# CASE STUDY REPORT

**Coronary Heart Disease Classification Problem**

**Group No.**: 11

**Student Names**: Kush Adhvaryu and Parth Modhia

## Executive Summary:

Cardiovascular disease is one of the most significant reasons for the number of deaths among all people around the world. The early forecast of cardiovascular ailments can help in compelling high-risk patients to change their risky habits and follow a healthy routine such that any unfavorable events can be prevented.

The goal of this study was to forecast if a patient would get coronary heart disease in a 10-year span or not according to the traits he possesses.

The dataset is publicly accessible on the Kaggle website, and it is from an in progress cardiovascular examination, on the inhabitants of the town of Framingham, Massachusetts. The dataset has over 4000 records and 15 features or attributes. Each attribute is a potential risk factor. There are both demographic, behavioral, and medical risk factors.

While exploring our data, we found that the total number of rows with missing values were 582 which is only 13 percent of the entire dataset, so we excluded them.

As we have a classification problem, we have applied kNN, Naive Bayes, Random Forest, Logistic Regression, Neural Net, Linear Discriminant Analysis and Support Vector Machine and then evaluated the performances of each model.

We formulated 7 different supervised learning models and created confusion matrices, Lift Charts/ ROC curves and found out accuracies for all of them. Men appear to be more vulnerable to coronary illness than Women. An expansion in age, number of cigarettes smoked every day and systolic Blood Pressure additionally show expanded chances of having coronary illness.

Out of all the 7 supervised learning models, our KNN model is most accurate with an accuracy of 86%.

## I. Background and Introduction

Cardiovascular diseases are probably the most significant reason for the number of deaths among all people around the world. The prediction of cardiovascular diseases is viewed as one of the most significant subjects in the field of data analytics under the healthcare domain. World Health Organization has evaluated 12 million demises happening around the world, consistently because of heart related problems. Around 610,000 individuals fall at the hands of coronary illness in the United States each year–that is 1 in every four demises; that is, one person dies every 37 seconds due to it.

Coronary illness portrays a scope of conditions that influences the heart. Sicknesses under the coronary illness umbrella comprises of blood vessels ailments, heart rhythm issues (arrhythmias); and heart absconds you're brought into the world with (congenital heart defects), etc.

The expression "coronary illness" is frequently utilized conversely with the expression "cardiovascular diseases." Cardiovascular sickness by and large alludes to conditions that include restricted or blocked vessels that can prompt a respiratory failure, chest torment (angina) or stroke. Other heart conditions, for example, those that influence your heart's muscle, valves or rhythm, likewise are viewed as types of coronary illness. Numerous types of coronary illness can be prevented or treated by following and implementing solid & healthy lifestyle decisions. The early forecast of cardiovascular ailments can help in compelling high-risk patients to change their risky habits and follow a healthy routine such that any unfavorable events can be prevented.

Our study means to pinpoint the most pertinent/hazardous elements involved in the cardiovascular diseases and to anticipate the general hazard. The classification goal is to forecast if a patient would get coronary heart disease in a 10-year span.

## II. Data Exploration and Visualization

We utilized different visualization techniques to understand data. We checked the multi-collinearity for linear models.
In the end, the model is intended to pinpoint the most relevant/risk factors of heart disease as well as predict the overall risk. It’s a binary classification problem.
Predictors are Age, Glucose level, Gender, Education, Current Smoker, Cigarette per day, Blood Pressure Medications, Prevalent Stroke, Prevalent Hypertension, Diabetes, Total Cholesterol, Systolic Blood Pressure, Diastolic Blood Pressure, Body Mass Index & Heart rate.The Outcome variable is a binary variable, the 10-year risk of coronary heart disease CHD with the class either Yes or No.

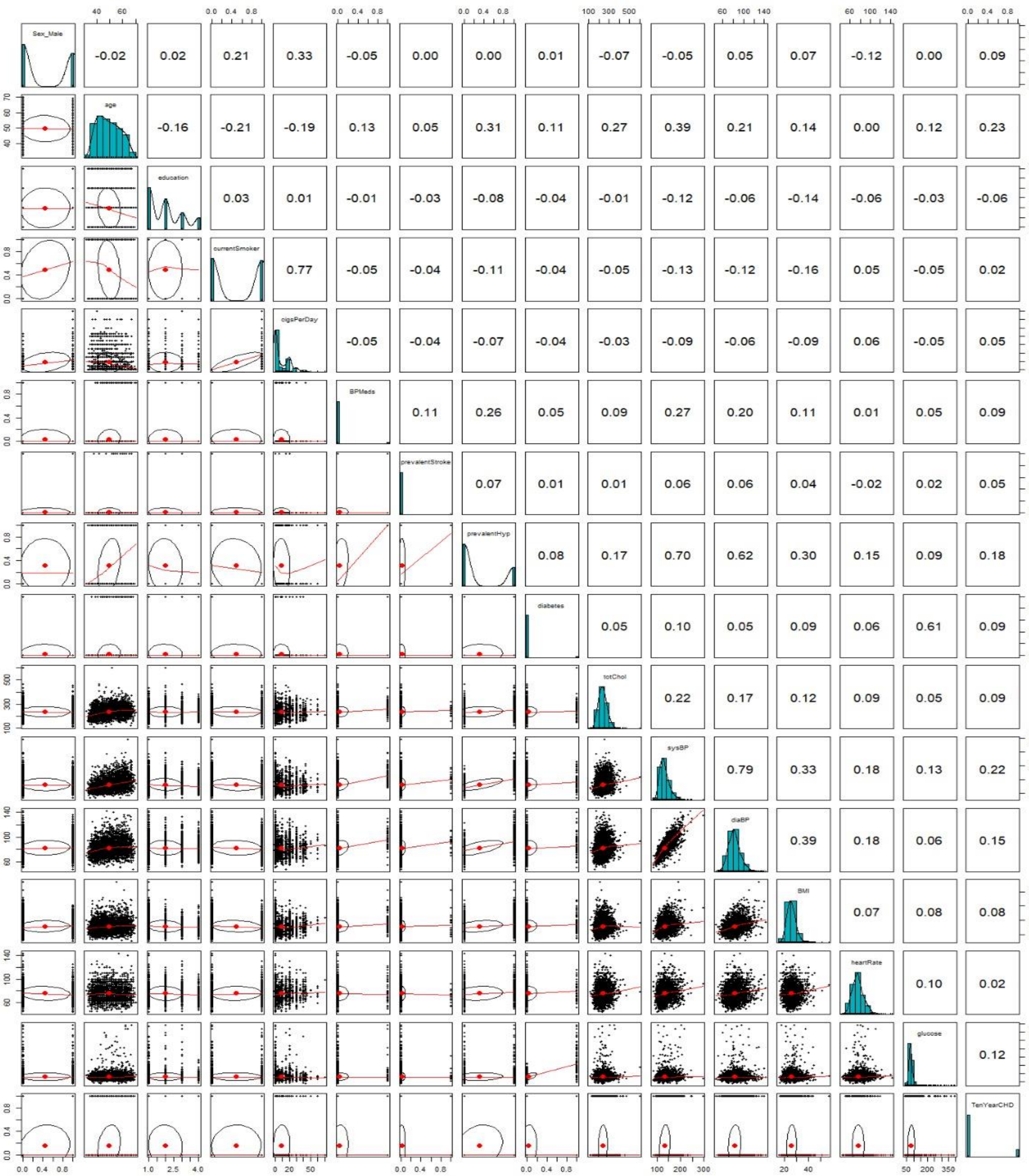



Fig.1 Correlation Plot

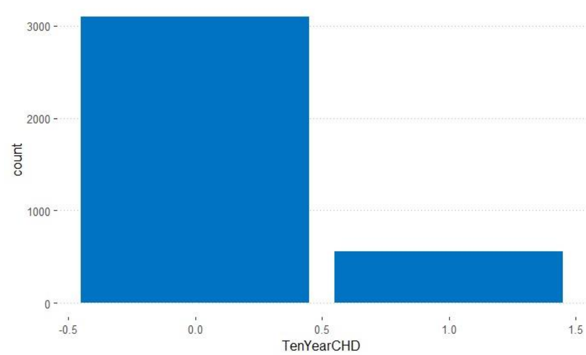



Fig.2 Count plot

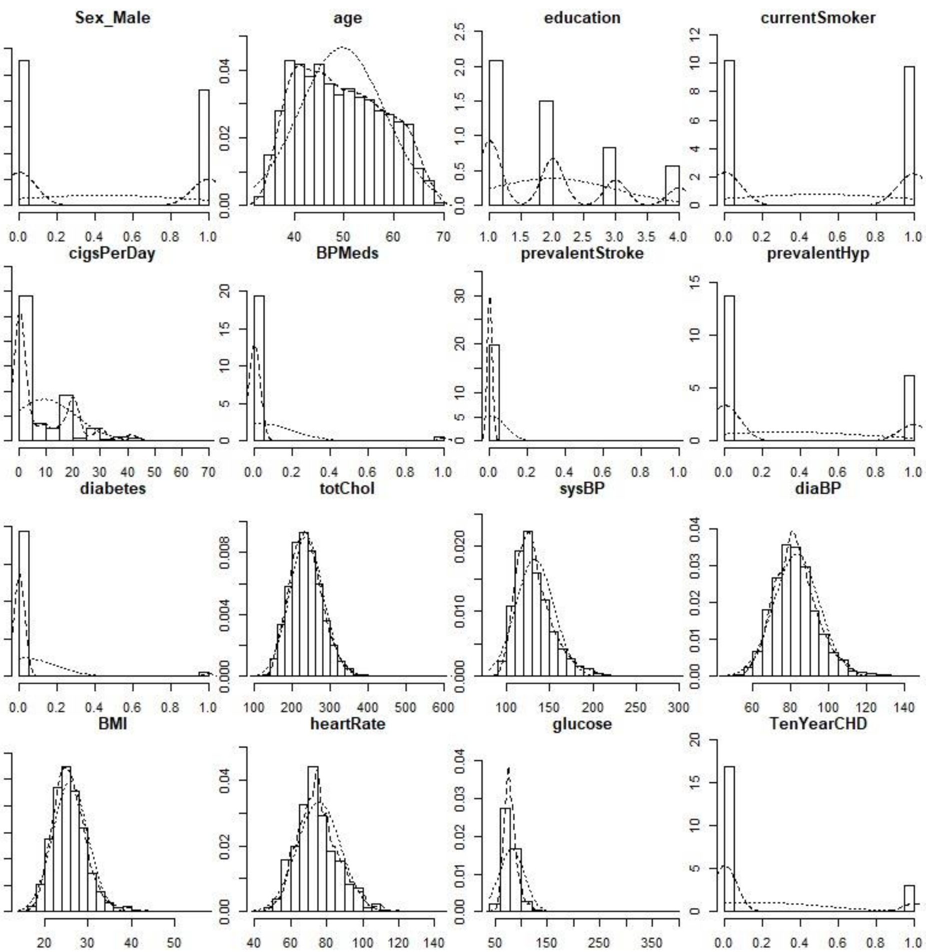



Fig.3 Histogram for all the variables in the data set.

Fig.2 shows that our data has _ patients who will not have heart disease while _ patients will have heart disease in the next 10 years.

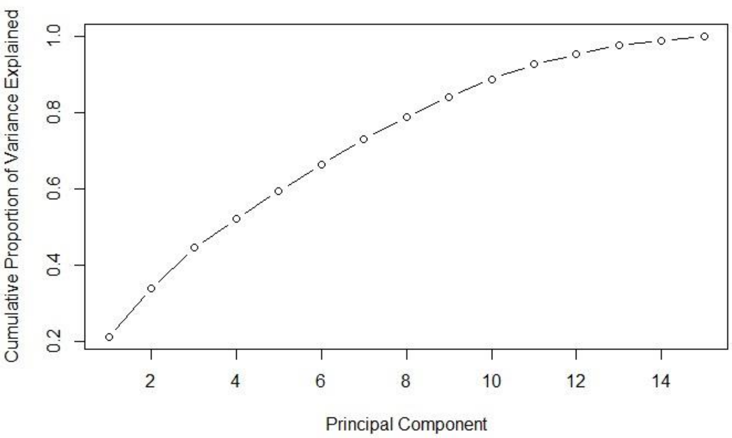
Fig.3 shows the distribution of individual features in our dataset.

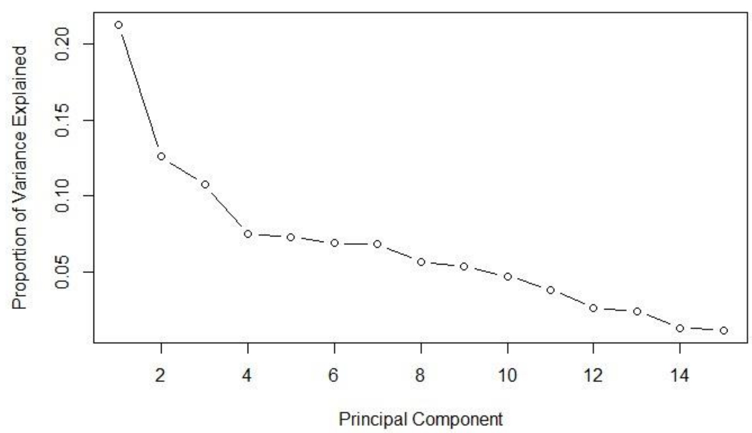
# III. Data Preparation and Preprocessing

We had checked for missing or NA values in the dataset & descriptive statistic of each variable to see if there were any outliers. We eliminated the rows which had missing values because if we had imputed the values here, I would create a bias in our features

Total number of rows with missing values was 582. Since it was only 13.72 percent of the entire dataset the rows with missing values were excluded.

```
    Sex_Male             age           education      currentSmoker        cigsPerDay         BPMeds       
 Min.   :0.0000   Min.   :32.00   Min.   :1.00   Min.   :0.0000   Min.   : 0.000   Min.   :0.00000  
 1st Qu.:0.0000   1st Qu.:42.00   1st Qu.:1.00   1st Qu.:0.0000   1st Qu.: 0.000   1st Qu.:0.00000  
 Median :0.0000   Median :49.00   Median :2.00   Median :0.0000   Median : 0.000   Median :0.00000  
 Mean   :0.4437   Mean   :49.55   Mean   :1.98   Mean   :0.4891   Mean   : 9.025   Mean   :0.03034  
 3rd Qu.:1.0000   3rd Qu.:56.00   3rd Qu.:3.00   3rd Qu.:1.0000   3rd Qu.:20.000   3rd Qu.:0.00000  
 Max.   :1.0000   Max.   :70.00   Max.   :4.00   Max.   :1.0000   Max.   :70.000   Max.   :1.00000  
 prevalentStroke      prevalentHyp       diabetes           totChol          sysBP            diaBP       
 Min.   :0.000000   Min.   :0.0000   Min.   :0.00000   Min.   :113.0   Min.   : 83.5   Min.   : 48.00  
 1st Qu.:0.000000   1st Qu.:0.0000   1st Qu.:0.00000   1st Qu.:206.0   1st Qu.:117.0   1st Qu.: 75.00  
 Median :0.000000   Median :0.0000   Median :0.00000   Median :234.0   Median :128.0   Median : 82.00  
 Mean   :0.005741   Mean   :0.3116   Mean   :0.02706   Mean   :236.8   Mean   :132.4   Mean   : 82.92  
 3rd Qu.:0.000000   3rd Qu.:1.0000   3rd Qu.:0.00000   3rd Qu.:263.0   3rd Qu.:143.9   3rd Qu.: 90.00  
 Max.   :1.000000   Max.   :1.0000   Max.   :1.00000   Max.   :600.0   Max.   :295.0   Max.   :142.50  
      BMI           heartRate          glucose          TenYearCHD    
 Min.   :15.54   Min.   : 44.00   Min.   : 40.00   Min.   :0.0000  
 1st Qu.:23.08   1st Qu.: 68.00   1st Qu.: 71.00   1st Qu.:0.0000  
 Median :25.38   Median : 75.00   Median : 78.00   Median :0.0000  
 Mean   :25.78   Mean   : 75.73   Mean   : 81.85   Mean   :0.1523  
 3rd Qu.:28.04   3rd Qu.: 82.00   3rd Qu.: 87.00   3rd Qu.:0.0000  
 Max.   :56.80   Max.   :143.00   Max.   :394.00   Max.   :1.0000  
```




We performed Principal Component Analysis. PCA looks for properties that show as much variation across classes as possible to build the principal component space. The algorithm uses the concepts of variance matrix, covariance matrix, eigenvector and eigenvalues pairs to perform PCA, providing a set of eigenvectors and its respectively eigenvalues as a result. It is very simple; the eigenvectors represent the new set of axes of the principal component space and the eigenvalues carry the information of quantity of variance that each eigenvector have. So, in order to reduce the dimension of the dataset we are going to choose those eigenvectors that have more variance and discard those with less variance.

# IV. Data Mining Techniques and Implementation

As we have a classification problem, we have applied kNN, Naive Bayes, Random Forest, Logistic Regression, Neural Net, Linear Discriminant Analysis and Support Vector Machine. We will compare the models to find the best fit model for our problem.

### 1. K-Nearest Neighbors

RMSE was used to select the optimal model using the smallest value. The final value used for the model was k = 43. The model is classified as level 0 thus the patient will not have coronary heart disease in 10 years.

```
[1] "2205" "4036" "1554"
[1] 0
attr(,"nn.index")
     [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13] [,14] [,15] [,16] [,17] [,18] [,19]
[1,] 3157 3260 2654  228 3402 1886 3473 1348 3220  2690  2125   667  2117  2308   409   564  3380  2844  3004
     [,20] [,21] [,22] [,23] [,24] [,25] [,26] [,27] [,28] [,29] [,30] [,31] [,32] [,33] [,34] [,35] [,36] [,37]
[1,]  1749  3102  2967   406  1604  2295   272  1636  2746  3276  2369  1127  3467  1869   233  3582   851  2250
     [,38] [,39] [,40] [,41] [,42] [,43]
[1,]  1345  3048  2451  1012  2205  3627
attr(,"nn.dist")
         [,1]     [,2]     [,3]     [,4]     [,5]    [,6]     [,7]     [,8]     [,9]   [,10]    [,11]    [,12]
[1,] 6.977703 6.979081 7.248525 7.687337 7.827544 8.53705 8.629625 8.782091 8.898135 8.91105 8.911886 8.920804
        [,13]    [,14]    [,15]   [,16]    [,17]    [,18]    [,19]    [,20]    [,21]   [,22]    [,23]    [,24]
[1,] 8.930859 9.203902 9.204123 9.42979 9.457254 9.562825 9.691679 9.932137 10.10613 15.3833 15.51666 15.76507
        [,25]    [,26]    [,27]    [,28]    [,29]    [,30]    [,31]   [,32]    [,33]    [,34]    [,35]    [,36]
[1,] 15.80075 16.12166 16.12229 16.15352 16.16135 16.19502 16.19532 16.2192 16.22703 16.23609 16.25262 16.27233
        [,37]   [,38]    [,39]    [,40]    [,41]    [,42]    [,43]
[1,] 16.27513 16.2839 16.30901 16.31181 16.31598 16.33695 16.33944
Levels: 0
```

### 2. Naïve Bayes:-

```
Naive Bayes Classifier for Discrete Predictors

Call:
naiveBayes.default(x = X, y = Y, laplace = laplace)

A-priori probabilities:
Y
        0         1
0.8463993 0.1536007

Conditional probabilities:
   Sex_Male
Y        [,1]      [,2]
  0 0.4221863 0.4940410
  1 0.5489614 0.4983369

   age
Y            33          34          35          36          37          38          39          40          41
  0 0.001077006 0.005385030 0.009693053 0.028540657 0.022078621 0.034464190 0.046849758 0.056542811 0.044157243
  1 0.000000000 0.000000000 0.005934718 0.002967359 0.008902077 0.017804154 0.008902077 0.023738872 0.020771513
   age
Y            42          43          44          45          46          47          48          49          50
  0 0.044157243 0.045772752 0.042541734 0.036618201 0.048465267 0.032310178 0.037156704 0.028540657 0.035541195
  1 0.029673591 0.014836795 0.020771513 0.023738872 0.023738872 0.029673591 0.023738872 0.044510386 0.041543027
   age
Y            51          52          53          54          55          56          57          58          59
  0 0.031771675 0.032848681 0.030694669 0.029079160 0.037695207 0.027463651 0.025309639 0.022617124 0.024232633
  1 0.035608309 0.026706231 0.029673591 0.035608309 0.035608309 0.047477745 0.047477745 0.050445104 0.050445104
   age
Y            60          61          62          63          64          65          66          67          68
  0 0.024232633 0.021001616 0.023694130 0.022078621 0.019924610 0.009154550 0.006462036 0.006462036 0.003231018
  1 0.044510386 0.044510386 0.041543027 0.056379822 0.032640950 0.014836795 0.017804154 0.035608309 0.011869436
   age
Y            69          70
  0 0.001615509 0.000538503
  1 0.000000000 0.000000000

   education
Y           1         2         3         4
  0 0.3974152 0.3214863 0.1696284 0.1114701
  1 0.5103858 0.2195846 0.1602374 0.1097923

   currentSmoker
Y        [,1]      [,2]
  0 0.4927302 0.5000818
  1 0.5074184 0.5006884

   cigsPerDay
Y             0           1           2           3           4           5           6           7           8
  0 0.507269790 0.021540118 0.004308024 0.026925148 0.003231018 0.029617663 0.003769521 0.002692515 0.002154012
  1 0.492581602 0.008902077 0.005934718 0.017804154 0.000000000 0.026706231 0.005934718 0.000000000 0.000000000
   cigsPerDay
Y             9          10          11          13          15          16          17          18          19
  0 0.030156166 0.027463651 0.000538503 0.000538503 0.047926764 0.000538503 0.001077006 0.000538503 0.001077006
  1 0.011869436 0.023738872 0.002967359 0.000000000 0.062314540 0.000000000 0.002967359 0.005934718 0.000000000
   cigsPerDay
Y            20          23          25          30          35          38          40          43          45
  0 0.186322025 0.001077006 0.012924071 0.046849758 0.004846527 0.000538503 0.018309101 0.012924071 0.001077006
  1 0.181008902 0.000000000 0.017804154 0.071216617 0.008902077 0.000000000 0.026706231 0.020771513 0.000000000
   cigsPerDay
Y            50          60          70
  0 0.001615509 0.001615509 0.000538503
  1 0.002967359 0.002967359 0.000000000

   BPMeds
Y         [,1]      [,2]
  0 0.02584814 0.1587249
  1 0.06231454 0.2420854
```


```
   BPMeds
Y         [,1]      [,2]
  0 0.02584814 0.1587249
  1 0.06231454 0.2420854
```


```
   prevalentStroke
Y          [,1]       [,2]
  0 0.004846527 0.06946680
  1 0.008902077 0.09406959

   prevalentHyp
Y        [,1]      [,2]
  0 0.2827141 0.4504399
  1 0.4925816 0.5006884

   diabetes
Y         [,1]      [,2]
  0 0.01884760 0.1360233
  1 0.04747774 0.2129747

   totChol
Y           113         119         135         137         140         143         144         145         148
  0 0.000538503 0.000538503 0.000538503 0.000538503 0.000538503 0.001077006 0.000000000 0.000538503 0.001077006
  1 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000 0.002967359 0.002967359 0.000000000 0.000000000
   totChol
Y           149         150         152         153         154         155         156         157         158
  0 0.000000000 0.002154012 0.002154012 0.001077006 0.002692515 0.004308024 0.001077006 0.001077006 0.001615509
  1 0.002967359 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000 0.005934718
   totChol
Y           159         160         161         162         163         164         165         166         167
  0 0.002154012 0.004308024 0.002154012 0.002692515 0.003231018 0.002154012 0.008077544 0.001077006 0.004308024
  1 0.000000000 0.000000000 0.002967359 0.000000000 0.000000000 0.000000000 0.002967359 0.000000000 0.002967359
   totChol
Y           168         169         170         171         172         173         174         175         176
  0 0.001615509 0.001077006 0.004846527 0.001615509 0.002692515 0.003769521 0.003231018 0.006462036 0.004846527
  1 0.005934718 0.000000000 0.011869436 0.000000000 0.002967359 0.002967359 0.005934718 0.005934718 0.000000000
   totChol
Y           177         178         179         180         181         182         183         184         185
  0 0.002692515 0.003769521 0.003769521 0.005923533 0.003231018 0.002692515 0.003231018 0.002692515 0.010231556
  1 0.002967359 0.002967359 0.005934718 0.008902077 0.000000000 0.000000000 0.002967359 0.005934718 0.000000000
   totChol
Y           186         187         188         189         190         191         192         193         194
  0 0.007539041 0.003769521 0.005385030 0.004308024 0.009154550 0.000538503 0.003769521 0.009154550 0.005385030
  1 0.000000000 0.005934718 0.000000000 0.000000000 0.008902077 0.002967359 0.000000000 0.011869436 0.008902077
   totChol
Y           195         196         197         198         199         200         201         202         203
  0 0.011847065 0.007000539 0.011847065 0.006462036 0.005385030 0.015078083 0.005385030 0.008616047 0.007539041
  1 0.002967359 0.000000000 0.002967359 0.005934718 0.008902077 0.008902077 0.005934718 0.002967359 0.005934718
   totChol
Y           204         205         206         207         208         209         210         211         212
  0 0.004846527 0.011308562 0.008616047 0.008077544 0.006462036 0.006462036 0.012924071 0.004846527 0.008616047
  1 0.002967359 0.005934718 0.014836795 0.000000000 0.017804154 0.002967359 0.014836795 0.017804154 0.011869436
   totChol
Y           213         214         215         216         217         218         219         220         221
  0 0.009693053 0.010231556 0.010770059 0.009154550 0.005923533 0.004846527 0.008077544 0.014539580 0.007000539
  1 0.002967359 0.008902077 0.014836795 0.000000000 0.008902077 0.005934718 0.008902077 0.011869436 0.008902077
   totChol
Y           222         223         224         225         226         227         228         229         230
  0 0.008616047 0.006462036 0.007000539 0.009154550 0.010231556 0.007000539 0.005385030 0.011847065 0.015078083
  1 0.005934718 0.008902077 0.002967359 0.023738872 0.000000000 0.008902077 0.005934718 0.008902077 0.020771513
   totChol
Y           231         232         233         234         235         236         237         238         239
  0 0.004308024 0.014001077 0.010231556 0.010231556 0.011308562 0.003231018 0.007000539 0.011847065 0.009154550
  1 0.000000000 0.017804154 0.005934718 0.011869436 0.014836795 0.005934718 0.005934718 0.008902077 0.008902077
   totChol
Y           240         241         242         243         244         245         246         247         248
  0 0.019924610 0.009693053 0.009154550 0.007000539 0.006462036 0.014001077 0.010770059 0.004308024 0.008616047
  1 0.011869436 0.008902077 0.002967359 0.008902077 0.005934718 0.011869436 0.008902077 0.008902077 0.005934718
```

```
Y          249         250         251         252         253         254         255         256         257
  0 0.005385030 0.010770059 0.002692515 0.008616047 0.008077544 0.012924071 0.005385030 0.006462036 0.003769521
  1 0.008902077 0.008902077 0.000000000 0.011869436 0.008902077 0.000000000 0.002967359 0.005934718 0.008902077
   totChol
Y          258         259         260         261         262         263         264         265         266
  0 0.010770059 0.005385030 0.014539580 0.005923533 0.008616047 0.003769521 0.004846527 0.007000539 0.005923533
  1 0.005934718 0.011869436 0.020771513 0.002967359 0.002967359 0.008902077 0.008902077 0.008902077 0.017804154
   totChol
Y          267         268         269         270         271         272         273         274         275
  0 0.003231018 0.005923533 0.002154012 0.011308562 0.003769521 0.004846527 0.009154550 0.007539041 0.008616047
  1 0.008902077 0.000000000 0.002967359 0.017804154 0.005934718 0.014836795 0.014836795 0.002967359 0.011869436
   totChol
Y          276         277         278         279         280         281         282         283         284
  0 0.002154012 0.001615509 0.003231018 0.005923533 0.004846527 0.003769521 0.003231018 0.002154012 0.001615509
  1 0.005934718 0.000000000 0.005934718 0.000000000 0.008902077 0.000000000 0.002967359 0.005934718 0.000000000
   totChol
Y          285         286         287         288         289         290         291         292         293
  0 0.005923533 0.003769521 0.004308024 0.003231018 0.002692515 0.003769521 0.004846527 0.003769521 0.003231018
  1 0.011869436 0.005934718 0.005934718 0.000000000 0.002967359 0.002967359 0.002967359 0.002967359 0.005934718
   totChol
Y          294         295         296         297         298         299         300         301         302
  0 0.003231018 0.003769521 0.002692515 0.003231018 0.001615509 0.000538503 0.004308024 0.001077006 0.001077006
  1 0.002967359 0.000000000 0.005934718 0.000000000 0.002967359 0.005934718 0.011869436 0.000000000 0.002967359
   totChol
Y          303         304         305         306         307         308         309         310         311
  0 0.002692515 0.002692515 0.004308024 0.001615509 0.001077006 0.002154012 0.002692515 0.005385030 0.001615509
  1 0.005934718 0.002967359 0.005934718 0.002967359 0.000000000 0.000000000 0.002967359 0.002967359 0.000000000
   totChol
Y          312         313         314         315         316         317         318         319         320
  0 0.001615509 0.003231018 0.002154012 0.002154012 0.000538503 0.001077006 0.001615509 0.000538503 0.002692515
  1 0.014836795 0.002967359 0.000000000 0.000000000 0.002967359 0.000000000 0.000000000 0.000000000 0.005934718
   totChol
Y          321         322         323         325         326         327         328         329         330
  0 0.000538503 0.001077006 0.001077006 0.002154012 0.002692515 0.000000000 0.001077006 0.001077006 0.001077006
  1 0.000000000 0.000000000 0.000000000 0.002967359 0.000000000 0.005934718 0.002967359 0.000000000 0.000000000
   totChol
Y          331         332         333         334         335         336         337         338         339
  0 0.000538503 0.001077006 0.001077006 0.001077006 0.000538503 0.000538503 0.000538503 0.000538503 0.000538503
  1 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000 0.002967359
   totChol
Y          340         342         344         345         346         347         350         351         352
  0 0.001615509 0.001615509 0.001615509 0.001077006 0.001615509 0.000538503 0.000538503 0.000538503 0.001077006
  1 0.002967359 0.002967359 0.000000000 0.002967359 0.002967359 0.000000000 0.002967359 0.000000000 0.002967359
   totChol
Y          353         354         355         358         360         363         367         370         371
  0 0.000538503 0.000538503 0.000538503 0.000000000 0.000538503 0.000538503 0.000538503 0.000000000 0.000538503
  1 0.000000000 0.000000000 0.000000000 0.002967359 0.000000000 0.000000000 0.000000000 0.002967359 0.000000000
   totChol
Y          372         373         382         385         398         410         432         439         453
  0 0.000000000 0.000538503 0.000538503 0.000538503 0.000538503 0.000538503 0.000000000 0.000000000 0.000538503
  1 0.002967359 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000 0.002967359 0.002967359 0.000000000

   sysBP
Y         83.5          85        85.5          90          92        92.5          93        93.5          94
  0 0.000000000 0.000538503 0.000000000 0.000538503 0.000538503 0.000000000 0.001077006 0.000538503 0.001077006
  1 0.002967359 0.000000000 0.002967359 0.000000000 0.000000000 0.002967359 0.000000000 0.000000000 0.000000000
   sysBP
Y           95        95.5          96        96.5          97        97.5          98        98.5          99
  0 0.001615509 0.000538503 0.003769521 0.001615509 0.002154012 0.001615509 0.002692515 0.000538503 0.002154012
  1 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000 0.005934718 0.000000000 0.000000000
   sysBP
Y         99.5         100       100.5         101       101.5         102       102.5         103       103.5
  0 0.000538503 0.008616047 0.002154012 0.006462036 0.000538503 0.008077544 0.002692515 0.005923533 0.000000000
  1 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000 0.002967359 0.002967359 0.008902077 0.002967359
   sysBP
Y          104         105       105.5         106       106.5         107       107.5         108       108.5
  0 0.003231018 0.011847065 0.001077006 0.010231556 0.000538503 0.007539041 0.008616047 0.012385568 0.002154012
  1 0.005934718 0.000000000 0.000000000 0.002967359 0.000000000 0.000000000 0.005934718 0.000000000 0.000000000
```

```
   sysBP
Y        113.5         114       114.5         115       115.5         116       116.5         117       117.5
  0 0.002692515 0.014539580 0.001077006 0.023155627 0.002692515 0.017232095 0.001615509 0.010231556 0.010231556
  1 0.002967359 0.017804154 0.002967359 0.011869436 0.000000000 0.008902077 0.002967359 0.008902077 0.002967359
   sysBP
Y          118       118.5         119       119.5         120       120.5         121       121.5         122
  0 0.016155089 0.003769521 0.015078083 0.001615509 0.023155627 0.002692515 0.011847065 0.003769521 0.018309101
  1 0.002967359 0.011869436 0.005934718 0.000000000 0.014836795 0.008902077 0.000000000 0.002967359 0.011869436
   sysBP
Y        122.5         123       123.5         124       124.5         125       125.5         126       126.5
  0 0.005923533 0.019386107 0.002692515 0.021001616 0.002692515 0.020463113 0.003231018 0.018309101 0.004308024
  1 0.005934718 0.023738872 0.000000000 0.005934718 0.002967359 0.014836795 0.002967359 0.020771513 0.002967359
   sysBP
Y          127       127.5         128       128.5         129       129.5         130       130.5         131
  0 0.016155089 0.010231556 0.019386107 0.004308024 0.019386107 0.002154012 0.026925148 0.002692515 0.012385568
  1 0.014836795 0.014836795 0.023738872 0.000000000 0.002967359 0.005934718 0.008902077 0.000000000 0.008902077
   sysBP
Y        131.5         132       132.5         133       133.5         134       134.5         135       135.5
  0 0.005923533 0.016155089 0.005923533 0.012924071 0.002154012 0.013462574 0.004308024 0.016155089 0.001077006
  1 0.000000000 0.008902077 0.005934718 0.011869436 0.002967359 0.014836795 0.000000000 0.011869436 0.000000000
   sysBP
Y          136       136.5         137       137.5         138       138.5         139       139.5         140
  0 0.009154550 0.004846527 0.009693053 0.005923533 0.010770059 0.000538503 0.008616047 0.000538503 0.015616586
  1 0.008902077 0.005934718 0.011869436 0.005934718 0.011869436 0.002967359 0.023738872 0.000000000 0.008902077
   sysBP
Y        140.5         141       141.5         142       142.5         143       143.5         144       144.5
  0 0.001077006 0.011308562 0.001077006 0.008077544 0.004846527 0.006462036 0.003231018 0.007539041 0.001615509
  1 0.000000000 0.017804154 0.000000000 0.008902077 0.002967359 0.005934718 0.000000000 0.008902077 0.002967359
   sysBP
Y          145       145.5         146       146.5         147       147.5         148       148.5         149
  0 0.012924071 0.001077006 0.008077544 0.003231018 0.004846527 0.003231018 0.008077544 0.001077006 0.006462036
  1 0.020771513 0.002967359 0.020771513 0.002967359 0.005934718 0.002967359 0.011869436 0.005934718 0.008902077
   sysBP
Y        149.5         150       150.5         151       151.5         152       152.5         153       153.5
  0 0.002154012 0.010770059 0.001615509 0.004308024 0.000538503 0.005923533 0.001615509 0.004308024 0.000538503
  1 0.000000000 0.020771513 0.002967359 0.002967359 0.002967359 0.000000000 0.005934718 0.011869436 0.000000000
   sysBP
Y          154       154.5         155       155.5         156       156.5         157       157.5         158
  0 0.007000539 0.001077006 0.008077544 0.000000000 0.005385030 0.000000000 0.003769521 0.001077006 0.007539041
  1 0.011869436 0.000000000 0.008902077 0.005934718 0.014836795 0.002967359 0.014836795 0.000000000 0.017804154
   sysBP
Y        158.5         159       159.5         160       160.5         161       161.5         162       162.5
  0 0.001615509 0.004846527 0.000538503 0.007000539 0.000538503 0.001615509 0.001077006 0.002154012 0.001615509
  1 0.000000000 0.005934718 0.000000000 0.000000000 0.005934718 0.005934718 0.000000000 0.000000000 0.005934718
   sysBP
Y          163       163.5         164       164.5         165         166       166.5         167       167.5
  0 0.004308024 0.002154012 0.003769521 0.000000000 0.005385030 0.003231018 0.000000000 0.001077006 0.002154012
  1 0.005934718 0.000000000 0.005934718 0.002967359 0.008902077 0.005934718 0.002967359 0.008902077 0.000000000
   sysBP
Y          168       168.5         169         170         171         172       172.5         173         174
  0 0.001615509 0.001615509 0.001077006 0.004308024 0.001077006 0.001077006 0.000538503 0.001615509 0.002154012
  1 0.000000000 0.000000000 0.002967359 0.002967359 0.005934718 0.002967359 0.002967359 0.008902077 0.000000000
   sysBP
Y          175       175.5         176       176.5         177       177.5         178         179       179.5
  0 0.002692515 0.000538503 0.002692515 0.000538503 0.000538503 0.001077006 0.001077006 0.000538503 0.000538503
  1 0.005934718 0.000000000 0.000000000 0.000000000 0.000000000 0.008902077 0.002967359 0.002967359 0.000000000
   sysBP
Y          180       180.5         181         182       182.5         183         184         185       185.5
  0 0.001615509 0.000538503 0.001615509 0.001077006 0.000538503 0.000538503 0.000000000 0.000000000 0.000538503
  1 0.002967359 0.000000000 0.002967359 0.008902077 0.002967359 0.002967359 0.002967359 0.005934718 0.000000000
   sysBP
Y          186       186.5         187       187.5         188       188.5         189         190         191
  0 0.000538503 0.001077006 0.000538503 0.000538503 0.000538503 0.000538503 0.001077006 0.002154012 0.001077006
  1 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000 0.002967359 0.005934718 0.002967359
   sysBP
Y          192       192.5         193         194         195         196         197       197.5         198
  0 0.001077006 0.000000000 0.000538503 0.000538503 0.001077006 0.001077006 0.001077006 0.000000000 0.001077006
  1 0.000000000 0.005934718 0.000000000 0.000000000 0.005934718 0.008902077 0.002967359 0.002967359 0.002967359
```

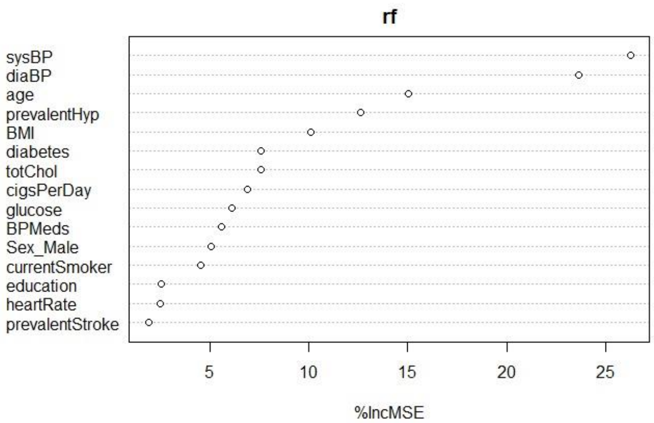
```
   sysBP
Y          199       199.5         200         202       202.5         204         205       205.5         206
  0 0.001077006 0.000000000 0.000538503 0.000538503 0.000538503 0.000000000 0.000538503 0.000538503 0.000538503
  1 0.000000000 0.002967359 0.005934718 0.000000000 0.000000000 0.005934718 0.000000000 0.000000000 0.000000000
   sysBP
Y          207       207.5         209         210         213         214         215         217         220
  0 0.000538503 0.000000000 0.000538503 0.000538503 0.000538503 0.000000000 0.000000000 0.000000000 0.000538503
  1 0.000000000 0.002967359 0.000000000 0.005934718 0.000000000 0.002967359 0.002967359 0.002967359 0.000000000
   sysBP
Y          230         232         248         295
  0 0.000538503 0.000538503 0.000000000 0.000000000
  1 0.000000000 0.000000000 0.002967359 0.002967359

   diaBP
Y           51          52          54          55          56          57        57.5          58          59
  0 0.000000000 0.001077006 0.000000000 0.000000000 0.001077006 0.001077006 0.000538503 0.001077006 0.001615509
  1 0.002967359 0.000000000 0.002967359 0.005934718 0.000000000 0.002967359 0.000000000 0.000000000 0.002967359
   diaBP
Y         59.5          60        60.5          61        61.5          62        62.5          63        63.5
  0 0.000538503 0.007539041 0.000538503 0.005923533 0.001077006 0.005385030 0.002154012 0.003769521 0.001615509
  1 0.000000000 0.002967359 0.000000000 0.002967359 0.000000000 0.002967359 0.005934718 0.000000000 0.002967359
   diaBP
Y           64        64.5          65        65.5          66        66.5          67        67.5          68
  0 0.005923533 0.002692515 0.010770059 0.001615509 0.010231556 0.003769521 0.010231556 0.007539041 0.012385568
  1 0.005934718 0.000000000 0.005934718 0.000000000 0.002967359 0.011869436 0.008902077 0.000000000 0.014836795
   diaBP
Y         68.5          69        69.5          70        70.5          71        71.5          72        72.5
  0 0.001615509 0.016155089 0.000538503 0.033925687 0.001077006 0.016693592 0.001077006 0.019924610 0.015078083
  1 0.000000000 0.005934718 0.000000000 0.029673591 0.000000000 0.011869436 0.000000000 0.008902077 0.008902077
   diaBP
Y           73        73.5          74        74.5          75        75.5          76        76.5          77
  0 0.025848142 0.003769521 0.024771136 0.004846527 0.028540657 0.002692515 0.019924610 0.004846527 0.018309101
  1 0.008902077 0.005934718 0.020771513 0.000000000 0.017804154 0.005934718 0.020771513 0.000000000 0.002967359
   diaBP
Y         77.5          78        78.5          79        79.5          80        80.5          81        81.5
  0 0.012385568 0.028002154 0.004846527 0.026386645 0.003231018 0.057081314 0.002154012 0.032310178 0.001615509
  1 0.005934718 0.038575668 0.002967359 0.023738872 0.002967359 0.050445104 0.000000000 0.035608309 0.002967359
   diaBP
Y           82        82.5          83        83.5          84        84.5          85        85.5          86
  0 0.039849219 0.014001077 0.021001616 0.005385030 0.027463651 0.002692515 0.036618201 0.007539041 0.027463651
  1 0.038575668 0.008902077 0.023738872 0.002967359 0.023738872 0.005934718 0.023738872 0.002967359 0.017804154
   diaBP
Y         86.5          87        87.5          88        88.5          89        89.5          90        90.5
  0 0.006462036 0.026925148 0.004846527 0.024232633 0.002154012 0.019386107 0.002154012 0.027463651 0.000538503
  1 0.002967359 0.020771513 0.002967359 0.032640950 0.002967359 0.014836795 0.005934718 0.032640950 0.000000000
   diaBP
Y           91        91.5          92        92.5          93        93.5          94        94.5          95
  0 0.010231556 0.001615509 0.016693592 0.006462036 0.013462574 0.001615509 0.012924071 0.002692515 0.011847065
  1 0.020771513 0.000000000 0.026706231 0.005934718 0.005934718 0.002967359 0.029673591 0.000000000 0.023738872
   diaBP
Y         95.5          96        96.5          97        97.5          98        98.5          99        99.5
  0 0.001615509 0.009154550 0.001077006 0.010770059 0.003231018 0.012385568 0.001077006 0.007539041 0.000000000
  1 0.000000000 0.011869436 0.002967359 0.005934718 0.000000000 0.020771513 0.000000000 0.002967359 0.002967359
   diaBP
Y          100       100.5         101       101.5         102       102.5         103       103.5         104
  0 0.008077544 0.001077006 0.004846527 0.000538503 0.005923533 0.001077006 0.002692515 0.001077006 0.001615509
  1 0.023738872 0.008902077 0.011869436 0.000000000 0.005934718 0.000000000 0.002967359 0.000000000 0.005934718
   diaBP
Y        104.5         105         106       106.5         107         108         109         110         111
  0 0.001077006 0.007000539 0.004846527 0.001615509 0.002154012 0.004846527 0.003769521 0.004308024 0.000538503
  1 0.000000000 0.014836795 0.005934718 0.002967359 0.011869436 0.002967359 0.002967359 0.002967359 0.002967359
   diaBP
Y          112       112.5         113         114       114.5         115       115.5         116         117
  0 0.000538503 0.000538503 0.001077006 0.002154012 0.000000000 0.001615509 0.000538503 0.000000000 0.000000000
  1 0.008902077 0.000000000 0.002967359 0.000000000 0.002967359 0.005934718 0.000000000 0.002967359 0.002967359
   diaBP
Y          118         119         120         121       122.5       124.5         125       127.5         130
  0 0.001615509 0.000538503 0.001077006 0.000538503 0.001077006 0.000538503 0.000000000 0.000538503 0.000538503
  1 0.005934718 0.000000000 0.014836795 0.002967359 0.000000000 0.000000000 0.005934718 0.000000000 0.008902077
```

```
   diaBP
Y          132         133         135         136
  0 0.000538503 0.000538503 0.000000000 0.000538503
  1 0.000000000 0.000000000 0.002967359 0.000000000

   BMI
Y      [,1]     [,2]
  0 25.66423 3.910715
  1 26.33519 4.527399

   heartRate
Y           44          45          47          48          50          52          53          54          55
  0 0.000538503 0.001077006 0.000538503 0.001077006 0.007539041 0.004846527 0.000538503 0.002692515 0.006462036
  1 0.000000000 0.000000000 0.000000000 0.000000000 0.002967359 0.005934718 0.002967359 0.000000000 0.008902077
   heartRate
Y           56          57          58          59          60          61          62          63          64
  0 0.005385030 0.003231018 0.009693053 0.001615509 0.060850835 0.000538503 0.011847065 0.022078621 0.015616586
  1 0.005934718 0.002967359 0.017804154 0.000000000 0.050445104 0.000000000 0.023738872 0.014836795 0.011869436
   heartRate
Y           65          66          67          68          69          70          71          72          73
  0 0.043080237 0.015078083 0.023694130 0.029617663 0.014001077 0.076467421 0.008616047 0.058158320 0.014001077
  1 0.059347181 0.017804154 0.020771513 0.041543027 0.008902077 0.091988131 0.005934718 0.047477745 0.020771513
   heartRate
Y           74          75          76          77          78          79          80          81          82
  0 0.008616047 0.137856758 0.010231556 0.011308562 0.023155627 0.009154550 0.090468498 0.002692515 0.015616586
  1 0.008902077 0.136498516 0.011869436 0.011869436 0.017804154 0.011869436 0.091988131 0.002967359 0.011869436
   heartRate
Y           83          84          85          86          87          88          89          90          91
  0 0.011308562 0.005385030 0.058696823 0.009154550 0.005923533 0.022078621 0.001615509 0.037695207 0.003231018
  1 0.005934718 0.008902077 0.035608309 0.005934718 0.005934718 0.005934718 0.000000000 0.053412463 0.000000000
   heartRate
Y           92          93          94          95          96          97          98         100         101
  0 0.009693053 0.002692515 0.006462036 0.019386107 0.009154550 0.000538503 0.005385030 0.022078621 0.000538503
  1 0.014836795 0.000000000 0.005934718 0.029673591 0.011869436 0.000000000 0.002967359 0.017804154 0.000000000
   heartRate
Y          102         103         104         105         106         107         108         110         115
  0 0.002154012 0.000538503 0.001077006 0.003231018 0.000538503 0.001077006 0.002154012 0.006462036 0.000538503
  1 0.000000000 0.000000000 0.000000000 0.002967359 0.000000000 0.000000000 0.000000000 0.017804154 0.002967359
   heartRate
Y          120         122         125         130         140
  0 0.001077006 0.000538503 0.000538503 0.000538503 0.000538503
  1 0.002967359 0.000000000 0.000000000 0.000000000 0.000000000

   glucose
Y           40          44          45          47          50          52          53          54          55
  0 0.000538503 0.000538503 0.001615509 0.001077006 0.001077006 0.000538503 0.001615509 0.002154012 0.004308024
  1 0.000000000 0.000000000 0.002967359 0.000000000 0.002967359 0.000000000 0.002967359 0.000000000 0.002967359
   glucose
Y           57          58          59          60          61          62          63          64          65
  0 0.005385030 0.005923533 0.003769521 0.018847604 0.005923533 0.014539580 0.014001077 0.010770059 0.023155627
  1 0.008902077 0.000000000 0.005934718 0.011869436 0.002967359 0.008902077 0.020771513 0.008902077 0.035608309
   glucose
Y           66          67          68          69          70          71          72          73          74
  0 0.019386107 0.028540657 0.026386645 0.019924610 0.043080237 0.021540118 0.025309639 0.040387722 0.036618201
  1 0.014836795 0.014836795 0.014836795 0.011869436 0.035608309 0.011869436 0.038575668 0.017804154 0.029673591
   glucose
Y           75          76          77          78          79          80          81          82          83
  0 0.045772752 0.032848681 0.043618740 0.038233710 0.027463651 0.038772213 0.016155089 0.025309639 0.037156704
  1 0.056379822 0.041543027 0.044510386 0.038575668 0.023738872 0.020771513 0.011869436 0.017804154 0.050445104
   glucose
Y           84          85          86          87          88          89          90          91          92
  0 0.031771675 0.030156166 0.021001616 0.029617663 0.016155089 0.005923533 0.020463113 0.008077544 0.007000539
  1 0.026706231 0.029673591 0.014836795 0.038575668 0.020771513 0.005934718 0.023738872 0.002967359 0.008902077
   glucose
Y           93          94          95          96          97          98          99         100         101
  0 0.013462574 0.011308562 0.010770059 0.008616047 0.008077544 0.004308024 0.007000539 0.016155089 0.001077006
  1 0.014836795 0.014836795 0.020771513 0.005934718 0.008902077 0.011869436 0.000000000 0.014836795 0.000000000
```

```
Y           102         103         104         105         106         107         108         109         110
  0 0.006462036 0.007000539 0.003769521 0.001615509 0.002154012 0.003769521 0.003769521 0.000538503 0.002154012
  1 0.000000000 0.014836795 0.005934718 0.005934718 0.008902077 0.000000000 0.000000000 0.005934718 0.002967359
   glucose
Y           112         113         114         115         116         117         118         120         121
  0 0.003231018 0.001615509 0.000538503 0.003769521 0.002154012 0.003231018 0.002692515 0.002154012 0.000538503
  1 0.000000000 0.002967359 0.005934718 0.002967359 0.002967359 0.002967359 0.005934718 0.000000000 0.000000000
   glucose
Y           122         123         124         125         127         129         131         132         136
  0 0.000538503 0.000000000 0.000538503 0.000538503 0.001077006 0.000538503 0.000538503 0.000538503 0.001077006
  1 0.000000000 0.008902077 0.002967359 0.000000000 0.002967359 0.000000000 0.000000000 0.002967359 0.000000000
   glucose
Y           137         140         144         147         148         150         156         163         167
  0 0.001615509 0.001077006 0.000538503 0.000000000 0.000000000 0.000538503 0.000538503 0.000000000 0.000000000
  1 0.000000000 0.000000000 0.000000000 0.002967359 0.002967359 0.000000000 0.000000000 0.002967359 0.002967359
   glucose
Y           170         172         173         177         183         193         202         205         206
  0 0.000538503 0.000538503 0.000000000 0.000538503 0.000538503 0.000538503 0.000538503 0.000000000 0.000000000
  1 0.000000000 0.000000000 0.002967359 0.000000000 0.000000000 0.000000000 0.000000000 0.002967359 0.005934718
   glucose
Y           207         216         225         254         256         292         348         386         394
  0 0.000538503 0.000000000 0.000538503 0.000538503 0.000000000 0.000000000 0.000538503 0.000538503 0.000000000
  1 0.000000000 0.002967359 0.000000000 0.000000000 0.002967359 0.002967359 0.000000000 0.000000000 0.005934718
```

## 3. Random Forest:-

Percentage of predicted classifications correct 84.6994%
The model is classified as level 0 thus the patient will not have coronary heart disease in 10 years.




```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 1217  212
         1   18   17

               Accuracy : 0.8429
                 95% CI : (0.8232, 0.8612)
    No Information Rate : 0.8436
    P-Value [Acc > NIR] : 0.5462

                  Kappa : 0.0911

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.98543
            Specificity : 0.07424
         Pos Pred Value : 0.85164
         Neg Pred Value : 0.48571
             Prevalence : 0.84358
         Detection Rate : 0.83128
   Detection Prevalence : 0.97609
      Balanced Accuracy : 0.52983

       'Positive' Class : 0
```

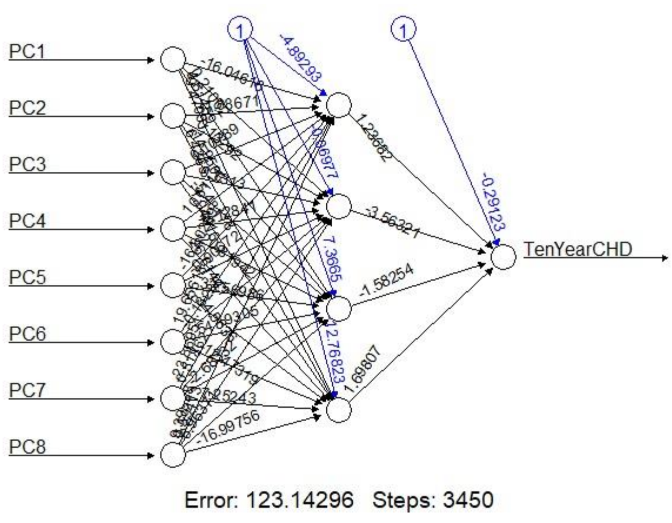
## 4. Logistic Regression:-

For logistic regression algorithm, we found the accuracy to be 83.87%.

```
Call:
glm(formula = as.factor(TenYearCHD) ~ ., family = "binomial",
    data = train.df)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.6873  -0.5937  -0.4265  -0.2882   2.7505

Coefficients:
                 Estimate Std. Error z value             Pr(>|z|)
(Intercept)     -8.401462   0.930109  -9.033 < 0.0000000000000002 ***
Sex_Male         0.436716   0.139550   3.129             0.001751 **
age              0.057447   0.008656   6.636      0.0000000000321 ***
education       -0.141695   0.065933  -2.149             0.031629 *
currentSmoker    0.168800   0.198921   0.849             0.396117
cigsPerDay       0.015696   0.007818   2.008             0.044664 *
BPMeds          -0.017661   0.301117  -0.059             0.953231
prevalentStroke  1.675153   0.815762   2.053             0.040026 *
prevalentHyp     0.016324   0.174708   0.093             0.925557
diabetes         0.230871   0.377203   0.612             0.540498
totChol          0.003407   0.001408   2.420             0.015535 *
sysBP            0.018204   0.004893   3.720             0.000199 ***
diaBP            0.002043   0.008306   0.246             0.805748
BMI              0.001437   0.016413   0.088             0.930215
heartRate       -0.007146   0.005553  -1.287             0.198117
glucose          0.006026   0.002818   2.139             0.032472 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1875.2  on 2193  degrees of freedom
Residual deviance: 1647.4  on 2178  degrees of freedom
AIC: 1679.4

Number of Fisher Scoring iterations: 5

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
0.01372 0.06422 0.11638 0.15269 0.19968 0.90813
        0         1
0.1348806 0.2515136

    FALSE TRUE
  0  1840   19
  1   301   34
```

## 5. Neural Net:-

The model is classified as level 0 thus the patient will not have coronary heart disease in 10 years.




```
Confusion Matrix and Statistics

        prediction
actual     0    1
     0  1211   33
     1   200   20

               Accuracy : 0.8408
                 95% CI : (0.8211, 0.8592)
    No Information Rate : 0.9638
    P-Value [Acc > NIR] : 1

                  Kappa : 0.0936

 Mcnemar's Test P-Value : <0.0000000000000002

            Sensitivity : 0.85826
            Specificity : 0.37736
         Pos Pred Value : 0.97347
         Neg Pred Value : 0.09091
             Prevalence : 0.96380
         Detection Rate : 0.82719
   Detection Prevalence : 0.84973
      Balanced Accuracy : 0.61781

       'Positive' Class : 0
```

## 6. Linear Discriminant Analysis:-

```
Call:
lda(TenYearCHD ~ ., data = train.df, na.action = "na.omit")

Prior probabilities of groups:
        0         1 
0.8473108 0.1526892 

Group means:
   Sex_Male      age education currentSmoker cigsPerDay     BPMeds prevalentStroke prevalentHyp   diabetes
0 0.4410974 48.79828  1.997848     0.4862829   8.844002 0.02689618     0.001613771    0.2974718 0.02313072
1 0.5313433 54.32537  1.740299     0.5164179  10.447761 0.06865672     0.011940299    0.5223881 0.07761194
   totChol   sysBP    diaBP      BMI heartRate  glucose
0 235.9252 130.752 82.38811 25.78779  75.41151 80.64497
1 249.3642 145.806 88.06119 26.65618  76.03582 89.57313

Coefficients of linear discriminants:
                         LD1
Sex_Male         0.385006951
age              0.056147895
education       -0.141663379
currentSmoker    0.173434582
cigsPerDay       0.017049104
BPMeds           0.162248582
prevalentStroke  3.024922180
prevalentHyp    -0.027685808
diabetes         0.482156884
totChol          0.002575647
sysBP            0.026878585
diaBP           -0.003697390
BMI             -0.007378632
heartRate       -0.007452305
glucose          0.009190746
   0    1 
1420   44 
```

## 7. Support Vector Machine:-

The model is classified as level 0 thus the patient will not have coronary heart disease in 10 years.

```
Support Vector Machines with Linear Kernel

2561 samples
  15 predictor
   2 classes: '0', '1'

Pre-processing: centered (15), scaled (15)
Resampling: Cross-Validated (10 fold, repeated 3 times)
Summary of sample sizes: 2305, 2304, 2306, 2305, 2305, 2305, ...
Resampling results:

  Accuracy   Kappa
  0.8508412  0

Tuning parameter 'C' was held constant at a value of 1
```

## V. Performance Evaluation

Evaluating our machine learning algorithm is an essential part of any project. Most of the times we use classification accuracy to measure the performance of our model, however it is not enough to truly judge our model. In this project, we have made use of confusion matrices, ROC curves & Lift charts to evaluate different models.

Classification Accuracy is what we usually mean, when we use the term accuracy. It is the ratio of number of correct predictions to the total number of input samples.

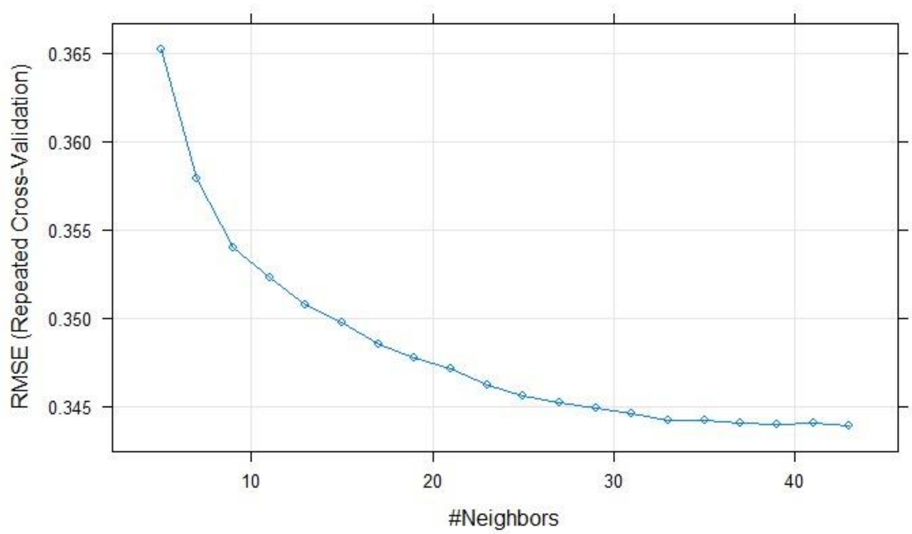
Confusion Matrix as the name suggests gives us a matrix as output and describes the complete performance of the model.

### 1. K-Nearest Neighbors: -

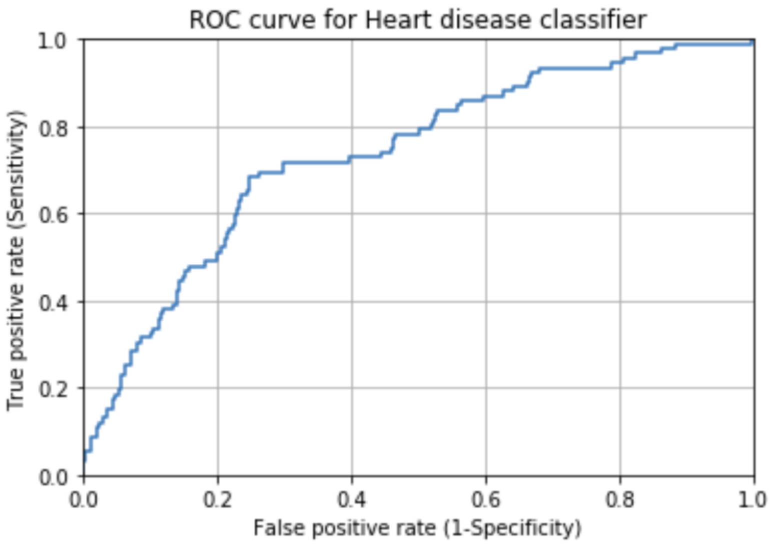
The confusion matrix is created: -

```
knn.pred    0    1
       0 1257  204
       1    1    2
Accuracy:        0.86
Precision:       0.999
Recall:          0.86
F-measure:       0.925
```

```
 k  RMSE       Rsquared    MAE
 5  0.3651653  0.03613560  0.2229062
 7  0.3579119  0.04043436  0.2237255
 9  0.3539518  0.04243793  0.2242940
11  0.3523180  0.04144572  0.2247124
13  0.3507094  0.04266692  0.2252169
15  0.3497306  0.04248450  0.2257110
17  0.3485067  0.04454184  0.2258991
19  0.3477617  0.04582425  0.2260211
21  0.3471409  0.04663127  0.2261944
23  0.3462341  0.04932322  0.2263275
25  0.3456145  0.05075775  0.2261673
27  0.3452036  0.05170426  0.2262544
29  0.3448638  0.05272717  0.2262164
31  0.3445690  0.05398300  0.2262074
33  0.3441698  0.05567272  0.2262103
35  0.3441968  0.05547592  0.2262222
37  0.3440556  0.05621621  0.2262110
39  0.3439878  0.05617782  0.2261798
41  0.3440141  0.05575533  0.2263301
43  0.3439350  0.05621896  0.2263700

RMSE was used to select the optimal model using the smallest value.
The final value used for the model was k = 43.
```



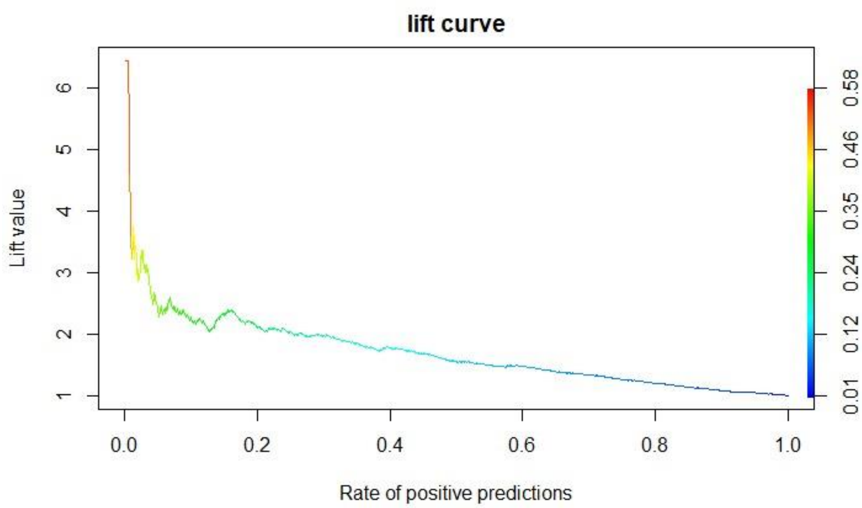
## 2. Naïve Bayes :-

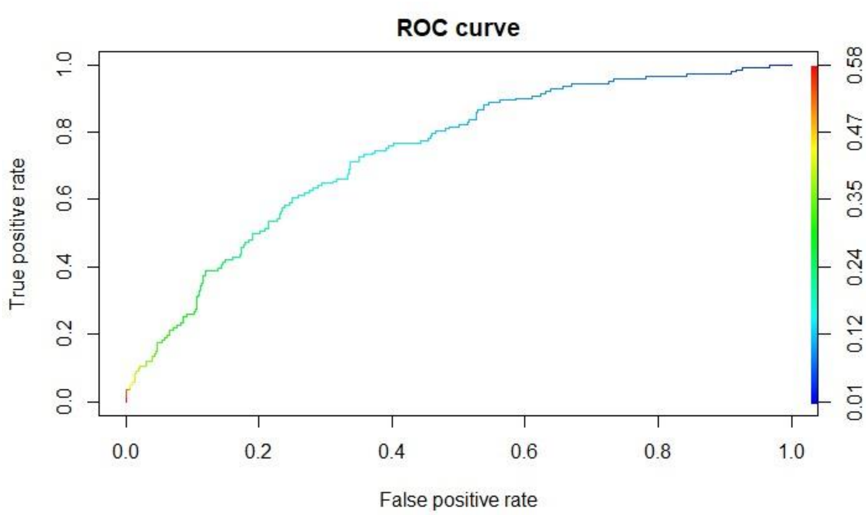



Confusion Matrix & Statistics

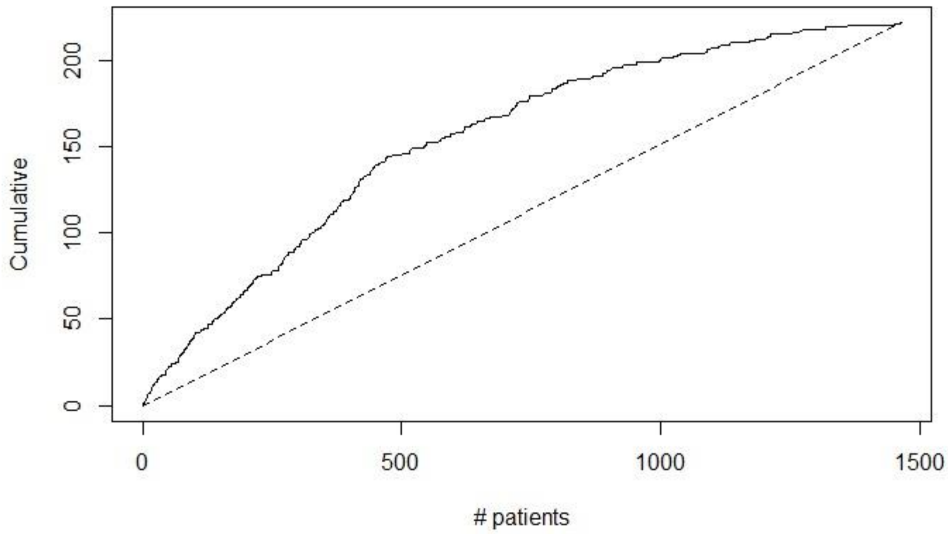
| actual | prediction 0 | prediction 1 |
|---|---|---|
| 0 | 658 | 1 |
| 1 | 88 | 4 |

## 3. Random Forest:-



## 4. Logistic Regression :-




Confusion Matrix

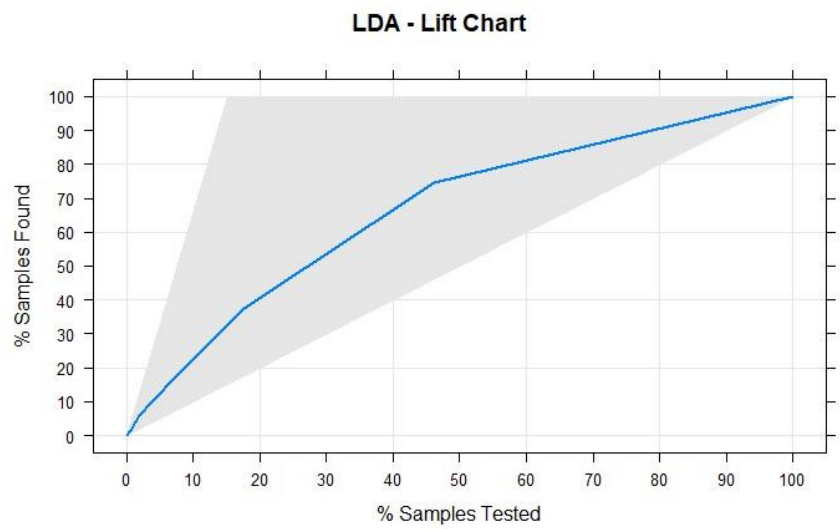
```
   FALSE TRUE
0   1840   19
1    301   34
```

## 5. Neural Net :-

```
Confusion Matrix and Statistics

      prediction
actual    0    1
     0 1211   33
     1  200   20

               Accuracy : 0.8408
                 95% CI : (0.8211, 0.8592)
    No Information Rate : 0.9638
    P-Value [Acc > NIR] : 1

                  Kappa : 0.0936

 Mcnemar's Test P-Value : <0.0000000000000002

            Sensitivity : 0.85826
            Specificity : 0.37736
         Pos Pred Value : 0.97347
         Neg Pred Value : 0.09091
             Prevalence : 0.96380
         Detection Rate : 0.82719
   Detection Prevalence : 0.84973
      Balanced Accuracy : 0.61781

       'Positive' Class : 0
```

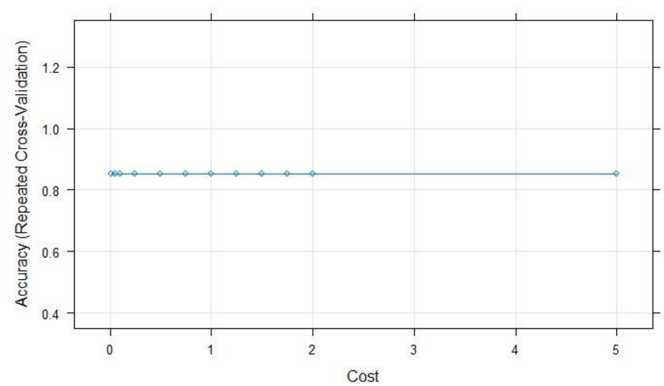
## 6. Linear Discriminant Analysis :-

```
Confusion Matrix and Statistics

      0    1
 0 1217  203
 1   25   19

               Accuracy : 0.8443
                 95% CI : (0.8247, 0.8625)
    No Information Rate : 0.8484
    P-Value [Acc > NIR] : 0.6844

                  Kappa : 0.0976

 Mcnemar's Test P-Value : <0.0000000000000002

            Sensitivity : 0.08559
            Specificity : 0.97987
         Pos Pred Value : 0.43182
         Neg Pred Value : 0.85704
             Prevalence : 0.15164
         Detection Rate : 0.01298
   Detection Prevalence : 0.03005
      Balanced Accuracy : 0.53273

       'Positive' Class : 1
```




## 7. Support Vector Machine :-

```
Confusion Matrix and Statistics

test_pred   0   1
        0 922 175
        1   0   0

               Accuracy : 0.8405
                 95% CI : (0.8174, 0.8617)
    No Information Rate : 0.8405
    P-Value [Acc > NIR] : 0.5202

                  Kappa : 0

 Mcnemar's Test P-Value : <0.0000000000000002

            Sensitivity : 1.0000
            Specificity : 0.0000
         Pos Pred Value : 0.8405
         Neg Pred Value :    NaN
             Prevalence : 0.8405
         Detection Rate : 0.8405
   Detection Prevalence : 1.0000
      Balanced Accuracy : 0.5000

       'Positive' Class : 0
```



## VI. Discussion and Recommendation

We formulated 7 different supervised learning models and created confusion matrices, Lift Charts/ ROC curves and found out accuracies for all of them. Out of all the 7 supervised learning models, our KNN model is most accurate with an accuracy of 86%.

- Men appear to be more vulnerable to coronary illness than Women. An expansion in age, number of cigarettes smoked every day and systolic Blood Pressure additionally show expanded chances of having coronary illness.
- Total cholesterol shows no huge change in the chances of getting coronary heart disease. This could be because of the presence of 'good cholesterol(HDL) while the total cholesterol was calculated. Glucose also causes a truly irrelevant change in the chances of getting coronary heart disease.
- Overall model could be improved if more amount of data is available.

## VII. Summary

The goal of our study was to forecast if a patient would get coronary heart disease in a 10-year span or not according to the traits he possesses. The early forecast of cardiovascular ailments can help in compelling high-risk patients to change their risky habits and follow a healthy routine such that any unfavorable events can be prevented. For our classification problem, we have applied kNN, Naive Bayes, Random Forest, Logistic Regression, Neural Net, Linear Discriminant Analysis and Support Vector Machine and then evaluated the performances of each model.
Out of all the 7 supervised learning models, our KNN model is most accurate with an accuracy of 86%. Changes in features like age, number of cigarettes smoked every day and systolic Blood Pressure have a directly proportional relationship with the 10 year coronary heart disease. Changes in the level total cholesterol and glucose level do not cause much changes in the chances of getting coronary heart disease.

## Appendix: R Code for use case study

```
library(stats)
heart <- read.csv("D:/Data Mining/Data Mining - Group 11/Project/framingham.csv", na.strings = "",
stringsAsFactors = FALSE)
dim(heart)
str(heart)
heart[heart == "NA"] <- NA
heart_update <- na.omit(heart)
i <- c(3,5,6,10,13,14,15)                    # Specify columns you want to change
#We can now use the apply function to change columns 2, 3, 5, 6, 10, 13, 14, and 15 to numeric:
heart_update[ , i] <- apply(heart_update[ , i], 2,       # Specify own function within apply
          function(x) as.numeric(as.character(x)))
#Let's check the classes of the variables of our data frame:
sapply(heart_update, class)                  # Get classes of all columns

names(heart_update)[1] <- "Sex_Male"
summary(heart_update)
library(plyr)
library(psych)
multi.hist(heart_update[,sapply(heart_update, is.numeric)])
library(ggplot2)
library(ggpubr)
theme_set(theme_pubr())
ggplot(heart_update, aes(TenYearCHD)) +
  geom_bar(fill = "#0073C2FF") +
  theme_pubclean()
library(psych)
pairs.panels(heart_update,
       method = "pearson", # correlation method
       hist.col = "#00AFBB",
       density = TRUE,  # show density plots
       ellipses = TRUE # show correlation ellipses
       )

#PCA
set.seed(112)
index = sample( 1:nrow(heart_update), nrow(heart_update) * 0.6, replace = FALSE )
trainset = heart_update[index, ]
test = heart_update[-index, ]
testset = test[, 1:15]
pca_trainset = trainset[, 1:15]
pca_testset = testset
pca = prcomp( pca_trainset, scale = T )
# variance
pr_var = ( pca$sdev )^2
# % of variance
prop_varex = pr_var / sum( pr_var )
# Plot
plot( prop_varex, xlab = "Principal Component",
         ylab = "Proportion of Variance Explained", type = "b" )
#Scree Plot
plot(cumsum( prop_varex ), xlab = "Principal Component",
              ylab = "Cumulative Proportion of Variance Explained", type = "b" )
```

```
# Creating a new dataset
train = data.frame(TenYearCHD = trainset$TenYearCHD, pca$x )
t = as.data.frame(predict(pca, newdata = pca_testset))
new_trainset = train[, 1:9]
new_testset =  t[, 1:8]
# Build the neural network (NN)
library( neuralnet )
n = names( new_trainset )
f = as.formula( paste( "TenYearCHD ~", paste( n[!n %in% "TenYearCHD" ], collapse = "+" ) ) )
nn = neuralnet(f, new_trainset, hidden = 4, linear.output = FALSE, threshold=0.01 )
# Plot the NN
plot(nn, rep = "best" )
# Test the resulting output
nn.results = compute(nn, new_testset)
# Results
results = data.frame( actual = test$TenYearCHD,
                     prediction = round(nn.results$net.result))
# Confusion Matrix
library(caret)
t = table(results)
print(confusionMatrix(t))
Predict=compute(nn,new_testset)
prob.result <- Predict$net.result
prob <- Predict$net.result



# Model 1: kNN
set.seed(743)
train.index <- sample(row.names(heart_update), 0.6 * dim(heart_update))
valid.index <- setdiff(row.names(heart_update), train.index)
train.df <- heart_update[train.index, ]
valid.df <- heart_update[valid.index, ]

# new patient
new.df <- data.frame(Sex_Male = 1, age = 52, education = 2, currentSmoker = 1, cigsPerDay = 20,
BPMeds = 1, prevalentStroke = 1, prevalentHyp = 1, diabetes = 1, totChol = 200, sysBP = 145, diaBP =
100, BMI = 28, heartRate = 80, glucose = 70)
# initialize normalized training, validation data, complete data frames to originals
train.norm.df <- train.df
valid.norm.df <- valid.df
heart.norm.df <- heart_update

# use preProcess() from the caret package to normalize variables.

library(caret)
norm.values <- preProcess(train.df[, 1:15], method=c("center", "scale"))
train.norm.df[, 1:15] <- predict(norm.values, train.df[, 1:15])
valid.norm.df[, 1:15] <- predict(norm.values, valid.df[, 1:15])
heart.norm.df[, 1:15] <- predict(norm.values, heart_update[, 1:15])
new.norm.df <- predict(norm.values, new.df)
# use knn() to compute knn.
# knn() is available in library FNN (provides a list of the nearest neighbors)
# and library class (allows a numerical output variable).
library(FNN)
nn <- knn(train = train.norm.df[, 1:15], test = new.norm.df, cl = train.norm.df[, 16], k = 3)
```

```
row.names(train.df)[attr(nn, "nn.index")]
nn

library(caret)
# initialize a data frame with two columns: k, and accuracy.
accuracy.df <- data.frame(k = seq(1, 45, 1), accuracy = rep(0, 45))
# compute knn for different k on validation.
valid.norm.df$TenYearCHD <- as.factor(valid.norm.df$TenYearCHD)
for(i in 1:45) {
  knn.pred <- knn(train.norm.df[, 1:15], valid.norm.df[, 1:15],
                cl = train.norm.df[, 16], k = i)
accuracy.df[i, 2] <- confusionMatrix(knn.pred, valid.norm.df[, 16])$overall[1]
}
accuracy.df
knn.pred.new <- knn(heart.norm.df[, 1:15], new.norm.df,
cl = heart.norm.df[, 16], k = 43)
row.names(train.df)[attr(nn, "nn.index")]
knn.pred.new
xtab = table(knn.pred, valid.norm.df[, 16])
print(xtab)
accuracy = sum(knn.pred == valid.norm.df[, 16])/length(valid.norm.df[, 16])
precision = xtab[1,1]/sum(xtab[,1])
recall = xtab[1,1]/sum(xtab[1,])
f = 2 * (precision * recall) / (precision + recall)
cat(paste("Accuracy:\t", format(accuracy, digits = 3), "\n",sep=" "))
cat(paste("Precision:\t", format(precision, digits = 3), "\n",sep=" "))
cat(paste("Recall:\t\t", format(recall, digits = 3), "\n",sep=" "))
cat(paste("F-measure:\t", format(f, digits = 3), "\n",sep=" "))

library(ISLR)
library(caret)

set.seed(300)
#Spliting data as training and test set. Using createDataPartition() function from caret
indxTrain <- createDataPartition(y = heart_update$TenYearCHD, p = 0.75,list = FALSE)
training <- heart_update[indxTrain,]
testing <- heart_update[-indxTrain,]
#Checking distibution in origanl data and partitioned data
prop.table(table(training$TenYearCHD)) * 100
prop.table(table(testing$TenYearCHD)) * 100
prop.table(table(heart_update$TenYearCHD)) * 100
trainX <- training[,names(training) != "TenYearCHD"]
preProcValues <- preProcess(x = trainX,method = c("center", "scale"))
preProcValues

set.seed(400)
ctrl <- trainControl(method="repeatedcv",repeats = 3) #,classProbs=TRUE,summaryFunction = twoClassSummary)
knnFit <- train(TenYearCHD ~ ., data = training, method = "knn", trControl = ctrl, preProcess = c("center","scale"), tuneLength = 20)
#Output of kNN fit
knnFit
#Plotting yields Number of Neighbours Vs accuracy (based on repeated cross validation)
plot(knnFit)
knnPredict <- predict(knnFit,newdata = testing )
#Get the confusion matrix to see accuracy value and other parameter values
```

```
#confusionMatrix(knnPredict, testing$TenYearCHD )
mean(knnPredict == testing$TenYearCHD)
#Now verifying 2 class summary function
ctrl <- trainControl(method="repeatedcv",repeats = 3,classProbs=TRUE,summaryFunction = twoClassSummary)
#knnFit <- train(TenYearCHD ~ ., data = training, method = "knn", trControl = ctrl, preProcess = c("center","scale"), tuneLength = 20)
#Output of kNN fit
knnFit
#Plotting yields Number of Neighbours Vs accuracy (based on repeated cross validation)
plot(knnFit, print.thres = 0.5, type="S")
knnPredict <- predict(knnFit,newdata = testing )
#Get the confusion matrix to see accuracy value and other parameter values
#confusionMatrix(knnPredict, testing$TenYearCHD )
mean(knnPredict == testing$TenYearCHD)
library(pROC)


## Naive bayes

library(e1071)

train.index <- sample(c(1:dim(heart_update)[1]), dim(heart_update)[1]*0.6)
train.df <- heart_update[train.index, ]
valid.df <- heart_update[-train.index, ]
train.df$age <- as.factor(train.df$age)
train.df$education <- as.factor(train.df$education)
train.df$cigsPerDay <- as.factor(train.df$cigsPerDay)
train.df$totChol <- as.factor(train.df$totChol)
train.df$sysBP <- as.factor(train.df$sysBP)
train.df$diaBP <- as.factor(train.df$diaBP)
train.df$heartRate <- as.factor(train.df$heartRate)
train.df$glucose <- as.factor(train.df$glucose)

disease.nb <- naiveBayes(TenYearCHD ~ ., data = train.df)
disease.nb
library(rpart)
library(rpart.plot)
# partition
set.seed(1)
train.index <- sample(c(1:dim(heart_update)[1]), dim(heart_update)[1]*0.6)
train.df <- heart_update[train.index, ]
valid.df <- heart_update[-train.index, ]
# classification tree
default.ct <- rpart(TenYearCHD ~ ., data = train.df, method = "class")
# plot tree
prp(default.ct, type = 1, extra = 1, under = TRUE, split.font = 1, varlen = -10)
# set argument type = "class" in predict() to generate predicted class membership.
default.ct.point.pred.train <- predict(default.ct, train.df, type = "class")
# generate confusion matrix for training data
train.df$TenYearCHD <- as.factor(train.df$TenYearCHD)
confusionMatrix(default.ct.point.pred.train, train.df$TenYearCHD)
# repeat the code for the validation set
valid.df$TenYearCHD <- as.factor(valid.df$TenYearCHD)
default.ct.point.pred.valid <- predict(default.ct, valid.df, type = "class")
confusionMatrix(default.ct.point.pred.valid, valid.df$TenYearCHD)
```

```
# argument xval refers to the number of folds to use in rpart's built-in
# cross-validation procedure
# argument cp sets the smallest value for the complexity parameter.
cv.ct <- rpart(TenYearCHD ~ ., data = train.df, method = "class", cp = 0.00001, minsplit = 5, xval = 5)
# use printcp() to print the table.
printcp(cv.ct)
```

```
## random forest
library(randomForest)
## random forest
rf <- randomForest(TenYearCHD ~ ., data = train.df, ntree = 500, mtry = 4, nodesize = 5, importance =
TRUE)
## variable importance plot
varImpPlot(rf, type = 1)
## confusion matrix
rf.pred <- predict(rf, valid.df)
confusionMatrix(rf.pred, valid.df$TenYearCHD)
require(rpart)
# Split randomly
x <- heart_update[sample(1:nrow(heart_update), nrow(heart_update), replace = F),]
x.train <- heart_update[1:floor(nrow(x)*.75), ]
x.evaluate <- heart_update[(floor(nrow(x)*.75)+1):nrow(x), ]
# Create a model using "random forest and bagging ensemble algorithms
# utilizing conditional inference trees."
require(party)
x.model <- cforest(as.factor(TenYearCHD) ~ ., data = x.train, control = cforest_unbiased(mtry = 3))
# Alternatively, use "recursive partitioning [...] in a conditional
# inference framework."
# ctree plots nicely (but cforest doesn"t plot)
# plot (x.model)
# Use the model to predict the evaluation.
x.evaluate$prediction <- predict(x.model, newdata=x.evaluate)
# Calculate the overall accuracy.
x.evaluate$correct <- x.evaluate$prediction == x.evaluate$TenYearCHD
print(paste("% of predicted classifications correct", mean(x.evaluate$correct) * 100))
# Extract the class probabilities.
x.evaluate$probabilities <- 1- unlist(treeresponse(x.model, newdata=x.evaluate),
use.names=F)[seq(1,nrow(x.evaluate)*2,2)]
# Plot the performance of the model applied to the evaluation set as
# an ROC curve.
require(ROCR)
pred <- prediction(x.evaluate$probabilities, x.evaluate$TenYearCHD)
perf <- performance(pred,"tpr","fpr")
plot(perf, main="ROC curve", colorize=T)
# And then a lift chart
perf <- performance(pred,"lift","rpp")
plot(perf, main="lift curve", colorize=T)
```

```
# run logistic regression
# partition data
set.seed(2)
train.index <- sample(c(1:dim(heart_update)[1]), dim(heart_update)[1]*0.6)
```

```
train.df <- heart_update[train.index, ]
valid.df <- heart_update[-train.index, ]
# use glm() (general linear model) with family = "binomial" to fit a logistic
# regression.
logit.reg <- glm(as.factor(TenYearCHD) ~ ., data = train.df, family = "binomial")
options(scipen=999)
summary(logit.reg)
predictTrain = predict(logit.reg, type = "response")
summary(predictTrain)
tapply(predictTrain, train.df$TenYearCHD, mean)
table(train.df$TenYearCHD, predictTrain > 0.5)
Sensitivity <- 34/335
#Sensitivity = 0.1014925
Specificity <- 1840/ 1859
#Specificity = 0.9897795
# use predict() with type = "response" to compute predicted probabilities.
logit.reg.pred <- predict(logit.reg, valid.df, type = "response")
# first 5 actual and predicted records
data.frame(actual = valid.df$TenYearCHD[1:5], predicted = logit.reg.pred[1:5])
library(gains)
gain <- gains(valid.df$TenYearCHD, logit.reg.pred, groups = length(logit.reg.pred))
# plot lift chart
plot(c(0, gain$cume.pct.of.total * sum(valid.df$TenYearCHD)) ~ c(0, gain$cume.obs), xlab = "# patients",
ylab = "Cumulative", main = "", type = "l")
lines(c(0, sum(valid.df$TenYearCHD)) ~ c(0, dim(valid.df)[1]), lty = 2)


#NeuralNet

library(neuralnet)
library(nnet)
library(caret)
# partition the data
set.seed(2)
train.index <- sample(c(1:dim(heart_update)[1]), dim(heart_update)[1]*0.6)
train.df <- heart_update[train.index, ]
valid.df <- heart_update[-train.index, ]
valid.index=setdiff(row.names(heart_update), train.index)

nn <- neuralnet(TenYearCHD ~ ., data = train.df, hidden = 2)
training.prediction = compute(nn, train.df)
training.class = apply(training.prediction$net.result, 1, which.max) - 1
training.class = as.factor(training.class)
confusionMatrix(training.class, as.factor(heart_update[train.index,]$TenYearCHD))
validation.prediction = compute(nn, valid.df)
validation.class = apply(validation.prediction$net.result,1,which.max) - 1
validation.class = as.factor(validation.class)
#confusionMatrix(validation.class, as.factor(heart_update[valid.index,]$TenYearCHD))
prob = compute(nn, valid.df[, -ncol(valid.df)] )
prob.result <- prob$net.result
detach(package:neuralnet,unload = T)
library(ROCR)
nn.pred = prediction(prob.result, valid.df$TenYearCHD)
pref <- performance(nn.pred, "tpr", "fpr")
plot(pref)
library(DiscriMiner)
```

```
da.reg <- linDA(heart_update[,1:15], heart_update[,16])
da.reg$functions
da.reg <- linDA(heart_update[, 1:15], heart_update[, 16])
# compute probabilities manually (below); or, use lda() in package MASS with predict()
propensity.risk <- exp(da.reg$scores[,2])/(exp(da.reg$scores[,1])+exp(da.reg$scores[,2]))
data.frame(Actual=heart_update$TenYearCHD,
da.reg$classification, da.reg$scores, propensity.risk=propensity.risk)
confusionMatrix(da.reg$classification, as.factor(heart_update$TenYearCHD))



## Linear Discriminant Analysis
set.seed(2)
train.index <- sample(c(1:dim(heart_update)[1]), dim(heart_update)[1]*0.6)
train.df <- heart_update[train.index, ]
valid.df <- heart_update[-train.index, ]
library(caret)
library(randomForest)
library(AUC)
library(MASS)
model.LDA <- lda(TenYearCHD~., data=train.df, na.action="na.omit")
model.LDA
pc <- predict(model.LDA, na.roughfix(valid.df))
summary(pc$class)
xtab <- table(pc$class, valid.df$TenYearCHD)
caret::confusionMatrix(xtab, positive = "1")
pb <- NULL
pb <- pc$posterior
pb <- as.data.frame(pb)
colnames(pb) <- c("X", "Y")
pred.LDA <- data.frame(valid.df$TenYearCHD, pb$Y)
colnames(pred.LDA) <- c("target","score")
pred.LDA$target <- as.factor(pred.LDA$target)
lift.LDA <- lift(target ~ score, data = pred.LDA, cuts=10, class="1")
xyplot(lift.LDA, main="LDA - Lift Chart", type=c("l","g"), lwd=2
       , scales=list(x=list(alternating=FALSE,tick.number = 10)
                     ,y=list(alternating=FALSE,tick.number = 10)))



## Support Vector Machine

library(caret)
intrain <- createDataPartition(y = heart_update$TenYearCHD, p= 0.7, list = FALSE)
training <- heart_update[intrain,]
testing <- heart_update[-intrain,]
training[["TenYearCHD"]] = factor(training[["TenYearCHD"]])
trctrl <- trainControl(method = "repeatedcv", number = 10, repeats = 3)
svm_Linear <- train(TenYearCHD ~., data = training, method = "svmLinear", trControl=trctrl, preProcess
= c("center", "scale"), tuneLength = 10)
svm_Linear
test_pred <- predict(svm_Linear, newdata = testing)
test_pred

confusionMatrix(table(test_pred, testing$TenYearCHD))
grid <- expand.grid(C = c(0,0.01, 0.05, 0.1, 0.25, 0.5, 0.75, 1, 1.25, 1.5, 1.75, 2,5))
```

```
svm_Linear_Grid <- train(TenYearCHD ~., data = training, method = "svmLinear",
trControl=trctrl, preProcess = c("center", "scale"), tuneGrid = grid, tuneLength = 10)
svm_Linear_Grid
plot(svm_Linear_Grid)
test_pred_grid <- predict(svm_Linear_Grid, newdata = testing)
test_pred_grid
confusionMatrix(table(test_pred_grid, testing$TenYearCHD))
```