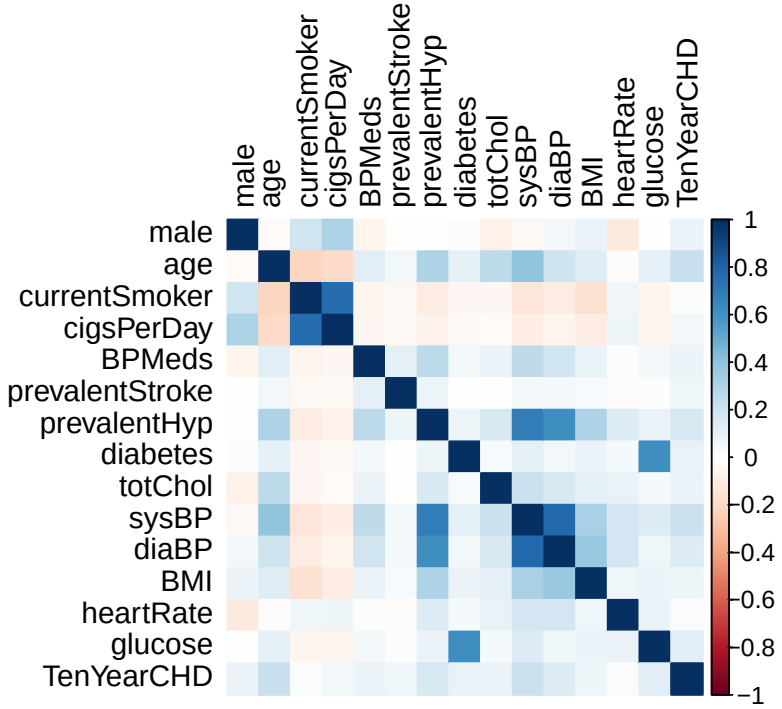
This plot shows the distribution of the data. The data is fairly distributed for features like *age*, *BMI*, *cigsPerDay*, *diaBP*, *sysBP*, *male*, *heartRate*, *totChol* and it also has features with unbalanced data distribution like *BPMeds*, *glucose*, *prevalentHyp*, *prevalentStroke* and *TenYearCHD*



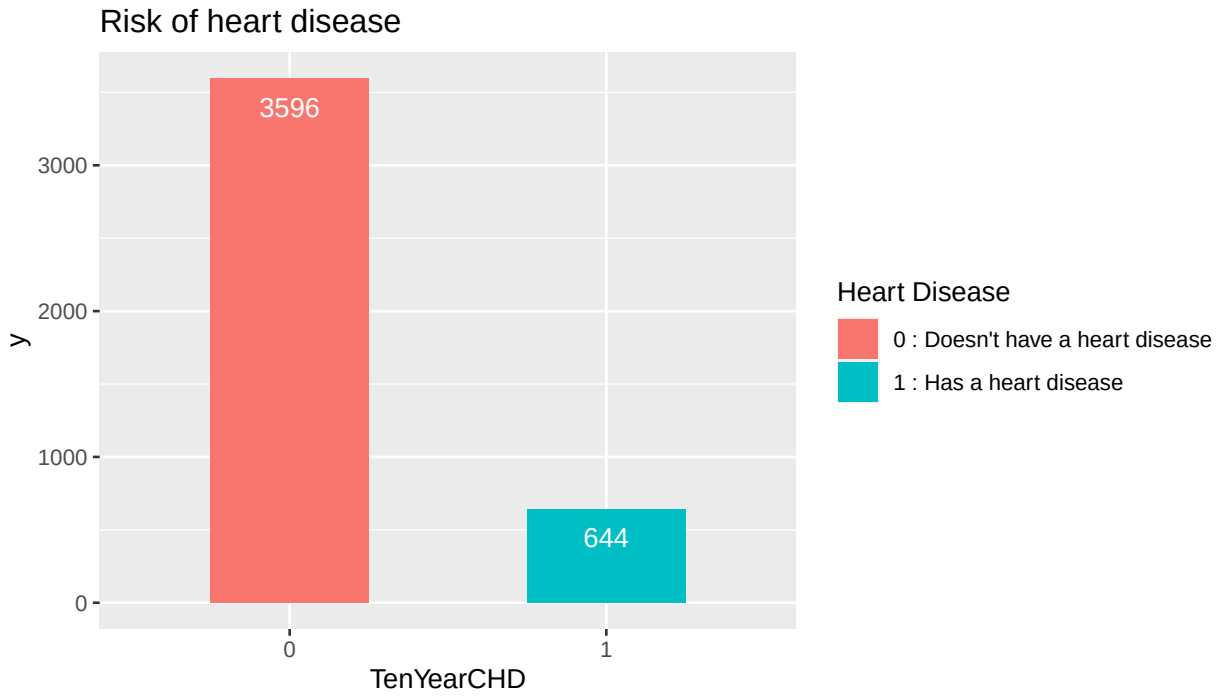
The box plots below shows that there are outliers present in the following features cigPerDay, totChol, sysBP, diaBP, BMI, heartRate, and glucose



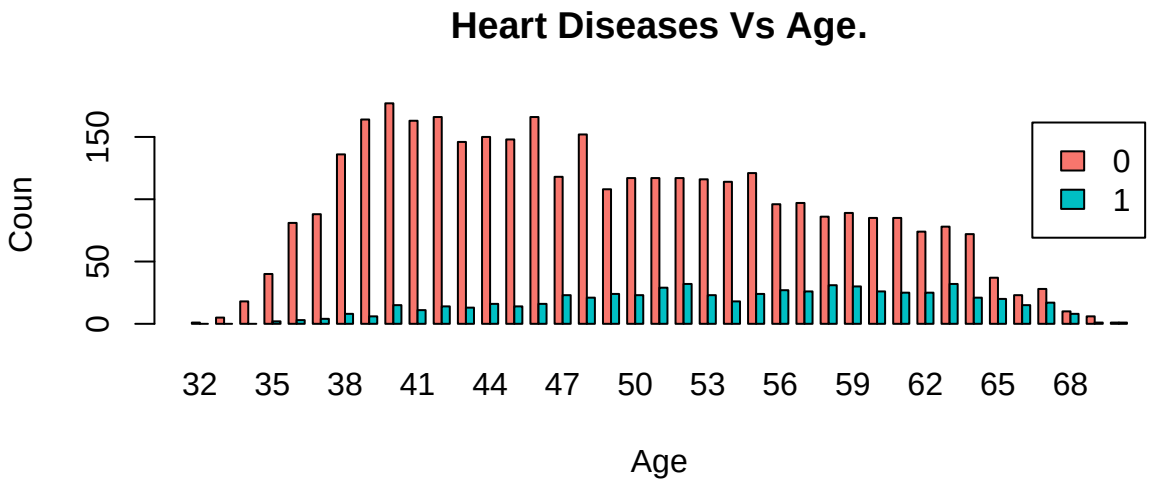
To select which feature will be useful, we plotted a correlation heat map. Columns *currentSmoker* and *cigsPrDay* shows almost the same results as *education* but show some correlation with *heartRate* and *TenYearCHD* which are some important features to take into consideration for heart disease prediction, so based on the correlation map we have decided to remove *education* but keep *currentSmoker* and *cigsPrDay* cause they show slight correlation with some important features.



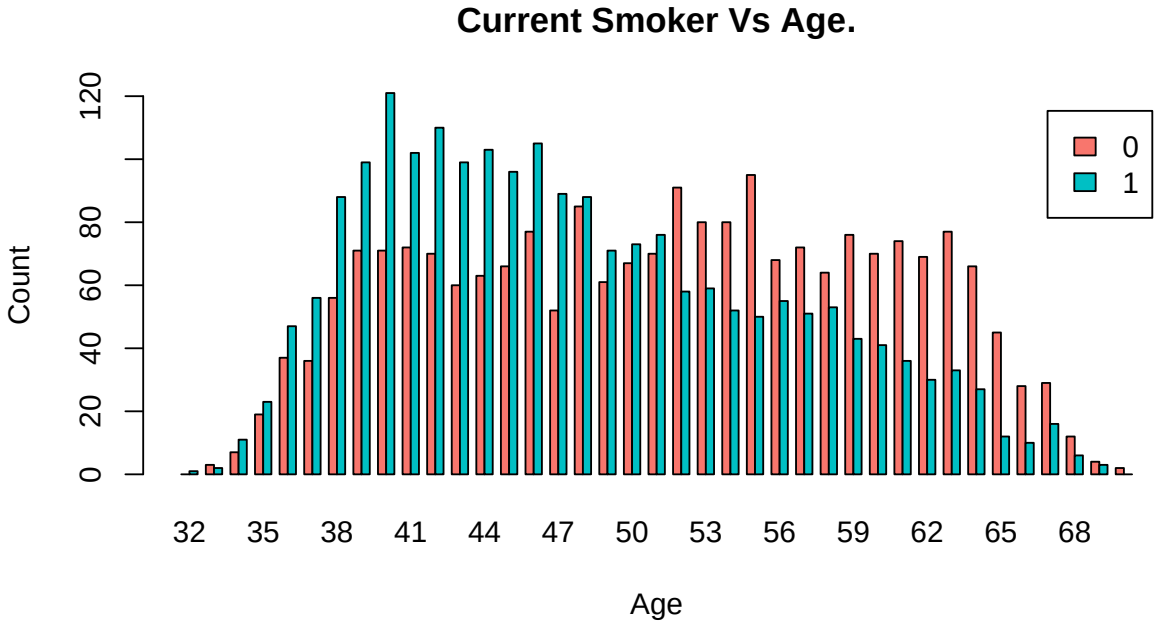
The bar plot below indicates that there are 3596 without heart disease and 644 patients with the disease



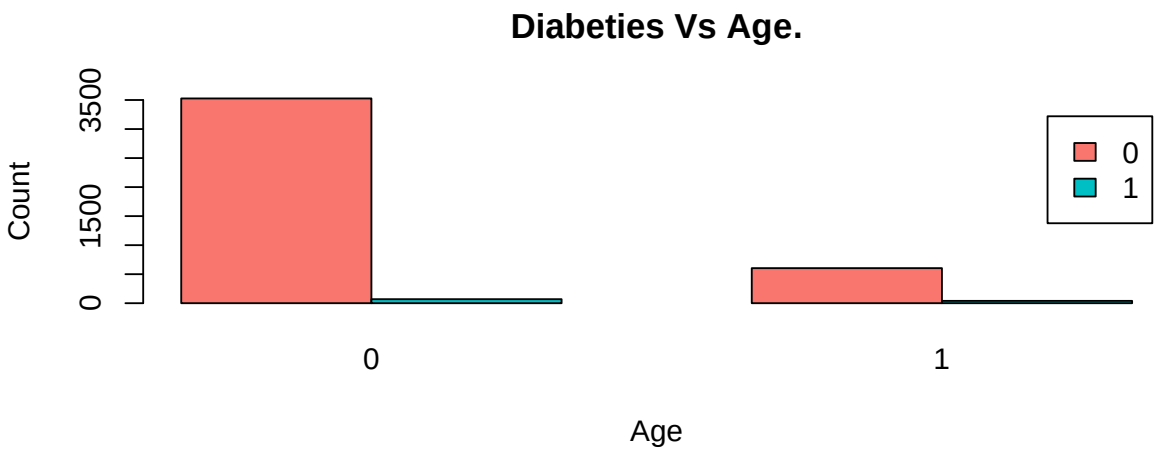
This group bar graph below visualizes number of people having heart diseases Vs their age. From the visualization we can say that heart disease rate is higher among people with no history of Coronary Heart Disease while in their late 40's and people with a history of Coronary Heart Disease have a larger distribution between their 50's and 60's.



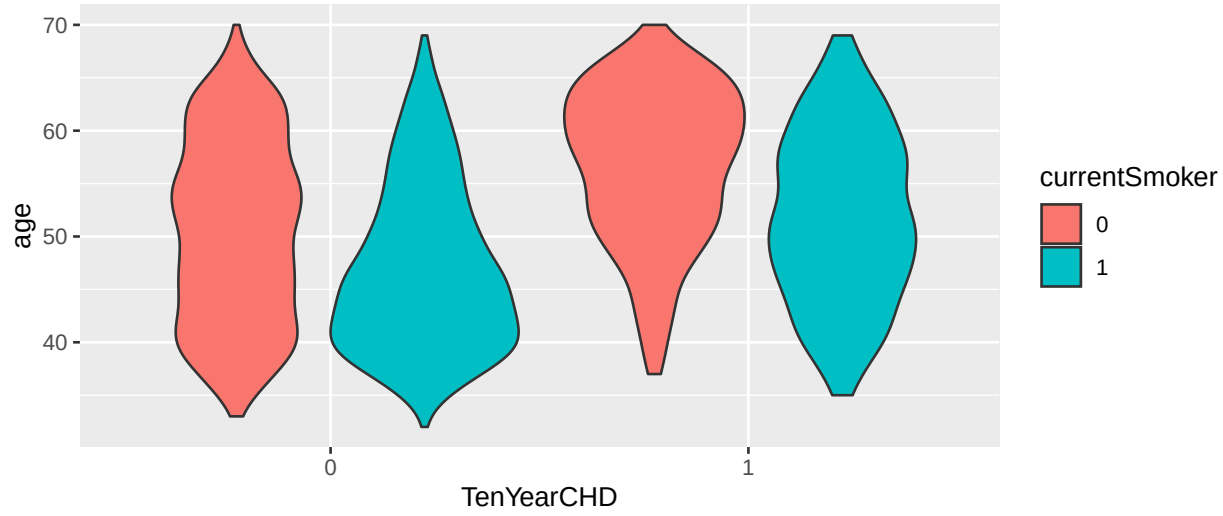
This group bar graph visualizes of people who are smokers Vs their age. As heart disease rate might be directly proportional to the smoking habits of a person we performed visualization of smokers Vs their age, we can see this same relation in the correlation map where it shows that both the smoking related features only show positive correlation towards heart rate. From the visualization we can say that people who are between the age group of late 30's and early 60's smoke a-lot.



This group bar graph visualizes of people having diabetes Vs TenYearCHD As heart disease might be directly proportional to diabetes of a person we have created visualization of people having diabetes Vs TenYearCHD. From the visualization we can say that out of 102 people who are having diabetes, 36 people are having the risk of heart disease in a 10 year period.



From the grouped violin plot below, we can say that mean age of people who smoke is less than the mean age of people who don't for people with and without heart disease. We can also see that people who are smokers having risk of heart disease are between the age of 45-60. But non-smokers having risk of heart disease are between the age of 55-65.



Milestone 3: Algorithm Testing

Logistic Regression

We build a logistic regression model using *glm* function which is used to fit generalize linear models. We first run the model for all the variables in the data-set and get a summary to see the significance of each variable, the significance is based on the p-value lower than 1. We plan on using the most significant variables in the dataset and compare each models performance on the test data-set.

```
glm.fit <- glm(TenYearCHD ~ ., data=final_cleaned, family="binomial")
summary(glm.fit)
```

```
##
## Call:
## glm(formula = TenYearCHD ~ ., family = "binomial", data = final_cleaned)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.0014  -0.5964  -0.4328  -0.2926   2.8013
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -8.178471   0.641809 -12.743  < 2e-16 ***
## male1             0.508437   0.100508   5.059 4.22e-07 ***
## age               0.062279   0.006169  10.095  < 2e-16 ***
## currentSmoker1    0.029978   0.144319   0.208 0.835449
## cigsPerDay        0.020352   0.005698   3.572 0.000354 ***
## BPMeds1           0.246817   0.218493   1.130 0.258630
## prevalentStroke1  0.966557   0.441565   2.189 0.028602 *
## prevalentHyp1     0.234885   0.128494   1.828 0.067552 .
## diabetes1         0.145056   0.295832   0.490 0.623900
## totChol           0.001787   0.001018   1.755 0.079187 .
## sysBP             0.014041   0.003538   3.969 7.23e-05 ***
## diaBP            -0.002948   0.005969  -0.494 0.621350
## BMI               0.004354   0.011721   0.371 0.710294
## heartRate        -0.001627   0.003882  -0.419 0.675104
## glucose           0.007077   0.002136   3.314 0.000921 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 3612.2  on 4239  degrees of freedom
## Residual deviance: 3209.4  on 4225  degrees of freedom
## AIC: 3239.4
##
## Number of Fisher Scoring iterations: 5
```

We then build the model using only the significant variables

```
set.seed(123)
index = createDataPartition(final_cleaned$TenYearCHD, p = 0.70, list = FALSE)
```

```
log_train = final_cleaned[index, ]
log_test = final_cleaned[-index, ]

glm.fit1 <- glm(TenYearCHD ~ male + age + cigsPerDay + prevalentStroke + sysBP + glucose, data=log_trai
summary(glm.fit1)
```

```
##
## Call:
## glm(formula = TenYearCHD ~ male + age + cigsPerDay + prevalentStroke +
##     sysBP + glucose, family = "binomial", data = log_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.0740  -0.5910  -0.4320  -0.2962   2.8121
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -8.616556   0.468418 -18.395  < 2e-16 ***
## male1             0.395767   0.117407   3.371 0.000749 ***
## age               0.066238   0.007060   9.382  < 2e-16 ***
## cigsPerDay        0.026259   0.004607   5.699 1.20e-08 ***
## prevalentStroke1  1.752644   0.618877   2.832 0.004626 **
## sysBP             0.017159   0.002376   7.221 5.16e-13 ***
## glucose           0.008077   0.002021   3.997 6.41e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2529.6  on 2968  degrees of freedom
## Residual deviance: 2239.7  on 2962  degrees of freedom
## AIC: 2253.7
##
## Number of Fisher Scoring iterations: 5
```

```
glm.probs <- predict(glm.fit1, newdata = log_test, type="response",family = binomial(link="logit"), con
```

We then calculate the accuracy of the model using the confusion matrix

```
glm.pred <- ifelse(glm.probs > 0.5, "TRUE", "FALSE")
log_cm <- table(glm.pred,log_test$TenYearCHD)
log_cm
```

```
##
## glm.pred    0    1
##    FALSE 1065  178
##    TRUE    13   15
```

```
log_acc <- (log_cm[1] + log_cm[4]) / sum(log_cm) * 100
log_acc
```

```
## [1] 84.97246
```

We got a 85% accuracy in the logistic regression model

Naive Bayes

Preparing the dataset for implementing naive bayes algorithm. The data is splitted into testing and training data for the model with the ratio being 70:30.

```
set.seed(123)
split <- sample.split(final_cleaned, SplitRatio = 0.7)
train <- subset(final_cleaned, split==TRUE)
test  <- subset(final_cleaned, split==FALSE)

x = train[-15]

y = train$TenYearCHD

y <- as.factor(y)
```

We have implemented naive bayes algorithm on the dataset and built the following model which resulted the confusion matrix and the accuracy for our dataset.

```
naive_bayes <- naiveBayes(x,y)

set.seed(123)  # Setting Seed
classifier_cl <- naiveBayes(TenYearCHD ~ ., data = train)
classifier_cl
```

```
##
## Naive Bayes Classifier for Discrete Predictors
##
## Call:
## naiveBayes.default(x = X, y = Y, laplace = laplace)
##
## A-priori probabilities:
## Y
##       0       1
## 0.85138 0.14862
##
## Conditional probabilities:
##    male
## Y           0         1
##   0 0.5881131 0.4118869
##   1 0.4833333 0.5166667
##
##    age
## Y      [,1]     [,2]
##   0 48.7315 8.470680
##   1 54.4119 7.959585
##
##    currentSmoker
## Y           0         1
##   0 0.5020781 0.4979219
##   1 0.5166667 0.4833333
##
##    cigsPerDay
```

```
## Y        [,1]     [,2]
##   0  8.952203 11.80795
##   1 10.095238 12.97916
##
##    BPMeds
## Y           0          1
##   0 0.97547797 0.02452203
##   1 0.92380952 0.07619048
##
##    prevalentStroke
## Y             0           1
##   0 0.995843724 0.004156276
##   1 0.983333333 0.016666667
##
##    prevalentHyp
## Y          0         1
##   0 0.7281796 0.2718204
##   1 0.4904762 0.5095238
##
##    diabetes
## Y           0          1
##   0 0.97838736 0.02161264
##   1 0.92142857 0.07857143
##
##    totChol
## Y       [,1]     [,2]
##   0 235.5657 43.43189
##   1 245.6810 45.80834
##
##    sysBP
## Y       [,1]     [,2]
##   0 130.1793 20.48570
##   1 144.5452 26.50069
##
##    diaBP
## Y       [,1]     [,2]
##   0 82.14048 11.32191
##   1 87.18214 13.84153
##
##    BMI
## Y       [,1]     [,2]
##   0 25.67771 4.022401
##   1 26.48790 4.477938
##
##    heartRate
## Y       [,1]     [,2]
##   0 75.76392 11.94119
##   1 76.37619 12.40434
##
##    glucose
## Y       [,1]     [,2]
##   0 80.80424 19.79024
##   1 90.95476 42.95644
```

```
y_pred <- predict(classifier_cl, newdata = test)

cm <- table(test$TenYearCHD, y_pred)
cm
```

```
##    y_pred
##        0    1
##   0 1084  106
##   1  171   53
```

```
confusionMatrix(cm)
```

```
## Confusion Matrix and Statistics
##
##    y_pred
##        0    1
##   0 1084  106
##   1  171   53
##
##                Accuracy : 0.8041
##                  95% CI : (0.7824, 0.8245)
##     No Information Rate : 0.8876
##     P-Value [Acc > NIR] : 1.0000000
##
##                   Kappa : 0.1672
##
##  Mcnemar's Test P-Value : 0.0001204
##
##             Sensitivity : 0.8637
##             Specificity : 0.3333
##          Pos Pred Value : 0.9109
##          Neg Pred Value : 0.2366
##              Prevalence : 0.8876
##          Detection Rate : 0.7666
##    Detection Prevalence : 0.8416
##       Balanced Accuracy : 0.5985
##
##        'Positive' Class : 0
##
```

The overall accuracy that we gained through this model is nearly 81%

K-means Clustering

Below we performed k-means on our data-set we need to standardize our feature for the better implementation of the Euclidian distance in the algorithm

```
kmeans_data <- final_cleaned
normalize <- function(x) {
  return((x-min(x))/ (max(x) - min(x)))
}
```

```
kmeans_data$age <- normalize(kmeans_data$age)
kmeans_data$cigsPerDay <- normalize(kmeans_data$cigsPerDay)
kmeans_data$totChol <- normalize(kmeans_data$totChol)
kmeans_data$sysBP <- normalize(kmeans_data$sysBP)
kmeans_data$diaBP <- normalize(kmeans_data$diaBP)
kmeans_data$BMI <- normalize(kmeans_data$BMI)
kmeans_data$heartRate <- normalize(kmeans_data$heartRate)
kmeans_data$glucose <- normalize(kmeans_data$glucose)
```

We build the K-Means Model using the kmeans() function and we have assigned K as 2, since we know our data-set consist of only 2 predictors i.e either a patient has a heart disese or he doesn't.

```
set.seed(240) # Setting seed
kmeans.re <- kmeans(kmeans_data, centers = 2, nstart = 20)
```

Computing the confusion matrix for the model

```
km_cm <- table(kmeans_data$TenYearCHD, kmeans.re$cluster)
km_cm
```

```
##
##        1    2
##   0 1834 1762
##   1  311  333
```

```
kmeans_acc = sum(diag(km_cm))/sum(km_cm) * 100
kmeans_acc
```

```
## [1] 51.10849
```

The accuracy gained from k-means clustering is nearly 51%

Milestone 4: Core Algorithm Tuning

Support Vector Machine

One of the efficient algorithm that we would like to use on our dataset is SVM. Here the data points are separated by hyperplane with maximum margin.For understanding model performance, we will be splitting our data in the ratio of 70:30 for training and testing purposes.

```
set.seed(123) # original dataset
split = sample.split(final_cleaned, SplitRatio = 0.70);
train_1 = subset(final_cleaned, split == TRUE);
test_1 = subset(final_cleaned, split == FALSE);
```

Feature scaling is done for our training and testing data so that all the features are normalized and contribute proportionally to the final result.

```
cols <- c("cigsPerDay","totChol","diaBP","BMI","heartRate","glucose")
levels(train_1$TenYearCHD) <- c("no","yes")
levels(test_1$TenYearCHD) <- c("no","yes")
```

SVM model will be implemented by importing SVM module, passing radial kernel (for creating support vector classifier object) in svm() function. And we can fit the training data in our model and further evaluate it.

```
ctrl <- trainControl(method="repeatedcv", repeats=5, summaryFunction=twoClassSummary, classProbs=TRUE)

svm.tune <- train(TenYearCHD~.,data = train_1,method = 'svmRadial',truelength = 9,preProc = c('center',

grid <- expand.grid(sigma = c(0.057,0.068,0.077),C = c(0.75, 0.9, 1, 1.1, 1.25))

svm.tuneRD <- train(TenYearCHD~.,data = train_1,method = 'svmRadial',truelength = 9,preProc = c('center

msvm1<- svm(formula = train_1$TenYearCHD ~ .,
         data = train_1,
         type = 'C-classification',
         kernel = 'radial');
msvm1
```

```
##
## Call:
## svm(formula = train_1$TenYearCHD ~ ., data = train_1, type = "C-classification",
##     kernel = "radial")
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  radial
##        cost:  1
##
## Number of Support Vectors:  1070
```

By comparing tune parameters we set the cost, sigma, and predicted values of the test and calculate the accuracy.

```
pred_svm1 <- predict(msvm1, test_1)
confusionMatrix(table(pred_svm1, test_1$TenYearCHD))
```

```
## Confusion Matrix and Statistics
##
##
## pred_svm1   no  yes
##       no  1189  223
##       yes    1    1
##
##                Accuracy : 0.8416
##                  95% CI : (0.8215, 0.8602)
##     No Information Rate : 0.8416
##     P-Value [Acc > NIR] : 0.5178
##
##                   Kappa : 0.0061
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.999160
##             Specificity : 0.004464
##          Pos Pred Value : 0.842068
##          Neg Pred Value : 0.500000
##              Prevalence : 0.841584
##          Detection Rate : 0.840877
##    Detection Prevalence : 0.998586
##       Balanced Accuracy : 0.501812
##
##        'Positive' Class : no
##
```

The overall accuracy that we gained through this model is nearly 84%

Neural Networks

Data Preprocessing

Since we will be using NN, all our attributes need to be numeric, so we converted all attributes to numeric, we further use min-max normalization on our data for it to better fit model. Once normalization is done we have split the data into training and testing in 70% to 30% ratio.

```
nn_data <- final_cleaned
nn_data$male = as.numeric(nn_data$male)
nn_data$currentSmoker = as.numeric(nn_data$currentSmoker)
nn_data$BPMeds = as.numeric(nn_data$BPMeds)
nn_data$prevalentStroke = as.numeric(nn_data$prevalentStroke)
nn_data$prevalentHyp = as.numeric(nn_data$prevalentHyp)
nn_data$diabetes = as.numeric(nn_data$diabetes)
nn_data$glucose = as.numeric(nn_data$glucose)
nn_data$heartRate = as.numeric(nn_data$heartRate)
nn_data$totChol = as.numeric(nn_data$totChol)
nn_data$cigsPerDay = as.numeric(nn_data$cigsPerDay)
```

```
nn_data$age = as.numeric(nn_data$age)

maxs <- apply(nn_data[0:13], 2, max)
mins <- apply(nn_data[0:13], 2, min)

nn_data[1:13] <- as.data.frame(scale(nn_data[1:13], center = mins, scale = maxs - mins))

split = sample.split(nn_data$TenYearCHD, SplitRatio = 0.70)
train = subset(nn_data, split==TRUE)
test = subset(nn_data, split==FALSE)
```

Training

For training we use a simple feed-forward network having 15 neurons in the input layer, 2 neurons in the output layer and one hidden layer containing 8 neurons. We have used back-propagation algorithm to update our weights with a learning rate of 0.005, the error function used is cross-entropy loss, activation function used is sigmoid, we took the default step-size as the weights weren't converging when used a lower max-step size.

```
nn <- neuralnet(formula = TenYearCHD~.,data=train,algorithm = "backprop", learningrate = 0.05,err.fct =
```

Summary of the feed-forward network

```
summary(nn)
tvals = test[,14]
mypredict <- compute(nn, test[1:13])$net.result

results <- data.frame(actual = test$TenYearCHD, prediction = mypredict)

mypredict <- data.frame("mypredict"=ifelse(max.col(mypredict[ ,1:2])==1, '1','2'))
```

The package used to implement the neural networks hasn't been updated for a while and it has some unresolved issues which arise when the parameters as set to a particular value.

Validation

In oder to validate the performance of the classifiers we used the following statistical parameters:

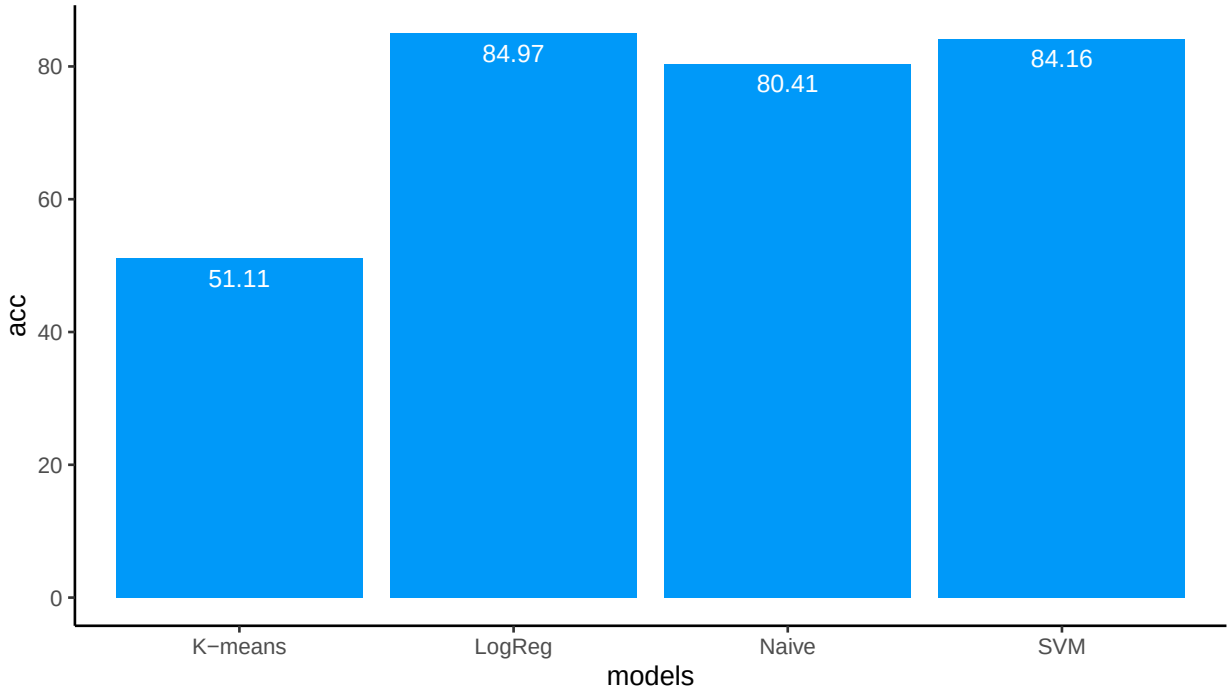
Accuracy: The accuracy is a measure of correctly classified instances in the dataset.

Precision: The precision is a measure of the model's ability to correctly predict the positives out of all the positive prediction it made.

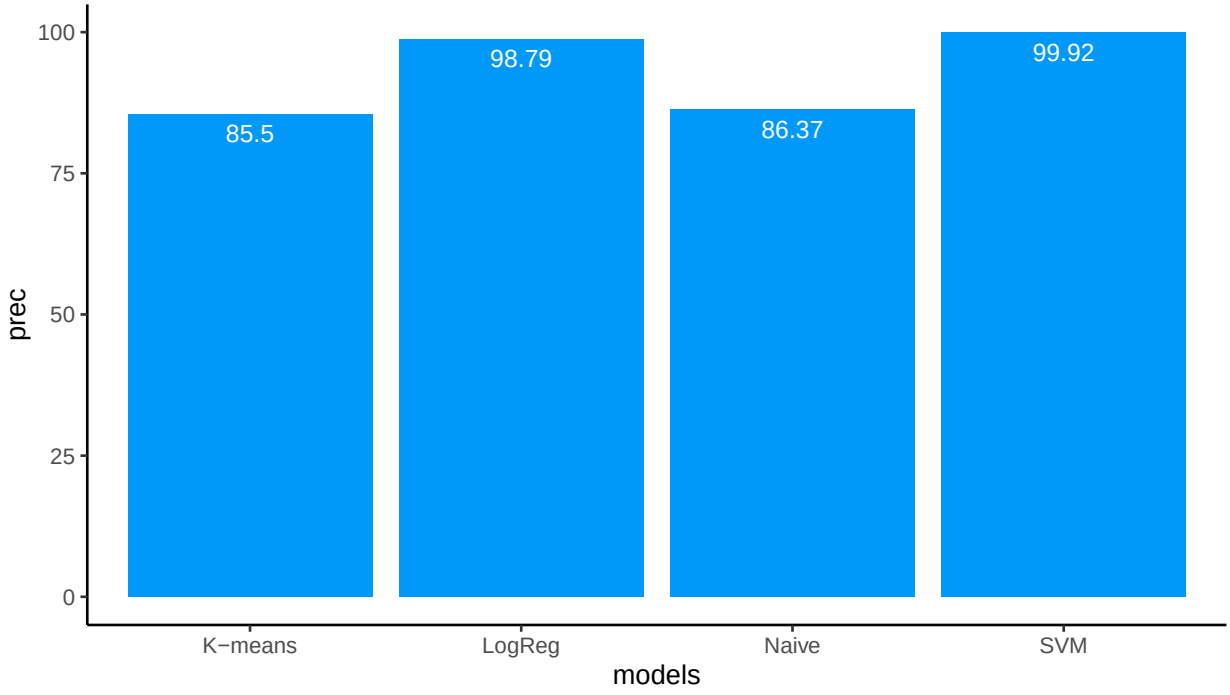
Recall: The recall is a measure of the model's ability to correctly predict the positives out of actual positives.

F1-score: The f1-score is a measure of the model's accuracy on the dataset.

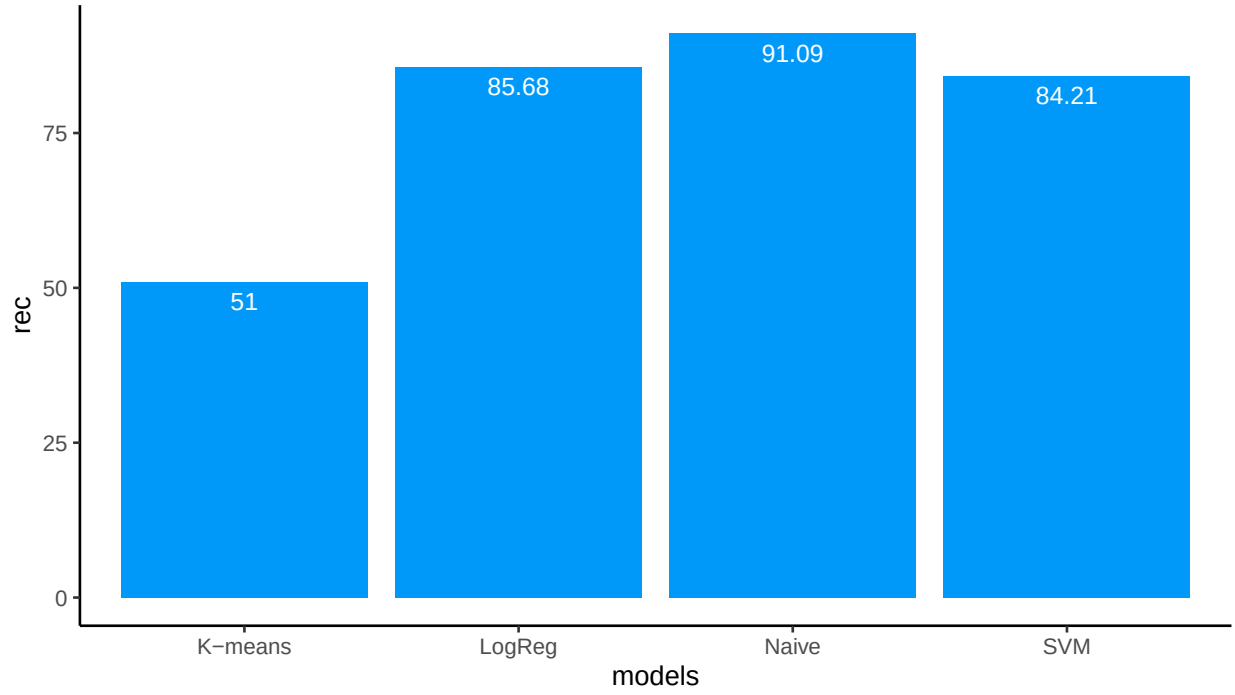
The graph below shows each models accuracy



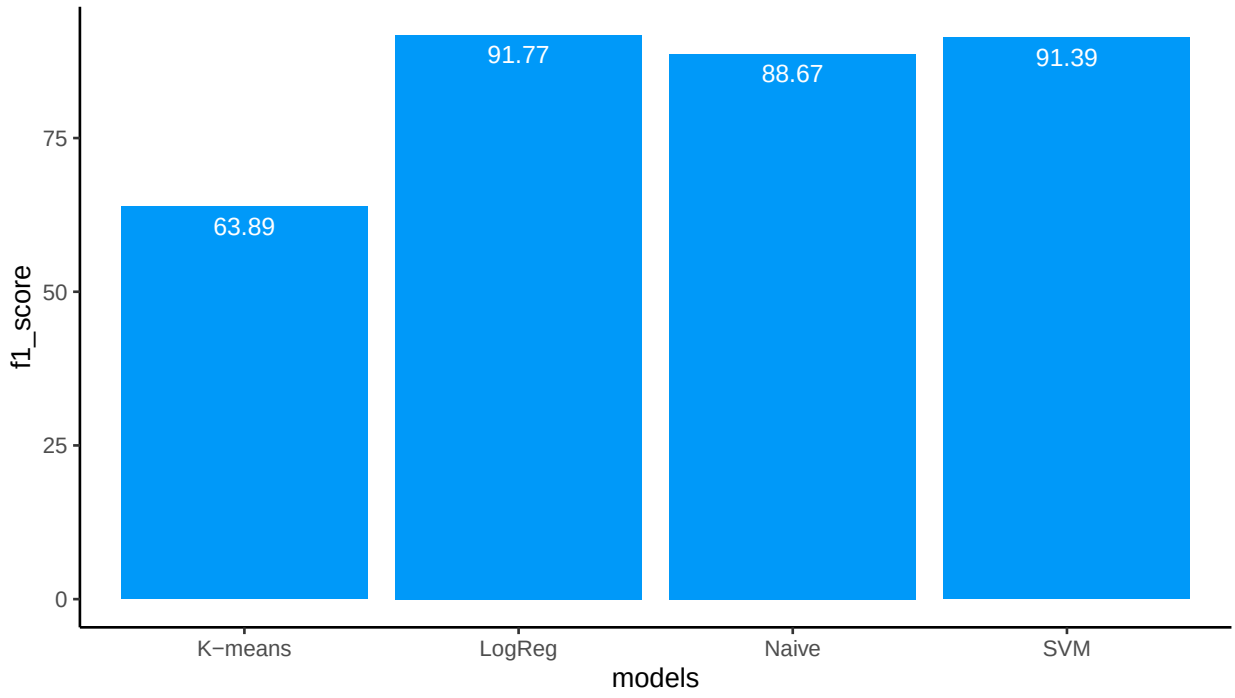
The graph below shows each models precision



The graph below shows each models recall



The graph below shows each models f1-score



The above graphs represent the accuracy, precision, recall, and f1-score for each model. Since Naive Bayes has the highest recall it can be considered a good model to predict CHD risk among individuals. From the f1 score we can say that logistic regression is the best model for the data-set.

Conclusion

We have implemented Logistic Regression, Naive Bayes, K-Means and Support Vector Machine.From these model, we got nearly of 85% accuracy in the Logistic regression model. For Naive Bayes, we got 81% accuracy, not making it better than the Regression model. For K-Means Algorithm, the accuracy is nearly 51%, it was discovered that clustering is not suitable for this dataset. Finally for Support Vector Machine, the results showed an accuracy of 84%, therefore making Logistic Regression the best model for this data-set.

Future Work

The framingham heart disease dataset consists of only 4240 instances and 16 features. The data present in this dataset is very unbalance and size of the data isn't comprehensive enough. In order to get a more generalized classification and prediction accuracy from these models we need to have access to a dataset with more correlated features. This will help in developing more efficient machine learning models to predict heart disease. Further we can try a more hybrid approach where we can use a based model to select the important features and then use those features in other models for classification.

Appendix—Code

```
knitr::opts_chunk$set(echo= TRUE, warning=FALSE, message=FALSE)
library(kableExtra)
library(plyr)
library(tidyverse)
library(gmodels)
library(ggmosaic)
library(corrplot)
library(mice)
library(caret)
library(rpart)
library(cluster)
library(fpc)
library(data.table)
library(knitr)
library(naniar)
library(visdat)
library(Hmisc)
library(shiny)
library(caTools)
library(corrplot)
library(e1071)
library(naivebayes)
library(neuralnet)
library(GGally)
library(pROC)
framingham <- read.csv("framingham.csv", header = T)
ncol(framingham)
```

```
nrow(framingham)
str(framingham)
summary(framingham)
framingham$male = as.factor(framingham$male)
framingham$education = as.factor(framingham$education)
framingham$currentSmoker = as.factor(framingham$currentSmoker)
framingham$BPMeds = as.factor(framingham$BPMeds)
framingham$prevalentStroke = as.factor(framingham$prevalentStroke)
framingham$prevalentHyp = as.factor(framingham$prevalentHyp)
framingham$diabetes = as.factor(framingham$diabetes)
framingham$glucose = as.numeric(framingham$glucose)
framingham$heartRate = as.numeric(framingham$heartRate)
framingham$totChol = as.numeric(framingham$totChol)
framingham$cigsPerDay = as.numeric(framingham$cigsPerDay)
framingham$age = as.numeric(framingham$age)
framingham$TenYearCHD = as.factor(framingham$TenYearCHD)
framingham_new <- framingham[,-c(3)]
vis_miss(framingham_new, cluster = TRUE, sort_miss = TRUE)
new_dataset <- mice(framingham_new, m = 5, method = c("", "", "", "pmm", "logreg", "", "", "", "pmm", "
final_cleaned <- mice::complete(new_dataset, 2)
one_fram <- final_cleaned
vis_fram <- data.frame(one_fram)

df2 = as.data.frame(sapply(vis_fram, as.integer))

ggplot(gather(df2), aes(value)) +
    geom_histogram(bins = 10, fill="#00bfc4",
  stat = "count") +
    facet_wrap(~key, scales = 'free_x')
boxplot(final_cleaned,col='#00bfc4')
cor_martix <- cor(df2)
corrplot(cor_martix, method="color", tl.col = "black")
vis_fram %>% ggplot(aes(x=TenYearCHD, fill = TenYearCHD)) + geom_bar(width = 0.5, position = position_d
counts <- table(vis_fram$TenYearCHD, vis_fram$age)
barplot(counts, main="Heart Diseases Vs Age.",
  xlab="Age", ylab="Coun", col=c("#F8766D","#00bfc4"),
  legend = rownames(counts), beside=TRUE)
counts <- table(vis_fram$currentSmoker, vis_fram$age)
barplot(counts, main="Current Smoker Vs Age.",
  xlab="Age", ylab="Count", col=c("#F8766D","#00bfc4"),
  legend = rownames(counts), beside=TRUE)
counts <- table(vis_fram$diabetes, vis_fram$TenYearCHD)
barplot(counts, main="Diabeties Vs Age.",
  xlab="Age", ylab="Count", col=c("#F8766D","#00bfc4"),
  legend = rownames(counts), beside=TRUE)
p <- ggplot(final_cleaned, aes(x=TenYearCHD, y=age, fill=currentSmoker)) + # fill=name allow to automat
  geom_violin()
p
glm.fit <- glm(TenYearCHD ~ ., data=final_cleaned, family="binomial")
summary(glm.fit)
set.seed(123)
index = createDataPartition(final_cleaned$TenYearCHD, p = 0.70, list = FALSE)
log_train = final_cleaned[index, ]
```

```
log_test = final_cleaned[-index, ]

glm.fit1 <- glm(TenYearCHD ~ male + age + cigsPerDay + prevalentStroke + sysBP + glucose, data=log_trai
summary(glm.fit1)

glm.probs <- predict(glm.fit1, newdata = log_test, type="response",family = binomial(link="logit"), con
glm.pred <- ifelse(glm.probs > 0.5, "TRUE", "FALSE")
log_cm <- table(glm.pred,log_test$TenYearCHD)
log_cm

log_acc <- (log_cm[1] + log_cm[4]) / sum(log_cm) * 100
log_acc
set.seed(123)
split <- sample.split(final_cleaned, SplitRatio = 0.7)
train <- subset(final_cleaned, split==TRUE)
test  <- subset(final_cleaned, split==FALSE)

x = train[-15]

y = train$TenYearCHD

y <- as.factor(y)
naive_bayes <- naiveBayes(x,y)

set.seed(123)  # Setting Seed
classifier_cl <- naiveBayes(TenYearCHD ~ ., data = train)
classifier_cl

y_pred <- predict(classifier_cl, newdata = test)

cm <- table(test$TenYearCHD, y_pred)
cm
confusionMatrix(cm)
kmeans_data <- final_cleaned
normalize <- function(x) {
  return((x-min(x))/ (max(x) - min(x)))
}

kmeans_data$age <- normalize(kmeans_data$age)
kmeans_data$cigsPerDay <- normalize(kmeans_data$cigsPerDay)
kmeans_data$totChol <- normalize(kmeans_data$totChol)
kmeans_data$sysBP <- normalize(kmeans_data$sysBP)
kmeans_data$diaBP <- normalize(kmeans_data$diaBP)
kmeans_data$BMI <- normalize(kmeans_data$BMI)
kmeans_data$heartRate <- normalize(kmeans_data$heartRate)
kmeans_data$glucose <- normalize(kmeans_data$glucose)
set.seed(240) # Setting seed
kmeans.re <- kmeans(kmeans_data, centers = 2, nstart = 20)
km_cm <- table(kmeans_data$TenYearCHD, kmeans.re$cluster)
km_cm
kmeans_acc = sum(diag(km_cm))/sum(km_cm) * 100
kmeans_acc
set.seed(123) # original dataset
```

```
split = sample.split(final_cleaned, SplitRatio = 0.70);
train_1 = subset(final_cleaned, split == TRUE);
test_1 = subset(final_cleaned, split == FALSE);
cols <- c("cigsPerDay","totChol","diaBP","BMI","heartRate","glucose")
levels(train_1$TenYearCHD) <- c("no","yes")
levels(test_1$TenYearCHD) <- c("no","yes")
ctrl <- trainControl(method="repeatedcv", repeats=5, summaryFunction=twoClassSummary, classProbs=TRUE)

svm.tune <- train(TenYearCHD~.,data = train_1,method = 'svmRadial',truelength = 9,preProc = c('center',

grid <- expand.grid(sigma = c(0.057,0.068,0.077),C = c(0.75, 0.9, 1, 1.1, 1.25))

svm.tuneRD <- train(TenYearCHD~.,data = train_1,method = 'svmRadial',truelength = 9,preProc = c('center

msvm1<- svm(formula = train_1$TenYearCHD ~ .,
         data = train_1,
         type = 'C-classification',
         kernel = 'radial');
msvm1
pred_svm1 <- predict(msvm1, test_1)
confusionMatrix(table(pred_svm1, test_1$TenYearCHD))
nn_data <- final_cleaned
nn_data$male = as.numeric(nn_data$male)
nn_data$currentSmoker = as.numeric(nn_data$currentSmoker)
nn_data$BPMeds = as.numeric(nn_data$BPMeds)
nn_data$prevalentStroke = as.numeric(nn_data$prevalentStroke)
nn_data$prevalentHyp = as.numeric(nn_data$prevalentHyp)
nn_data$diabetes = as.numeric(nn_data$diabetes)
nn_data$glucose = as.numeric(nn_data$glucose)
nn_data$heartRate = as.numeric(nn_data$heartRate)
nn_data$totChol = as.numeric(nn_data$totChol)
nn_data$cigsPerDay = as.numeric(nn_data$cigsPerDay)
nn_data$age = as.numeric(nn_data$age)

maxs <- apply(nn_data[0:13], 2, max)
mins <- apply(nn_data[0:13], 2, min)

nn_data[1:13] <- as.data.frame(scale(nn_data[1:13], center = mins, scale = maxs - mins))

split = sample.split(nn_data$TenYearCHD, SplitRatio = 0.70)
train = subset(nn_data, split==TRUE)
test = subset(nn_data, split==FALSE)
nn <- neuralnet(formula = TenYearCHD~.,data=train,algorithm = "backprop", learningrate = 0.05,err.fct =
summary(nn)
tvals = test[,14]
mypredict <- compute(nn, test[1:13])$net.result

results <- data.frame(actual = test$TenYearCHD, prediction = mypredict)

mypredict <- data.frame("mypredict"=ifelse(max.col(mypredict[ ,1:2])==1, '1','2'))
svm_cm <- table(pred_svm1, test_1$TenYearCHD)

svm_acc <- (svm_cm[1] + svm_cm[4]) / sum(svm_cm) * 100
```

```r
log_prec <- (log_cm[1])/ (log_cm[1] + log_cm[2]) * 100

log_rec <- (log_cm[1])/ (log_cm[1] + log_cm[3]) * 100

log_f1 <- (2 * (log_rec * log_prec))/(log_prec + log_rec)

naiv_acc <- (cm[1] + cm[4]) / sum(cm) * 100

naiv_prec <- (cm[1])/ (cm[1] + cm[2]) * 100

naiv_rec <- (cm[1])/ (cm[1] + cm[3]) * 100

naiv_f1 <- (2 * (naiv_rec * naiv_prec))/(naiv_prec + naiv_rec)

km_prec <- (km_cm[1])/ (km_cm[1] + km_cm[2]) * 100

km_rec <- (km_cm[1])/ (km_cm[1] + km_cm[3]) * 100

km_f1 <- (2 * (km_rec * km_prec))/(km_prec + km_rec)
svm_cm <- table(pred_svm1, test_1$TenYearCHD)

svm_prec <- (svm_cm[1])/ (svm_cm[1] + svm_cm[2]) * 100
svm_rec <- (svm_cm[1])/ (svm_cm[1] + svm_cm[3]) * 100
svm_f1 <- (2 * (svm_rec * svm_prec))/(svm_prec + svm_rec)
acc_data <- data.frame(
  models=c("LogReg", "Naive", "K-means", "SVM"),
  acc=round(c(log_acc,naiv_acc, kmeans_acc, svm_acc), digits = 2)
)

acc_plot<- ggplot(acc_data, aes(x = models, y = acc)) + geom_col(fill = "#0099f9")+ theme_classic()+ ge
acc_plot
prec_data <- data.frame(
  models=c("LogReg", "Naive", "K-means", "SVM"),
  prec=round(c(log_prec,naiv_prec, km_prec, svm_prec), digits = 2)
)

prec_plot<- ggplot(prec_data, aes(x = models, y = prec)) + geom_col(fill = "#0099f9")+ theme_classic()+
prec_plot
rec_data <- data.frame(
  models=c("LogReg", "Naive", "K-means", "SVM"),
  rec=round(c(log_rec,naiv_rec, km_rec, svm_rec), digits = 2)
)

rec_plot<- ggplot(rec_data, aes(x = models, y = rec)) + geom_col(fill = "#0099f9")+
  theme_classic()+ geom_text(aes(label=rec), vjust=1.6, color="white", size=3.5)
rec_plot
f1_data <- data.frame(
  models=c("LogReg", "Naive", "K-means", "SVM"),
  f1_score=round(c(log_f1,naiv_f1, km_f1, svm_f1), digits=2)
)

f1_plot<- ggplot(f1_data, aes(x = models, y = f1_score)) + geom_col(fill = "#0099f9")+ theme_classic()+
f1_plot
```

References

Kim, Jaekwon, Jongsik Lee, and Youngho Lee. 2015. "Data-Mining-Based Coronary Heart Disease Risk Prediction Model Using Fuzzy Logic and Decision Tree." *Healthcare Informatics Research* 21 (3): 167–74.

Tasnim, Farzana, and Sultana Umme Habiba. 2021. "A Comparative Study on Heart Disease Prediction Using Data Mining Techniques and Feature Selection," 338–41.