Appendix

```
> #Appendix 1 (Simple Imputation GLM)
> library(dplyr)
> library(purrr)
> library(MASS)
> library(tidyr)
> library(caret)
> library(caret)
> library(class)
> library(e1071)
> library(pROC)
> library(ROCR)
> library(tidyverse)
> library(caret)
> library(glmnet)
>
> #Read the data
>                              data                                <-
read.table(file="C:/Users/isaac/Desktop/heart10.txt",header   =   T,
fileEncoding="UTF-8-BOM")
> head(data)
  Age Sex ChestPainType RestingBP Cholesterol FastingBS RestingECG
MaxHR ExerciseAngina HeartDisease
1  64   F          <NA>        95           0         1     Normal
145              N            1
2  48   M           ATA       100         159         0     Normal
NA              N            0
3  67   M           ASY       120         237         0     Normal
71              N            1
4  63   M           ASY       126           0         0         ST
NA              N            0
5  59   M           ASY       170         326         0        LVH
140              Y            1
6  49   M           ASY       130         206         0     Normal
NA              N            1
> summary(data)
      Age         Sex     ChestPainType   RestingBP      Cholesterol
FastingBS       RestingECG      MaxHR       ExerciseAngina
```

```
 Min.   :28.00   F:169   ASY :364      Min.   :  0.0   Min.   :  0.0
Min.   :0.0000   LVH   :166   Min.   : 60.0   N:482

 1st Qu.:47.00   M:631   ATA :121      1st Qu.:120.0   1st Qu.:172.0
1st Qu.:0.0000   Normal:484   1st Qu.:120.0   Y:318

 Median :54.00           NAP :157      Median :130.0   Median :223.0
Median :0.0000   ST    :150   Median :138.5

 Mean   :53.34           TA  : 34      Mean   :132.5   Mean   :197.9
Mean   :0.2275                Mean   :137.1

 3rd Qu.:60.00           NA's:124      3rd Qu.:140.0   3rd Qu.:267.0
3rd Qu.:0.0000                3rd Qu.:155.0

 Max.   :77.00                         Max.   :200.0   Max.   :603.0
Max.   :1.0000                Max.   :202.0


NA's   :128

  HeartDisease

 Min.   :0.0000

 1st Qu.:0.0000

 Median :1.0000

 Mean   :0.5569

 3rd Qu.:1.0000

 Max.   :1.0000

 NA's   :53

> str(data)

'data.frame':   800 obs. of  10 variables:

 $ Age           : int  64 48 67 63 59 49 54 58 62 32 ...

 $ Sex            : Factor w/ 2 levels "F","M": 1 2 2 2 2 2 1 2 2
1 ...

 $ ChestPainType : Factor w/ 4 levels "ASY","ATA","NAP",..: NA 2 1 1
1 1 1 1 1 2 ...

 $ RestingBP     : int  95 100 120 126 170 130 138 100 135 105 ...

 $ Cholesterol   : int  0 159 237 0 326 206 274 234 297 198 ...

 $ FastingBS     : int  1 0 0 0 0 0 0 0 0 0 ...

 $ RestingECG    : Factor w/ 3 levels "LVH","Normal",..: 2 2 2 3 1 2
2 2 2 2 ...

 $ MaxHR         : int  145 NA 71 NA 140 NA 105 NA NA NA ...

 $ ExerciseAngina: Factor w/ 2 levels "N","Y": 1 1 1 1 2 1 2 1 2
1 ...

 $ HeartDisease  : int  1 0 1 0 1 1 1 NA 1 NA ...

>
```

```
> #Change the Heartdisease as factor
> data$HeartDisease <- as.factor(data$HeartDisease)
>
> #How many missing varaibles?
> table(is.na(data))

FALSE  TRUE
 7695   305
> colSums(is.na(data))
           Age                 Sex   ChestPainType         RestingBP
Cholesterol     FastingBS    RestingECG        MaxHR ExerciseAngina

             0                 0              124                 0
0              0               0            128                0

  HeartDisease

            53
>
> #Categorical(ChestPain)
>
> data2 <- data %>% filter(!is.na(data$ChestPainType))
> summary(data2)
      Age         Sex     ChestPainType   RestingBP      Cholesterol
FastingBS       RestingECG      MaxHR       ExerciseAngina

 Min.   :28.00   F:145   ASY:364       Min.   :  0.0   Min.   :  0.0
Min.   :0.0000   LVH   :146   Min.   : 60.0   N:401

 1st Qu.:47.00   M:531   ATA:121       1st Qu.:120.0   1st Qu.:176.8
1st Qu.:0.0000   Normal:399   1st Qu.:118.0   Y:275

 Median :54.00           NAP:157       Median :130.0   Median :228.0
Median :0.0000   ST    :131   Median :135.0

 Mean   :53.54           TA : 34       Mean   :132.8   Mean   :202.4
Mean   :0.2234                Mean   :136.1

 3rd Qu.:60.00                         3rd Qu.:140.0   3rd Qu.:271.0
3rd Qu.:0.0000                3rd Qu.:154.0

 Max.   :77.00                         Max.   :200.0   Max.   :603.0
Max.   :1.0000                Max.   :202.0


NA's   :108

 HeartDisease

 0   :272
 1   :357

 NA's: 47

>

> #Continuous (MaxHR)

> mean(data2$MaxHR, na.rm = T)

[1] 136.1303

> data2$MaxHR <- ifelse(is.na(data2$MaxHR), 136.1303, data2$MaxHR)

> table(is.na(data2$MaxHR))


FALSE

  676

>

> #Categorical (HeartDisease)

> data3 <- data2 %>% filter(!is.na(data2$HeartDisease))

> summary(data3)

      Age        Sex      ChestPainType   RestingBP       Cholesterol
FastingBS       RestingECG      MaxHR       ExerciseAngina

 Min.   :29.0   F:137   ASY:340       Min.   :  0.0   Min.   :  0.0
Min.   :0.0000   LVH   :138   Min.   : 60.0   N:367

 1st Qu.:48.0   M:492   ATA:109       1st Qu.:120.0   1st Qu.:176.0
1st Qu.:0.0000   Normal:370   1st Qu.:120.0   Y:262

 Median :54.0           NAP:148       Median :130.0   Median :227.0
Median :0.0000   ST    :121   Median :136.1

 Mean   :53.9           TA : 32       Mean   :134.1   Mean   :202.3
Mean   :0.2321                Mean   :135.5

 3rd Qu.:60.0                         3rd Qu.:142.0   3rd Qu.:271.0
3rd Qu.:0.0000                3rd Qu.:150.0

 Max.   :77.0                         Max.   :200.0   Max.   :603.0
Max.   :1.0000                Max.   :195.0

 HeartDisease

 0:272

 1:357




> str(data3)
```

```
'data.frame':   629 obs. of  10 variables:
 $ Age           : int  48 67 63 59 49 54 62 39 57 63 ...
 $ Sex            : Factor w/ 2 levels "F","M": 2 2 2 2 2 1 2 2 2
2 ...
 $ ChestPainType : Factor w/ 4 levels "ASY","ATA","NAP",..: 2 1 1 1
1 1 1 2 1 1 ...
 $ RestingBP     : int  100 120 126 170 130 138 135 130 95 185 ...
 $ Cholesterol   : int  159 237 0 326 206 274 297 215 0 0 ...
 $ FastingBS     : int  0 0 0 0 0 0 0 0 1 0 ...
 $ RestingECG    : Factor w/ 3 levels "LVH","Normal",..: 2 2 3 1 2 2
2 2 2 2 ...
 $ MaxHR         : num  136 71 136 140 136 ...
 $ ExerciseAngina: Factor w/ 2 levels "N","Y": 1 1 1 2 1 2 2 1 1
2 ...
 $ HeartDisease   : Factor w/ 2 levels "0","1": 1 2 1 2 2 2 2 1 2
2 ...
> colSums(is.na(data3))
           Age                   Sex   ChestPainType         RestingBP
Cholesterol     FastingBS    RestingECG         MaxHR ExerciseAngina

             0                 0                 0                 0
0              0               0               0               0

  HeartDisease

             0

>

> #Final dataset after simple imputation

> str(data3)

'data.frame':   629 obs. of  10 variables:
 $ Age           : int  48 67 63 59 49 54 62 39 57 63 ...
 $ Sex            : Factor w/ 2 levels "F","M": 2 2 2 2 2 1 2 2 2
2 ...
 $ ChestPainType : Factor w/ 4 levels "ASY","ATA","NAP",..: 2 1 1 1
1 1 1 2 1 1 ...
 $ RestingBP     : int  100 120 126 170 130 138 135 130 95 185 ...
 $ Cholesterol   : int  159 237 0 326 206 274 297 215 0 0 ...
 $ FastingBS     : int  0 0 0 0 0 0 0 0 1 0 ...
 $ RestingECG    : Factor w/ 3 levels "LVH","Normal",..: 2 2 3 1 2 2
2 2 2 2 ...
 $ MaxHR         : num  136 71 136 140 136 ...
 $ ExerciseAngina: Factor w/ 2 levels "N","Y": 1 1 1 2 1 2 2 1 1
2 ...
 $ HeartDisease  : Factor w/ 2 levels "0","1": 1 2 1 2 2 2 2 1 2
2 ...
>
> data3$HeartDisease <- as.numeric(data3$HeartDisease) -1
> #Split the dataset No iterative
> set.seed(4052)
>
> train.index <- sample(1:nrow(data3), 0.7*nrow(data3))
>
> train.data <- data3[train.index,]
> test.data <- data3[-train.index,]
>
> #Each data structure
> str(train.data)
'data.frame':   440 obs. of  10 variables:
 $ Age           : int  53 58 50 63 55 65 55 74 57 35 ...
 $ Sex            : Factor w/ 2 levels "F","M": 2 1 2 2 1 2 2 2 2
2 ...
 $ ChestPainType : Factor w/ 4 levels "ASY","ATA","NAP",..: 2 2 2 1
1 1 2 4 2 2 ...
 $ RestingBP     : int  120 180 120 185 180 135 140 145 140 150 ...
 $ Cholesterol   : int  181 393 168 0 327 254 196 216 265 264 ...
 $ FastingBS     : int  0 0 0 0 0 0 0 1 0 0 ...
 $ RestingECG    : Factor w/ 3 levels "LVH","Normal",..: 2 2 2 2 3 1
2 2 3 2 ...
 $ MaxHR         : num  132 110 160 98 117 127 150 116 145 168 ...
 $ ExerciseAngina: Factor w/ 2 levels "N","Y": 1 2 1 2 2 1 1 2 2
1 ...
 $ HeartDisease  : num  0 1 0 1 1 1 0 1 1 0 ...
>
> str(test.data)
'data.frame':   189 obs. of  10 variables:
 $ Age           : int  67 39 39 51 39 49 68 59 51 48 ...
 $ Sex            : Factor w/ 2 levels "F","M": 2 2 1 1 2 1 2 2 2
2 ...
```

```
 $ ChestPainType : Factor w/ 4 levels "ASY","ATA","NAP",..: 1 2 3 3
3 2 1 1 1 2 ...
 $ RestingBP     : int  120 130 138 120 160 124 135 130 130 140 ...
 $ Cholesterol   : int  237 215 220 295 147 201 0 126 179 238 ...
 $ FastingBS     : int  0 0 0 0 1 0 0 0 0 0 ...
 $ RestingECG    : Factor w/ 3 levels "LVH","Normal",..: 2 2 2 1 2 2
3 2 2 2 ...
 $ MaxHR         : num  71 136 152 136 160 ...
 $ ExerciseAngina: Factor w/ 2 levels "N","Y": 1 1 1 1 1 1 2 1 1
1 ...
 $ HeartDisease  : num  1 0 0 0 0 0 1 1 0 0 ...
>
> #Full model without penalty
> logit.model <-glm(HeartDisease~.,train.data,family="binomial")
> summary(logit.model)


Call:
glm(formula = HeartDisease ~ ., family = "binomial", data =
train.data)


Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.0513  -0.5771   0.2469   0.5186   2.3617


Coefficients:
                  Estimate Std. Error z value Pr(>|z|)
(Intercept)      -0.917080   1.823037  -0.503  0.61493
Age               0.031912   0.017064   1.870  0.06147 .
SexM              1.512082   0.355855   4.249 2.15e-05 ***
ChestPainTypeATA -1.979067   0.401082  -4.934 8.04e-07 ***
ChestPainTypeNAP -1.634549   0.331245  -4.935 8.03e-07 ***
ChestPainTypeTA  -0.704029   0.507850  -1.386  0.16566
RestingBP         0.005940   0.007538   0.788  0.43069
Cholesterol      -0.001719   0.001413  -1.217  0.22364
FastingBS         1.139001   0.346812   3.284  0.00102 **
RestingECGNormal -0.058605   0.348086  -0.168  0.86630
```

```
RestingECGST     -0.427597   0.451410  -0.947  0.34351    
MaxHR            -0.014870   0.006827  -2.178  0.02940 *  
ExerciseAnginaY   1.824818   0.322959   5.650 1.60e-08 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 598.13  on 439  degrees of freedom
Residual deviance: 344.24  on 427  degrees of freedom
AIC: 370.24

Number of Fisher Scoring iterations: 5
> #Error rate of training set
> logit.prob <- predict(logit.model,type="response")
> logit.pred = rep("0", dim(train.data)[1])
> logit.pred[logit.prob > .5] = "1"
> table(logit.pred, train.data$HeartDisease)

logit.pred   0   1
         0 147  35
         1  37 221
>
>
> mean(logit.pred == train.data$HeartDisease)
[1] 0.8363636
> ER.full.train <- 1- mean(logit.pred == train.data$HeartDisease)
> ER.full.train
[1] 0.1636364
> #ROC curve and AUC of training set
> library(pROC)
> #First method
> train_roc = roc(train.data$HeartDisease ~ logit.prob, plot = TRUE,
print.auc = TRUE)
Setting levels: control = 0, case = 1
Setting direction: controls < cases
> Fullmodel.auc.train <-as.numeric(train_roc$auc)
> Fullmodel.auc.train
[1] 0.9001359
>
> #Second method
> y_obs <- as.numeric(as.character(train.data$HeartDisease))
> logit.prob <- predict(logit.model,type="response")
> pred_lm <- prediction(logit.prob, y_obs)
> performance(pred_lm, "auc")@y.values[[1]]
[1] 0.9001359
>
> y_obs <- as.numeric(as.character(train.data$HeartDisease))
> logit.prob <-
predict(logit.model,newdata=train.data,type="response")
> pred_lm <- prediction(logit.prob, y_obs)
> perf.logit <- performance(pred_lm,"tpr","fpr")
>
> logit.train <- performance(pred_lm, "auc")@y.values[[1]]
> logit.train
[1] 0.900135
> #Error rate of validation set
> logit.prob2 <- predict(logit.model,test.data,type="response")
> logit.pred2 = rep("0", dim(test.data)[1])
> logit.pred2[logit.prob2 > .5] = "1"
> table(logit.pred2, test.data$HeartDisease)

logit.pred2  0  1
          0 61 13
          1 27 88
>
>
> mean(logit.pred2 == test.data$HeartDisease)
[1] 0.7883598
> ER.full.vali <- 1- mean(logit.pred2 == test.data$HeartDisease)
> ER.full.vali
[1] 0.2116402
> logit.prob <- predict(logit.model,train.data,type="response")
> logit.pred = rep("0", dim(train.data)[1])
> logit.pred[logit.prob > .5] = "1"
> table(logit.pred, train.data$HeartDisease)

logit.pred   0   1
         0 147  35
         1  37 221
>
>
> mean(logit.pred == train.data$HeartDisease)
[1] 0.8363636
> ER.full.train <- 1- mean(logit.pred == train.data$HeartDisease)
> ER.full.train
[1] 0.1636364
>
>
>
> #ROC curve and AUC of validation set
> test_roc = roc(test.data$HeartDisease ~ logit.prob2, plot = TRUE,
print.auc = TRUE)
Setting levels: control = 0, case = 1
Setting direction: controls < cases
> Fullmodel.auc.test<- as.numeric(test_roc$auc)
> Fullmodel.auc.test
[1] 0.8756751
>
> #second method
> y_obs2 <- as.numeric(as.character(test.data$HeartDisease))
>                            logit.prob2                          <-
predict(logit.model,newdata=test.data,type="response")
> pred_lm2 <- prediction(logit.prob2, y_obs2)
> perf.logit2 <- performance(pred_lm2,"tpr","fpr")
```

```
>
> logit.test <- performance(pred_lm2, "auc")@y.values[[1]]
> logit.test
[1] 0.8756751
> grid<-10^seq(10,-2,length=100)
>
> x.train<-model.matrix(HeartDisease~.,data=train.data)[,-1]
> y.train<-train.data$HeartDisease
> y.train <- as.numeric(y.train)
>
> x.vali<-model.matrix(HeartDisease~.,data=test.data)[,-1]
> y.vali<-test.data$HeartDisease
> y.vali <- as.numeric(y.vali)
>
> str(x.vali)
 num [1:189, 1:12] 67 39 39 51 39 49 68 59 51 48 ...
 - attr(*, "dimnames")=List of 2
  ..$ : chr [1:189] "2" "8" "11" "12" ...
  ..$ : chr [1:12] "Age" "SexM" "ChestPainTypeATA"
"ChestPainTypeNAP" ...
>
> #ridge regression error rate
> ridge<- glmnet(y=y.train, x=x.train, alpha=0,
lambda=grid,family="binomial")
>
>
> set.seed(4052)
> cv_fit<-cv.glmnet(y=y.train,x=x.train, alpha = 0, nfolds=10,
lambda = grid, family="binomial")
> plot(cv_fit)
>
> opt_lambda<-cv_fit$lambda.min
> opt_lambda
[1] 0.01747528
>
> ridge<-glmnet(y=y.train,x=x.train,alpha=0,lambda=opt_lambda)
> prob.ridge <- ridge %>% predict(newx=x.train)
> predicted.ridge <- ifelse(prob.ridge >0.5 , "1","0")
>
> coef(ridge)
13 x 1 sparse Matrix of class "dgCMatrix"
                            s0
(Intercept)       0.4460684463
Age               0.0044594038
SexM              0.2034377450
ChestPainTypeATA -0.3164825956
ChestPainTypeNAP -0.2605048850
ChestPainTypeTA  -0.1006386814
RestingBP         0.0006809926
Cholesterol      -0.0002847651
FastingBS         0.1586082392
RestingECGNormal -0.0122166499
RestingECGST     -0.0446388106
MaxHR            -0.0023450265
ExerciseAnginaY   0.2837471998
>
>
> #Training set for ridge
> observed.class <- train.data$HeartDisease
> mean(predicted.ridge == observed.class)
[1] 0.8431818
> ER.ridge.train<- 1-mean(predicted.ridge == observed.class)
> ER.ridge.train
[1] 0.1568182
> #Test set for ridge
> prob.ridge.test <- ridge %>% predict(newx=x.vali)
> predicted.ridge.test <- ifelse(prob.ridge.test >0.5 , "1","0")
>
> observed.class.test <- test.data$HeartDisease
> mean(predicted.ridge.test ==observed.class.test)
```

```
[1] 0.7777778
> ER.ridge.test <- 1-mean(predicted.ridge.test ==
observed.class.test)
> ER.ridge.test
[1] 0.2222222
> #ridge regression AUC and ROC
> #train set
> pred <- prediction(prob.ridge, train.data$HeartDisease)
> perf <-performance(pred,"tpr","fpr")
>
> performance(pred,"auc")
A performance instance
  'Area under the ROC curve'
> plot(perf,colorize=TRUE, col="black") # plot ROC curve
> AUC.ridge.train <- performance(pred, "auc")@y.values[[1]]
> AUC.ridge.train
[1] 0.9003269
>
> #validation set
> pred.test <- prediction(prob.ridge.test, test.data$HeartDisease)
> perf.test <-performance(pred.test,"tpr","fpr")
>
> performance(pred.test,"auc")
A performance instance
  'Area under the ROC curve'
> plot(perf.test,colorize=TRUE, col="black") # plot ROC curve
> AUC.ridge.test <- performance(pred.test, "auc")@y.values[[1]]
> AUC.ridge.test
[1] 0.8778128
> #lasso regression error rate
> lasso<-glmnet(y=y.train,x=x.train,alpha=1,lambda=grid)
>
> set.seed(4052)
> cv_fit2<-cv.glmnet(y=y.train,x=x.train, alpha = 1,
nfolds=10,lambda = grid)
```

```
> opt_lambda2<-cv_fit2$lambda.min
> lasso<-glmnet(y=y.train,x=x.train,alpha=1,lambda=opt_lambda2)
> lasso

Call:  glmnet(x = x.train, y = y.train, alpha = 1, lambda = opt_lambda2)

  Df   %Dev  Lambda
1  9 0.4701 0.01322
>
> prob.lasso <- lasso %>% predict(newx=x.train)
> predicted.lasso <- ifelse(prob.lasso >0.5 , "1","0")
>
> observed.class2 <- train.data$HeartDisease
> mean(predicted.lasso == observed.class2)
[1] 0.8431818
> ER.lasso.train<-1-mean(predicted.lasso == observed.class2)
> ER.lasso.train
[1] 0.1568182
>
> coef(lasso)
13 x 1 sparse Matrix of class "dgCMatrix"
                            s0
(Intercept)       0.4868844667
Age               0.0041424721
SexM              0.1914235101
ChestPainTypeATA -0.2862274691
ChestPainTypeNAP -0.2267117419
ChestPainTypeTA  -0.0271286026
RestingBP         .           
Cholesterol      -0.0001705566
FastingBS         0.1439753080
RestingECGNormal  .           
RestingECGST      .           
MaxHR            -0.0022010054
```

```
ExerciseAnginaY   0.2944448857
> #Test set for lasso
> prob.lasso.test <- lasso %>% predict(newx=x.vali)
> predicted.lasso.test <- ifelse(prob.lasso.test >0.5 , "1","0")
>
> observed.class2.test <- test.data$HeartDisease
> mean(predicted.lasso.test ==observed.class2.test)
[1] 0.7883598
> ER.lasso.test<- 1-mean(predicted.lasso.test == observed.class2.test)
> ER.lasso.test
[1] 0.211640
> #lasso regression AUC and ROC
>
> #train set
> pred.lasso <- prediction(prob.lasso, train.data$HeartDisease)
> perf.lasso <-performance(pred.lasso,"tpr","fpr")
>
> performance(pred,"auc")
A performance instance
  'Area under the ROC curve'
> plot(perf.lasso,colorize=TRUE, col="black") # plot ROC curve
> AUC.lasso.train <- performance(pred.lasso, "auc")@y.values[[1]]
> AUC.lasso.train
[1] 0.8985118
>
> #test set
> pred.lasso.test <- prediction(prob.lasso.test, test.data$HeartDisease)
> perf.lasso.test <-performance(pred.lasso.test,"tpr","fpr")
>
> performance(pred.lasso.test,"auc")
A performance instance
  'Area under the ROC curve'
> plot(perf.lasso.test,colorize=TRUE, col="black") # plot ROC curve
> AUC.lasso.test <- performance(pred.lasso.test, "auc")@y.values[[1]]

> AUC.lasso.test

[1] 0.8751125

> #What is our final model ???

> #Using ER

> data.frame(model=c("Full model Train","Full model  Test","Ridge model Train", "Ridge model  Test", "Lasso model Train", "Lasso model Test"),ER=c(ER.full.train

+

,ER.full.vali,ER.ridge.train, ER.ridge.test, ER.lasso.train,ER.lasso.test))

              model        ER

1  Full model Train 0.1636364

2  Full model  Test 0.2116402

3 Ridge model Train 0.1568182

4 Ridge model  Test 0.2222222

5 Lasso model Train 0.1568182

6 Lasso model  Test 0.2116402

>

>

> #Using ROC and AUC

> data.frame(model=c("Full model Train","Full model  Test","Ridge model Train", "Ridge model  Test", "Lasso model Train", "Lasso model Test"),AUC=c(Fullmodel.auc.train

+

,Fullmodel.auc.test,AUC.ridge.train, AUC.ridge.test,AUC.lasso.train,AUC.lasso.test))

              model       AUC

1  Full model Train 0.9001359

2  Full model  Test 0.8756751

3 Ridge model Train 0.9003269

4 Ridge model  Test 0.8778128

5 Lasso model Train 0.8985118

6 Lasso model  Test 0.8751125

>

>
```

```
> #ROC graph training
> plot(perf.logit, col='black', main ="GLM ROC curve training")
> plot(perf, col='red', main ="ROC curve", add=TRUE)
> plot(perf.lasso, col='blue', main ="ROC curve",add=TRUE)
>
> abline(0,1)
> legend('bottomright', inset=.1, legend=c('Glm','Ridge','Lasso')
+        , col=c('black','red','blue'), lty=1, lwd=2 )
> #ROC graph test
> plot(perf.logit2, col='black', main ="GLM ROC curve test set")
> plot(perf.test, col='red', main ="ROC curve", add=TRUE)
> plot(perf.lasso.test, col='blue', main ="ROC curve",add=TRUE)
>
> abline(0,1)
> legend('bottomright', inset=.1, legend=c('Glm','Ridge','Lasso')
+        , col=c('black','red','blue'), lty=1, lwd=2 )
```

```
> #Appendix 2 (Iterative regression imputation GLM)

> #Read the data

>                              data                              <-
read.table(file="C:/Users/isaac/Desktop/heart10.txt",header   =   T,
fileEncoding="UTF-8-BOM")

> summary(data)

      Age          Sex      ChestPainType    RestingBP       Cholesterol
FastingBS        RestingECG       MaxHR        ExerciseAngina

 Min.   :28.00    F:169    ASY :364       Min.   :  0.0    Min.   :  0.0
Min.   :0.0000    LVH    :166    Min.   : 60.0    N:482

 1st Qu.:47.00    M:631    ATA :121       1st Qu.:120.0    1st Qu.:172.0
1st Qu.:0.0000    Normal:484    1st Qu.:120.0    Y:318

 Median :54.00             NAP :157       Median :130.0    Median :223.0
Median :0.0000    ST     :150    Median :138.5

 Mean   :53.34             TA  : 34       Mean   :132.5    Mean   :197.9
Mean   :0.2275                   Mean   :137.1

 3rd Qu.:60.00             NA's:124       3rd Qu.:140.0    3rd Qu.:267.0
3rd Qu.:0.0000                   3rd Qu.:155.0

 Max.   :77.00                            Max.   :200.0    Max.   :603.0
Max.   :1.0000                   Max.   :202.0


NA's   :128

  HeartDisease

 Min.   :0.0000

 1st Qu.:0.0000

 Median :1.0000

 Mean   :0.5569

 3rd Qu.:1.0000

 Max.   :1.0000

 NA's   :53

> str(data)

'data.frame':   800 obs. of  10 variables:

 $ Age           : int  64 48 67 63 59 49 54 58 62 32 ...

 $ Sex            : Factor w/ 2 levels "F","M": 1 2 2 2 2 2 1 2 2
1 ...

 $ ChestPainType : Factor w/ 4 levels "ASY","ATA","NAP",..: NA 2 1 1
1 1 1 1 1 2 ...

 $ RestingBP     : int  95 100 120 126 170 130 138 100 135 105 ...

 $ Cholesterol   : int  0 159 237 0 326 206 274 234 297 198 ...
 $ FastingBS     : int  1 0 0 0 0 0 0 0 0 0 ...

 $ RestingECG    : Factor w/ 3 levels "LVH","Normal",..: 2 2 2 3 1 2
2 2 2 2 ...

 $ MaxHR         : int  145 NA 71 NA 140 NA 105 NA NA NA ...

 $ ExerciseAngina: Factor w/ 2 levels "N","Y": 1 1 1 1 2 1 2 1 2
1 ...

 $ HeartDisease  : int  1 0 1 0 1 1 1 NA 1 NA ...

>

> #Change the Heartdisease as factor

> data$HeartDisease <- as.factor(data$HeartDisease)

>

> #How many missing varaibles?

> table(is.na(data))


FALSE  TRUE

 7695   305

> colSums(is.na(data))

           Age                Sex   ChestPainType         RestingBP
Cholesterol     FastingBS    RestingECG         MaxHR ExerciseAngina

             0                  0             124                 0
0              0              0             128              0

  HeartDisease

            53

> #Another dataset with Iterative regression imputation

> iter_reg_data=data

> summary(iter_reg_data)

      Age          Sex     ChestPainType   RestingBP      Cholesterol
FastingBS       RestingECG      MaxHR       ExerciseAngina

 Min.   :28.00   F:169   ASY :364      Min.   :  0.0   Min.   :  0.0
Min.   :0.0000   LVH   :166   Min.   : 60.0   N:482

 1st Qu.:47.00   M:631   ATA :121      1st Qu.:120.0   1st Qu.:172.0
1st Qu.:0.0000   Normal:484   1st Qu.:120.0   Y:318

 Median :54.00           NAP :157      Median :130.0   Median :223.0
Median :0.0000   ST    :150   Median :138.5

 Mean   :53.34           TA  : 34      Mean   :132.5   Mean   :197.9
Mean   :0.2275                Mean   :137.1

 3rd Qu.:60.00           NA's:124      3rd Qu.:140.0   3rd Qu.:267.0
3rd Qu.:0.0000                3rd Qu.:155.0
```

```
 Max.   :77.00                        Max.   :200.0   Max.   :603.0
Max.   :1.0000                Max.   :202.0

NA's   :128

 HeartDisease

 0   :331

 1   :416

 NA's: 53

> colSums(is.na(iter_reg_data))

           Age                   Sex   ChestPainType         RestingBP
Cholesterol     FastingBS    RestingECG          MaxHR ExerciseAngina

             0                     0               124                 0
0               0               0             128                0

  HeartDisease

            53

>

>
iter_reg_data$MaxHR[is.na(iter_reg_data$MaxHR)]=mean(iter_reg_data$
MaxHR,na.rm=TRUE)

>

> summary(iter_reg_data$ChestPainType)

 ASY  ATA  NAP   TA NA's

 364  121  157   34  124

>
iter_reg_data$ChestPainType[is.na(iter_reg_data$ChestPainType)]="AS
Y"

>

> summary(iter_reg_data$ChestPainType)

ASY ATA NAP  TA

488 121 157  34

> summary(iter_reg_data$HeartDisease)

   0    1 NA's

 331  416   53

> iter_reg_data$HeartDisease[is.na(iter_reg_data$HeartDisease)]="1"

> summary(iter_reg_data$HeartDisease)

  0   1

331 469

>
```

```
>

> n_iter=20

> for(i in 1:n_iter)

+ {

+   #impute Price give rest

+   m_MaxHR=lm(MaxHR~.,iter_reg_data,subset=!is.na(data$MaxHR))

+   pred_MaxHR=predict(m_MaxHR,iter_reg_data[is.na(data$MaxHR),])

+   iter_reg_data$MaxHR[is.na(data$MaxHR)]=pred_MaxHR

+

+   #impute ChestPainType given rest

+   library(nnet)

+   m_ChestPainType=multinom(ChestPainType~.,iter_reg_data,

+                      subset=!is.na(data$ChestPainType),trace=FALSE)

+
pred_ChestPainType=predict(m_ChestPainType,iter_reg_data[is.na(data
$ChestPainType),])

+
iter_reg_data$ChestPainType[is.na(data$ChestPainType)]=pred_ChestPa
inType

+   #impute HeartDisease given rest

+
m_HeartDisease=glm(HeartDisease~.,iter_reg_data,subset=!is.na(data$
HeartDisease),family="binomial")

+
pred_HeartDisease=predict(m_HeartDisease,iter_reg_data[is.na(data$H
eartDisease),],type="response")

+
iter_reg_data$HeartDisease[is.na(data$HeartDisease)]=ifelse(pred_He
artDisease >0.5, "1","0")

+ }

>

> mean(iter_reg_data$MaxHR)

[1] 137.0224

>

>

> str(iter_reg_data$HeartDisease)

 Factor w/ 2 levels "0","1": 2 1 2 1 2 2 2 2 2 1 ...

> str(data)
```

```
'data.frame':   800 obs. of  10 variables:
 $ Age           : int  64 48 67 63 59 49 54 58 62 32 ...
 $ Sex            : Factor w/ 2 levels "F","M": 1 2 2 2 2 2 1 2 2
1 ...
 $ ChestPainType : Factor w/ 4 levels "ASY","ATA","NAP",..: NA 2 1 1
1 1 1 1 1 2 ...
 $ RestingBP     : int  95 100 120 126 170 130 138 100 135 105 ...
 $ Cholesterol   : int  0 159 237 0 326 206 274 234 297 198 ...
 $ FastingBS     : int  1 0 0 0 0 0 0 0 0 0 ...
 $ RestingECG    : Factor w/ 3 levels "LVH","Normal",..: 2 2 2 3 1 2
2 2 2 2 ...
 $ MaxHR         : int  145 NA 71 NA 140 NA 105 NA NA NA ...
 $ ExerciseAngina: Factor w/ 2 levels "N","Y": 1 1 1 1 2 1 2 1 2
1 ...
 $ HeartDisease  : Factor w/ 2 levels "0","1": 2 1 2 1 2 2 2 NA 2
NA ...
>                  iter_reg_data$HeartDisease                     <-
as.numeric(iter_reg_data$HeartDisease) -1
> data$HeartDisease <- as.numeric(data$HeartDisease) -1
>
>
>
> #Compare the distributions of observed data with imputed data
>
> par(mfrow=c(1,2))
>                  hist(data$HeartDisease,breaks=20,main="Observed
data",xlab="HeartDisease",freq=FALSE)
>           hist(iter_reg_data$HeartDisease,breaks=20,main="Imputed
data",xlab="HeartDisease",freq=FALSE)
>
>
> par(mfrow=c(1,2))
> barplot(prop.table(table(data$MaxHR)),
+         main="Observed data",xlab="MaxHR")
> barplot(prop.table(table(iter_reg_data$MaxHR)),
+         main="Imputed data",xlab="MaxHR")
> par(mfrow=c(1,2))
> barplot(prop.table(table(data$ChestPainType)),
+         main="Observed data",xlab="ChestPainType")
> barplot(prop.table(table(iter_reg_data$ChestPainType)),
+         main="Imputed data",xlab="ChestPainType")
>  iter_reg_data$HeartDisease <-as.factor(iter_reg_data$HeartDisease)
> data$HeartDisease <-as.factor(data$HeartDisease)
> data3 <- iter_reg_data
> str(data3)
'data.frame':   800 obs. of  10 variables:
 $ Age           : int  64 48 67 63 59 49 54 58 62 32 ...
 $ Sex            : Factor w/ 2 levels "F","M": 1 2 2 2 2 2 1 2 2
1 ...
 $ ChestPainType : Factor w/ 4 levels "ASY","ATA","NAP",..: 1 2 1 1
1 1 1 1 1 2 ...
 $ RestingBP     : int  95 100 120 126 170 130 138 100 135 105 ...
 $ Cholesterol   : int  0 159 237 0 326 206 274 234 297 198 ...
 $ FastingBS     : int  1 0 0 0 0 0 0 0 0 0 ...
 $ RestingECG    : Factor w/ 3 levels "LVH","Normal",..: 2 2 2 3 1 2
2 2 2 2 ...
 $ MaxHR         : num  145 149 71 120 140 ...
 $ ExerciseAngina: Factor w/ 2 levels "N","Y": 1 1 1 1 2 1 2 1 2
1 ...
 $ HeartDisease  : Factor w/ 2 levels "0","1": 2 1 2 1 2 2 2 2 2
1 ...
> data3$HeartDisease <- as.numeric(data3$HeartDisease) -1
> #Split the dataset No iterative
> set.seed(4052)
>
> train.index <- sample(1:nrow(data3), 0.7*nrow(data3))
>
> train.data <- data3[train.index,]
> test.data <- data3[-train.index,]
>
> #Each data structure
> str(train.data)
'data.frame':   560 obs. of  10 variables:
 $ Age           : int  46 41 52 32 61 52 45 54 55 56 ...
 $ Sex           : Factor w/ 2 levels "F","M": 2 1 2 1 2 2 1 2 1
2 ...
 $ ChestPainType : Factor w/ 4 levels "ASY","ATA","NAP",..: 1 2 1 2
1 1 2 2 2 3 ...
 $ RestingBP     : int  140 126 160 105 105 125 180 160 110 130 ...
 $ Cholesterol   : int  311 306 246 198 0 212 295 305 344 221 ...
 $ FastingBS     : int  0 0 0 0 1 0 0 0 0 0 ...
 $ RestingECG    : Factor w/ 3 levels "LVH","Normal",..: 2 2 3 2 2 2
2 2 3 1 ...
 $ MaxHR         : num  120 163 124 166 110 ...
 $ ExerciseAngina: Factor w/ 2 levels "N","Y": 2 1 2 1 2 1 1 1 1
1 ...
 $ HeartDisease  : num  1 0 1 0 1 1 0 0 0 0 ...
>
> str(test.data)
'data.frame':   240 obs. of  10 variables:
 $ Age           : int  64 48 55 39 39 58 59 62 50 58 ...
 $ Sex           : Factor w/ 2 levels "F","M": 1 2 2 2 1 2 2 2 2
2 ...
 $ ChestPainType : Factor w/ 4 levels "ASY","ATA","NAP",..: 1 2 1 2
3 1 3 1 1 1 ...
 $ RestingBP     : int  95 100 140 130 138 120 130 138 140 116 ...
 $ Cholesterol   : int  0 159 0 215 220 0 318 204 231 0 ...
 $ FastingBS     : int  1 0 0 0 0 0 0 0 0 0 ...
 $ RestingECG    : Factor w/ 3 levels "LVH","Normal",..: 2 2 2 2 2 1
2 3 3 2 ...
 $ MaxHR         : num  145 149 83 158 152 ...
 $ ExerciseAngina: Factor w/ 2 levels "N","Y": 1 1 1 1 1 2 2 2 2
1 ...
 $ HeartDisease  : num  1 0 1 0 0 1 0 1 1 1 ...
>
> #Full model without penalty
> logit.model <-glm(HeartDisease~.,train.data,family="binomial")
> summary(logit.model)


Call:
glm(formula = HeartDisease ~ ., family = "binomial", data =
train.data)
```

```
Deviance Residuals: 
    Min       1Q   Median       3Q      Max  
-3.0086  -0.4428   0.2113   0.4920   2.4861  

Coefficients:
                  Estimate Std. Error z value Pr(>|z|)    
(Intercept)      -1.806344   1.832942  -0.985 0.324384    
Age               0.047660   0.016578   2.875 0.004042 ** 
SexM              1.664126   0.360194   4.620 3.84e-06 ***
ChestPainTypeATA -2.405186   0.411764  -5.841 5.18e-09 ***
ChestPainTypeNAP -2.101580   0.318500  -6.598 4.16e-11 ***
ChestPainTypeTA  -1.075531   0.509049  -2.113 0.034616 *  
RestingBP         0.003527   0.007298   0.483 0.628962    
Cholesterol      -0.001954   0.001382  -1.415 0.157174    
FastingBS         1.258260   0.338222   3.720 0.000199 ***
RestingECGNormal -0.494458   0.340077  -1.454 0.145957    
RestingECGST     -0.730073   0.441886  -1.652 0.098499 .  
MaxHR            -0.010120   0.006715  -1.507 0.131830    
ExerciseAnginaY   1.891039   0.296845   6.370 1.88e-10 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 767.04  on 559  degrees of freedom
Residual deviance: 401.22  on 547  degrees of freedom
AIC: 427.22

Number of Fisher Scoring iterations: 5
> 
> #Error rate of training set
> logit.prob <- predict(logit.model,type="response")
> logit.pred = rep("0", dim(train.data)[1])
> logit.pred[logit.prob > .5] = "1"
> table(logit.pred, train.data$HeartDisease)

logit.pred   0   1
         0 195  35
         1  49 281
> mean(logit.pred == train.data$HeartDisease)
[1] 0.85
> ER.full.train <- 1- mean(logit.pred == train.data$HeartDisease)
> ER.full.train
[1] 0.15
> #ROC curve and AUC of training set
> library(pROC)
> #First method
> train_roc = roc(train.data$HeartDisease ~ logit.prob, plot = TRUE,
print.auc = TRUE)
Setting levels: control = 0, case = 1
Setting direction: controls < cases
> Fullmodel.auc.train <-as.numeric(train_roc$auc)
> Fullmodel.auc.train
[1] 0.9175529
>
> #Second method
> y_obs <- as.numeric(as.character(train.data$HeartDisease))
> logit.prob <- predict(logit.model,type="response")
> pred_lm <- prediction(logit.prob, y_obs)
> performance(pred_lm, "auc")@y.values[[1]]
[1] 0.9175529
>
> y_obs <- as.numeric(as.character(train.data$HeartDisease))
>                           logit.prob                            <-
predict(logit.model,newdata=train.data,type="response")
> pred_lm <- prediction(logit.prob, y_obs)
> perf.logit <- performance(pred_lm,"tpr","fpr")
>
```

```
> logit.train <- performance(pred_lm, "auc")@y.values[[1]]
> logit.train
[1] 0.9175529
> #Error rate of validation set
> logit.prob2 <- predict(logit.model,test.data,type="response")
> logit.pred2 = rep("0", dim(test.data)[1])
> logit.pred2[logit.prob2 > .5] = "1"
> table(logit.pred2, test.data$HeartDisease)

logit.pred2   0   1
          0  90  21
          1  22 107
>
>
> mean(logit.pred2 == test.data$HeartDisease)
[1] 0.8208333
> ER.full.vali <- 1- mean(logit.pred2 == test.data$HeartDisease)
> ER.full.vali
[1] 0.1791667
> logit.prob <- predict(logit.model,train.data,type="response")
> logit.pred = rep("0", dim(train.data)[1])
> logit.pred[logit.prob > .5] = "1"
> table(logit.pred, train.data$HeartDisease)

logit.pred   0   1
         0 195  35
         1  49 281
> mean(logit.pred == train.data$HeartDisease)
[1] 0.85
> ER.full.train <- 1- mean(logit.pred == train.data$HeartDisease)
> ER.full.train
[1] 0.15
> #ROC curve and AUC of validation set
> test_roc = roc(test.data$HeartDisease ~ logit.prob2, plot = TRUE,
print.auc = TRUE)
```

```
Setting levels: control = 0, case = 1
Setting direction: controls < cases
> Fullmodel.auc.test<- as.numeric(test_roc$auc)
> Fullmodel.auc.test
[1] 0.8889509
>
> #second method
> y_obs2 <- as.numeric(as.character(test.data$HeartDisease))
> logit.prob2 <-
predict(logit.model,newdata=test.data,type="response")
> pred_lm2 <- prediction(logit.prob2, y_obs2)
> perf.logit2 <- performance(pred_lm2,"tpr","fpr")
>
> logit.test <- performance(pred_lm2, "auc")@y.values[[1]]
> logit.test
[1] 0.888950
> grid<-10^seq(10,-2,length=100)
>
> x.train<-model.matrix(HeartDisease~.,data=train.data)[,-1]
> y.train<-train.data$HeartDisease
> y.train <- as.numeric(y.train)
>
> x.vali<-model.matrix(HeartDisease~.,data=test.data)[,-1]
> y.vali<-test.data$HeartDisease
> y.vali <- as.numeric(y.vali)
>
> str(x.vali)
 num [1:240, 1:12] 64 48 55 39 39 58 59 62 50 58 ...
 - attr(*, "dimnames")=List of 2
  ..$ : chr [1:240] "1" "2" "11" "12" ...
  ..$ : chr [1:12] "Age" "SexM" "ChestPainTypeATA"
"ChestPainTypeNAP" ...
>
> #ridge regression error rate
> ridge<- glmnet(y=y.train, x=x.train, alpha=0,
lambda=grid,family="binomial")
```

```
>

>

> set.seed(4052)

>  cv_fit<-cv.glmnet(y=y.train,x=x.train,  alpha  =  0,  nfolds=10,
lambda = grid, family="binomial")

> plot(cv_fit)

>

> opt_lambda<-cv_fit$lambda.min

> opt_lambda

[1] 0.01321941

>

> ridge<-glmnet(y=y.train,x=x.train,alpha=0,lambda=opt_lambda)

> prob.ridge <- ridge %>% predict(newx=x.train)

> predicted.ridge <- ifelse(prob.ridge >0.5 , "1","0")

>

> coef(ridge)

13 x 1 sparse Matrix of class "dgCMatrix"

                            s0

(Intercept)       0.4348571451

Age               0.0053556823

SexM              0.1876128014

ChestPainTypeATA -0.3721523207

ChestPainTypeNAP -0.3323156521

ChestPainTypeTA  -0.1631364171

RestingBP         0.0002346176

Cholesterol      -0.0002766255

FastingBS         0.1582243182

RestingECGNormal -0.0563533709

RestingECGST     -0.0674523208

MaxHR            -0.0016885127

ExerciseAnginaY   0.2741063886

>

>

> #Training set for ridge

> observed.class <- train.data$HeartDisease
> mean(predicted.ridge == observed.class)
[1] 0.8571429
> ER.ridge.train<- 1-mean(predicted.ridge == observed.class)
> ER.ridge.train
[1] 0.1428571
>
>
>
> #Test set for ridge
> prob.ridge.test <- ridge %>% predict(newx=x.vali)
> predicted.ridge.test <- ifelse(prob.ridge.test >0.5 , "1","0")
>
> observed.class.test <- test.data$HeartDisease
> mean(predicted.ridge.test ==observed.class.test)
[1] 0.8125
> ER.ridge.test <- 1-mean(predicted.ridge.test ==
observed.class.test)
> ER.ridge.test
[1] 0.1875
> #rideg regression AUC and ROC
> #train set
> pred <- prediction(prob.ridge, train.data$HeartDisease)
> perf <-performance(pred,"tpr","fpr")
>
> performance(pred,"auc")
A performance instance
  'Area under the ROC curve'
> plot(perf,colorize=TRUE, col="black") # plot ROC curve
> AUC.ridge.train <- performance(pred, "auc")@y.values[[1]]
> AUC.ridge.train
[1] 0.9164894
>
> #validation set
> pred.test <- prediction(prob.ridge.test, test.data$HeartDisease)
> perf.test <-performance(pred.test,"tpr","fpr")
```

```
>
> performance(pred.test,"auc")
A performance instance
  'Area under the ROC curve'
> plot(perf.test,colorize=TRUE, col="black") # plot ROC curve
> AUC.ridge.test <- performance(pred.test, "auc")@y.values[[1]]
> AUC.ridge.test
[1] 0.894322
>
> ridge$beta
12 x 1 sparse Matrix of class "dgCMatrix"
                            s0
Age                0.0053556823
SexM               0.1876128014
ChestPainTypeATA -0.3721523207
ChestPainTypeNAP -0.3323156521
ChestPainTypeTA  -0.1631364171
RestingBP          0.0002346176
Cholesterol      -0.0002766255
FastingBS          0.1582243182
RestingECGNormal -0.0563533709
RestingECGST     -0.0674523208
MaxHR            -0.0016885127
ExerciseAnginaY    0.2741063886
> #lasso regression error rate
> lasso<-glmnet(y=y.train,x=x.train,alpha=1,lambda=grid)
>
> set.seed(4052)
>     cv_fit2<-cv.glmnet(y=y.train,x=x.train,     alpha     =     1,
nfolds=10,lambda = grid)
> opt_lambda2<-cv_fit2$lambda.min
> lasso<-glmnet(y=y.train,x=x.train,alpha=1,lambda=opt_lambda2)
> lasso


Call:   glmnet(x = x.train, y = y.train, alpha = 1, lambda =
opt_lambda2)

  Df   %Dev Lambda
1 10 0.5202   0.01
>
> prob.lasso <- lasso %>% predict(newx=x.train)
> predicted.lasso <- ifelse(prob.lasso >0.5 , "1","0")
>
> observed.class2 <- train.data$HeartDisease
> mean(predicted.lasso == observed.class2)
[1] 0.8571429
> ER.lasso.train<-1-mean(predicted.lasso == observed.class2)
> ER.lasso.train
[1] 0.1428571
>
> coef(lasso)
13 x 1 sparse Matrix of class "dgCMatrix"
                            s0
(Intercept)       0.3833651683
Age               0.0052351163
SexM              0.1778608891
ChestPainTypeATA -0.3628929574
ChestPainTypeNAP -0.3092947489
ChestPainTypeTA  -0.1072256668
RestingBP         .
Cholesterol      -0.0001854224
FastingBS         0.1464411883
RestingECGNormal -0.0062603595
RestingECGST      .
MaxHR            -0.0014925904
ExerciseAnginaY   0.2779756514
> #Test set for lasso
> prob.lasso.test <- lasso %>% predict(newx=x.vali)
> predicted.lasso.test <- ifelse(prob.lasso.test >0.5 , "1","0")
```

```
>
> observed.class2.test <- test.data$HeartDisease
> mean(predicted.lasso.test ==observed.class2.test)
[1] 0.8333333
>       ER.lasso.test<-       1-mean(predicted.lasso.test        ==
observed.class2.test)
> ER.lasso.test
[1] 0.1666667
>
>
> lasso$beta
12 x 1 sparse Matrix of class "dgCMatrix"
                            s0
Age                0.0052351163
SexM               0.1778608891
ChestPainTypeATA -0.3628929574
ChestPainTypeNAP -0.3092947489
ChestPainTypeTA  -0.1072256668
RestingBP          .
Cholesterol      -0.0001854224
FastingBS          0.1464411883
RestingECGNormal -0.0062603595
RestingECGST       .
MaxHR            -0.0014925904
ExerciseAnginaY    0.2779756514
> #lasso regression AUC and ROC
>
> #train set
> pred.lasso <- prediction(prob.lasso, train.data$HeartDisease)
> perf.lasso <-performance(pred.lasso,"tpr","fpr")
>
> performance(pred,"auc")
A performance instance
  'Area under the ROC curve'
> plot(perf.lasso,colorize=TRUE, col="black") # plot ROC curve
> AUC.lasso.train <- performance(pred.lasso, "auc")@y.values[[1]]

> AUC.lasso.train

[1] 0.9145959

>

> #test set

> pred.lasso.test <- prediction(prob.lasso.test,
test.data$HeartDisease)

> perf.lasso.test <-performance(pred.lasso.test,"tpr","fpr")

>

> performance(pred.lasso.test,"auc")

A performance instance

  'Area under the ROC curve'

> plot(perf.lasso.test,colorize=TRUE, col="black") # plot ROC curve

> AUC.lasso.test <- performance(pred.lasso.test, "auc")@y.values[[1]]

> AUC.lasso.test

[1] 0.8950195

>

> #What is our final model ???

> #Using ER

> data.frame(model=c("Full model Train","Full model  Test","Ridge
model Train", "Ridge model  Test", "Lasso model Train", "Lasso model
Test"),ER=c(ER.full.train

+

,ER.full.vali,ER.ridge.train, ER.ridge.test,
ER.lasso.train,ER.lasso.test))

              model        ER

1  Full model Train 0.1500000

2  Full model  Test 0.1791667

3 Ridge model Train 0.1428571

4 Ridge model  Test 0.1875000

5 Lasso model Train 0.1428571

6 Lasso model  Test 0.1666667

>

>

> #Using ROC and AUC

> data.frame(model=c("Full model Train","Full model  Test","Ridge
model Train", "Ridge model  Test", "Lasso model Train", "Lasso model
Test"),AUC=c(Fullmodel.auc.train
+
,Fullmodel.auc.test,AUC.ridge.train,
AUC.ridge.test,AUC.lasso.train,AUC.lasso.test))
              model       AUC
1  Full model Train 0.9175529
2  Full model  Test 0.8889509
3 Ridge model Train 0.9164894
4 Ridge model  Test 0.8943220
5 Lasso model Train 0.9145959
6 Lasso model  Test 0.8950195
>
>
> #ROC graph traiingn
> plot(perf.logit, col='black', main ="GLM ROC curve training")
> plot(perf, col='red', main ="ROC curve", add=TRUE)
> plot(perf.lasso, col='blue', main ="ROC curve",add=TRUE)
>
> abline(0,1)
> legend('bottomright', inset=.1, legend=c('Glm','Ridge','Lasso')
+        , col=c('black','red','blue'), lty=1, lwd=2 )
>
>
>
>
> #ROC graph test
> plot(perf.logit2, col='black', main ="GLM ROC curve test set")
> plot(perf.test, col='red', main ="ROC curve", add=TRUE)
> plot(perf.lasso.test, col='blue', main ="ROC curve",add=TRUE)
>
> abline(0,1)
> legend('bottomright', inset=.1, legend=c('Glm','Ridge','Lasso')
+        , col=c('black','red','blue'), lty=1, lwd=2 )
```

Appendix 3 (Simple Imputation KNN)

```
> #Read the data

>                              data                             <-
read.table(file="C:/Users/isaac/Desktop/heart10.txt",header   =   T,
fileEncoding="UTF-8-BOM")

> summary(data)

      Age          Sex      ChestPainType   RestingBP      Cholesterol
FastingBS        RestingECG       MaxHR        ExerciseAngina

 Min.   :28.00   F:169   ASY :364      Min.   :  0.0   Min.   :  0.0
Min.   :0.0000   LVH   :166   Min.   : 60.0   N:482

 1st Qu.:47.00   M:631   ATA :121      1st Qu.:120.0   1st Qu.:172.0
1st Qu.:0.0000   Normal:484   1st Qu.:120.0   Y:318

 Median :54.00           NAP :157      Median :130.0   Median :223.0
Median :0.0000   ST    :150   Median :138.5

 Mean   :53.34           TA  : 34      Mean   :132.5   Mean   :197.9
Mean   :0.2275                Mean   :137.1

 3rd Qu.:60.00           NA's:124      3rd Qu.:140.0   3rd Qu.:267.0
3rd Qu.:0.0000                3rd Qu.:155.0

 Max.   :77.00                         Max.   :200.0   Max.   :603.0
Max.   :1.0000                Max.   :202.0


NA's   :128

  HeartDisease

 Min.   :0.0000

 1st Qu.:0.0000

 Median :1.0000

 Mean   :0.5569

 3rd Qu.:1.0000

 Max.   :1.0000

 NA's   :53

> str(data)

'data.frame':   800 obs. of  10 variables:

 $ Age          : int  64 48 67 63 59 49 54 58 62 32 ...

 $ Sex            : Factor w/ 2 levels "F","M": 1 2 2 2 2 2 1 2 2
1 ...

 $ ChestPainType : Factor w/ 4 levels "ASY","ATA","NAP",..: NA 2 1 1
1 1 1 1 1 2 ...

 $ RestingBP     : int  95 100 120 126 170 130 138 100 135 105 ...
 $ Cholesterol   : int  0 159 237 0 326 206 274 234 297 198 ...
 $ FastingBS     : int  1 0 0 0 0 0 0 0 0 0 ...
 $ RestingECG    : Factor w/ 3 levels "LVH","Normal",..: 2 2 2 3 1 2
2 2 2 2 ...
 $ MaxHR         : int  145 NA 71 NA 140 NA 105 NA NA NA ...
 $ ExerciseAngina: Factor w/ 2 levels "N","Y": 1 1 1 1 2 1 2 1 2
1 ...
 $ HeartDisease  : int  1 0 1 0 1 1 1 NA 1 NA ...
>
> #Change the Heartdisease as factor
> data$HeartDisease <- as.factor(data$HeartDisease)
>
> #How many missing varaibles?
> table(is.na(data))

FALSE  TRUE
 7695   305
> colSums(is.na(data))
           Age            Sex  ChestPainType      RestingBP
Cholesterol     FastingBS     RestingECG         MaxHR ExerciseAngina
             0              0            124              0
0              0              0            128              0
  HeartDisease
            53
>
> #Categorical(ChestPain)
>
> data2 <- data %>% filter(!is.na(data$ChestPainType))
> summary(data2)
      Age         Sex     ChestPainType   RestingBP      Cholesterol
FastingBS       RestingECG      MaxHR       ExerciseAngina
 Min.   :28.00   F:145   ASY:364       Min.   :  0.0   Min.   :  0.0
Min.   :0.0000   LVH   :146   Min.   : 60.0   N:401
 1st Qu.:47.00   M:531   ATA:121       1st Qu.:120.0   1st Qu.:176.8
1st Qu.:0.0000   Normal:399   1st Qu.:118.0   Y:275
 Median :54.00           NAP:157       Median :130.0   Median :228.0
Median :0.0000   ST    :131   Median :135.0
 Mean   :53.54           TA : 34       Mean   :132.8   Mean   :202.4
Mean   :0.2234               Mean   :136.1

 3rd Qu.:60.00                        3rd Qu.:140.0   3rd Qu.:271.0
3rd Qu.:0.0000               3rd Qu.:154.0

 Max.   :77.00                        Max.   :200.0   Max.   :603.0
Max.   :1.0000               Max.   :202.0

NA's   :108
 HeartDisease
 0   :272
 1   :357
 NA's: 47
```

```
> str(data2)
'data.frame':   676 obs. of  10 variables:
 $ Age           : int  48 67 63 59 49 54 58 62 32 39 ...
 $ Sex            : Factor w/ 2 levels "F","M": 2 2 2 2 2 1 2 2 1
2 ...
 $ ChestPainType : Factor w/ 4 levels "ASY","ATA","NAP",..: 2 1 1 1
1 1 1 1 2 2 ...
 $ RestingBP     : int  100 120 126 170 130 138 100 135 105 130 ...
 $ Cholesterol   : int  159 237 0 326 206 274 234 297 198 215 ...
 $ FastingBS     : int  0 0 0 0 0 0 0 0 0 0 ...
 $ RestingECG    : Factor w/ 3 levels "LVH","Normal",..: 2 2 3 1 2 2
2 2 2 2 ...
 $ MaxHR         : int  NA 71 NA 140 NA 105 NA NA NA NA ...
 $ ExerciseAngina: Factor w/ 2 levels "N","Y": 1 1 1 2 1 2 1 2 1
1 ...
 $ HeartDisease  : Factor w/ 2 levels "0","1": 1 2 1 2 2 2 NA 2 NA
1 ...
>
>
> #Continuous (MaxHR)
> mean(data2$MaxHR, na.rm = T)
[1] 136.1303
> data2$MaxHR <- ifelse(is.na(data2$MaxHR), 136.1303, data2$MaxHR)
> table(is.na(data2$MaxHR))

FALSE
  676
>
> #Categorical (HeartDisease)
> data3 <- data2 %>% filter(!is.na(data2$HeartDisease))
> summary(data3)
      Age        Sex      ChestPainType   RestingBP       Cholesterol
FastingBS       RestingECG      MaxHR       ExerciseAngina
 Min.   :29.0   F:137   ASY:340       Min.   :  0.0   Min.   :  0.0
Min.   :0.0000   LVH   :138   Min.   : 60.0   N:367
 1st Qu.:48.0   M:492   ATA:109       1st Qu.:120.0   1st Qu.:176.0
1st Qu.:0.0000   Normal:370   1st Qu.:120.0   Y:262
 Median :54.0           NAP:148       Median :130.0   Median :227.0
Median :0.0000   ST    :121   Median :136.1
 Mean   :53.9           TA : 32       Mean   :134.1   Mean   :202.3
Mean   :0.2321                Mean   :135.5
 3rd Qu.:60.0                         3rd Qu.:142.0   3rd Qu.:271.0
3rd Qu.:0.0000                3rd Qu.:150.0
 Max.   :77.0                         Max.   :200.0   Max.   :603.0
Max.   :1.0000                Max.   :195.0
 HeartDisease
 0:272
 1:357
> str(data3)
'data.frame':   629 obs. of  10 variables:
 $ Age           : int  48 67 63 59 49 54 62 39 57 63 ...
 $ Sex             : Factor w/ 2 levels "F","M": 2 2 2 2 2 1 2 2 2
2 ...
 $ ChestPainType : Factor w/ 4 levels "ASY","ATA","NAP",..: 2 1 1 1
1 1 1 2 1 1 ...
 $ RestingBP     : int  100 120 126 170 130 138 135 130 95 185 ...
 $ Cholesterol   : int  159 237 0 326 206 274 297 215 0 0 ...
 $ FastingBS     : int  0 0 0 0 0 0 0 0 1 0 ...
 $ RestingECG    : Factor w/ 3 levels "LVH","Normal",..: 2 2 3 1 2 2
2 2 2 2 ...
 $ MaxHR         : num  136 71 136 140 136 ...
 $ ExerciseAngina: Factor w/ 2 levels "N","Y": 1 1 1 2 1 2 2 1 1
2 ...

 $ HeartDisease  : Factor w/ 2 levels "0","1": 1 2 1 2 2 2 2 1 2
2 ...

> colSums(is.na(data3))

           Age             Sex   ChestPainType       RestingBP
Cholesterol     FastingBS    RestingECG         MaxHR ExerciseAngina

             0               0               0               0
0             0              0              0              0

  HeartDisease

             0

>

> #Final dataset after simple imputation

> str(data3)

'data.frame':   629 obs. of  10 variables:

 $ Age           : int  48 67 63 59 49 54 62 39 57 63 ...

 $ Sex            : Factor w/ 2 levels "F","M": 2 2 2 2 2 1 2 2 2
2 ...

 $ ChestPainType : Factor w/ 4 levels "ASY","ATA","NAP",..: 2 1 1 1
1 1 1 2 1 1 ...

 $ RestingBP     : int  100 120 126 170 130 138 135 130 95 185 ...

 $ Cholesterol   : int  159 237 0 326 206 274 297 215 0 0 ...

 $ FastingBS     : int  0 0 0 0 0 0 0 0 1 0 ...

 $ RestingECG    : Factor w/ 3 levels "LVH","Normal",..: 2 2 3 1 2 2
2 2 2 2 ...

 $ MaxHR         : num  136 71 136 140 136 ...

 $ ExerciseAngina: Factor w/ 2 levels "N","Y": 1 1 1 2 1 2 2 1 1
2 ...

 $ HeartDisease  : Factor w/ 2 levels "0","1": 1 2 1 2 2 2 2 1 2
2 ...

>

> data3$HeartDisease <- as.numeric(data3$HeartDisease) -1

> #Split the dataset No iterative

> set.seed(4052)

> train.index <- sample(1:nrow(data3), 0.7*nrow(data3))

> train.data <- data3[train.index,]

> test.data <- data3[-train.index,]

>
```

```
> #Each data structure
> str(train.data)
'data.frame':   440 obs. of  10 variables:
 $ Age           : int  53 58 50 63 55 65 55 74 57 35 ...
 $ Sex           : Factor w/ 2 levels "F","M": 2 1 2 2 1 2 2 2 2
2 ...
 $ ChestPainType : Factor w/ 4 levels "ASY","ATA","NAP",..: 2 2 2 1
1 1 2 4 2 2 ...
 $ RestingBP     : int  120 180 120 185 180 135 140 145 140 150 ...
 $ Cholesterol   : int  181 393 168 0 327 254 196 216 265 264 ...
 $ FastingBS     : int  0 0 0 0 0 0 0 1 0 0 ...
 $ RestingECG    : Factor w/ 3 levels "LVH","Normal",..: 2 2 2 2 3 1
2 2 3 2 ...
 $ MaxHR         : num  132 110 160 98 117 127 150 116 145 168 ...
 $ ExerciseAngina: Factor w/ 2 levels "N","Y": 1 2 1 2 2 1 1 2 2
1 ...
 $ HeartDisease  : num  0 1 0 1 1 1 0 1 1 0 ...
> str(test.data)
'data.frame':   189 obs. of  10 variables:
 $ Age           : int  67 39 39 51 39 49 68 59 51 48 ...
 $ Sex           : Factor w/ 2 levels "F","M": 2 2 1 1 2 1 2 2 2
2 ...
 $ ChestPainType : Factor w/ 4 levels "ASY","ATA","NAP",..: 1 2 3 3
3 2 1 1 1 2 ...
 $ RestingBP     : int  120 130 138 120 160 124 135 130 130 140 ...
 $ Cholesterol   : int  237 215 220 295 147 201 0 126 179 238 ...
 $ FastingBS     : int  0 0 0 0 1 0 0 0 0 0 ...
 $ RestingECG    : Factor w/ 3 levels "LVH","Normal",..: 2 2 2 1 2 2
3 2 2 2 ...
 $ MaxHR         : num  71 136 152 136 160 ...
 $ ExerciseAngina: Factor w/ 2 levels "N","Y": 1 1 1 1 1 1 2 1 1
1 ...
 $ HeartDisease  : num  1 0 0 0 0 0 1 1 0 0 ...

> train.data$Age <- as.numeric(train.data$Age)
> train.data$Sex <- as.numeric(train.data$Sex)
> train.data$RestingBP <- as.numeric(train.data$RestingBP)
> train.data$FastingBS <- as.numeric(train.data$FastingBS)
> train.data$RestingECG <- as.numeric(train.data$RestingECG)
> train.data$Cholesterol <- as.numeric(train.data$Cholesterol)
> train.data$ChestPainType <- as.numeric(train.data$ChestPainType)
> train.data$ExerciseAngina <- as.numeric(train.data$ExerciseAngina)
> train.data$HeartDisease <- as.numeric(train.data$HeartDisease)

> test.data$Age <- as.numeric(test.data$Age)
> test.data$Sex <- as.numeric(test.data$Sex)
> test.data$RestingBP <- as.numeric(test.data$RestingBP)
> test.data$FastingBS <- as.numeric(test.data$FastingBS)
> test.data$RestingECG <- as.numeric(test.data$RestingECG)
> test.data$Cholesterol <- as.numeric(test.data$Cholesterol)
> test.data$ChestPainType <- as.numeric(test.data$ChestPainType)
> test.data$ExerciseAngina <- as.numeric(test.data$ExerciseAngina)
> test.data$HeartDisease <- as.numeric(test.data$HeartDisease)
>
>
> str(train.data)
'data.frame':   440 obs. of  10 variables:
 $ Age           : num  53 58 50 63 55 65 55 74 57 35 ...
 $ Sex           : num  2 1 2 2 1 2 2 2 2 2 ...
 $ ChestPainType : num  2 2 2 1 1 1 2 4 2 2 ...
 $ RestingBP     : num  120 180 120 185 180 135 140 145 140 150 ...
 $ Cholesterol   : num  181 393 168 0 327 254 196 216 265 264 ...
 $ FastingBS     : num  0 0 0 0 0 0 0 1 0 0 ...
 $ RestingECG    : num  2 2 2 2 3 1 2 2 3 2 ...
 $ MaxHR         : num  132 110 160 98 117 127 150 116 145 168 ...
 $ ExerciseAngina: num  1 2 1 2 2 1 1 2 2 1 ...
 $ HeartDisease  : num  0 1 0 1 1 1 0 1 1 0 ...
> sqrt(440)
[1] 20.97618
>
>
>                           knn3.train                           <-
knn(train.data[,1:9],train.data[,1:9],train.data[,10],k=3)
```

```
>  ER.knn3.train<-sum(knn3.train!=train.data[,10])/length(knn3.train)
> ER.knn3.train
[1] 0.1863636
>                           knn5.train                           <-
knn(train.data[,1:9],train.data[,1:9],train.data[,10],k=5)
>  ER.knn5.train<-sum(knn5.train!=train.data[,10])/length(knn5.train)
> ER.knn5.train
[1] 0.225
>                          knn10.train                           <-
knn(train.data[,1:9],train.data[,1:9],train.data[,10],k=10)
>                                                   ER.knn10.train<-
sum(knn10.train!=train.data[,10])/length(knn10.train)
> ER.knn10.train
[1] 0.275
>
>
>                          knn21.train                           <-
knn(train.data[,1:9],train.data[,1:9],train.data[,10],k=21)
>                                                   ER.knn21.train<-
sum(knn21.train!=train.data[,10])/length(knn21.train)
> ER.knn21.train
[1] 0.2568182
> str(data3)
'data.frame':   629 obs. of  10 variables:
 $ Age           : int  48 67 63 59 49 54 62 39 57 63 ...
 $ Sex            : Factor w/ 2 levels "F","M": 2 2 2 2 2 1 2 2 2
2 ...
 $ ChestPainType : Factor w/ 4 levels "ASY","ATA","NAP",..: 2 1 1 1
1 1 1 2 1 1 ...
 $ RestingBP     : int  100 120 126 170 130 138 135 130 95 185 ...
 $ Cholesterol   : int  159 237 0 326 206 274 297 215 0 0 ...
 $ FastingBS     : int  0 0 0 0 0 0 0 0 1 0 ...
 $ RestingECG    : Factor w/ 3 levels "LVH","Normal",..: 2 2 3 1 2 2
2 2 2 2 ...
 $ MaxHR         : num  136 71 136 140 136 ...
 $ ExerciseAngina: Factor w/ 2 levels "N","Y": 1 1 1 2 1 2 2 1 1
2 ...
 $ HeartDisease  : num  0 1 0 1 1 1 1 0 1 1 ...
```

```
> #Best K is sqrt(N) Test set apply
> str(train.data)
'data.frame':   440 obs. of  10 variables:
 $ Age           : num  53 58 50 63 55 65 55 74 57 35 ...
 $ Sex           : num  2 1 2 2 1 2 2 2 2 2 ...
 $ ChestPainType : num  2 2 2 1 1 1 2 4 2 2 ...
 $ RestingBP     : num  120 180 120 185 180 135 140 145 140 150 ...
 $ Cholesterol   : num  181 393 168 0 327 254 196 216 265 264 ...
 $ FastingBS     : num  0 0 0 0 0 0 0 1 0 0 ...
 $ RestingECG    : num  2 2 2 2 3 1 2 2 3 2 ...
 $ MaxHR         : num  132 110 160 98 117 127 150 116 145 168 ...
 $ ExerciseAngina: num  1 2 1 2 2 1 1 2 2 1 ...
 $ HeartDisease  : num  0 1 0 1 1 1 0 1 1 0 ...
>
> str(train.data)
'data.frame':   440 obs. of  10 variables:
 $ Age           : num  53 58 50 63 55 65 55 74 57 35 ...
 $ Sex           : num  2 1 2 2 1 2 2 2 2 2 ...
 $ ChestPainType : num  2 2 2 1 1 1 2 4 2 2 ...
 $ RestingBP     : num  120 180 120 185 180 135 140 145 140 150 ...
 $ Cholesterol   : num  181 393 168 0 327 254 196 216 265 264 ...
 $ FastingBS     : num  0 0 0 0 0 0 0 1 0 0 ...
 $ RestingECG    : num  2 2 2 2 3 1 2 2 3 2 ...
 $ MaxHR         : num  132 110 160 98 117 127 150 116 145 168 ...
 $ ExerciseAngina: num  1 2 1 2 2 1 1 2 2 1 ...
 $ HeartDisease  : num  0 1 0 1 1 1 0 1 1 0 ...
>
>                             knn3.test                          <-
knn(train.data[,1:9],test.data[,1:9],train.data[,10],k=3)
> ER.knn3.test<-sum(knn3.test!=test.data[,10])/length(knn3.test)
> ER.knn3.test
[1] 0.3703704
>                             knn5.test                          <-
knn(train.data[,1:9],test.data[,1:9],train.data[,10],k=5)
> ER.knn5.test<-sum(knn5.test!=test.data[,10])/length(knn5.test)
> ER.knn5.test
[1] 0.3597884
>                            knn10.test                           <-
knn(train.data[,1:9],test.data[,1:9],train.data[,10],k=10)
> ER.knn10.test<-sum(knn10.test!=test.data[,10])/length(knn10.test)
> ER.knn10.test
[1] 0.3492063
>                            knn21.test                           <-
knn(train.data[,1:9],test.data[,1:9],train.data[,10],k=21)
> ER.knn21.test<-sum(knn21.test!=test.data[,10])/length(knn21.test)
> ER.knn21.test
[1] 0.3544974
>
>
> data.frame(model=c("k with 3   Training","k with 3 Test set","k
with 5   Training","k with 5 Test set",
+                   "k with 10   Training","k with 10 Test set", "k
with 21   Training","k with 21 Test set"),
+
ER=c(ER.knn3.train,ER.knn3.test,ER.knn5.train,ER.knn5.test,ER.knn10
.train,ER.knn10.test,ER.knn21.train,ER.knn21.test))
                 model        ER
1  k with 3   Training 0.1863636
2    k with 3 Test set 0.3703704
3  k with 5   Training 0.2250000
4    k with 5 Test set 0.3597884
5 k with 10   Training 0.2750000
6   k with 10 Test set 0.3492063
7 k with 21   Training 0.2568182
8   k with 21 Test set 0.3544974
>
>
> #ROC curve and AUC of training set
> library(pROC)
> knn3.train <- as.numeric(knn3.train)
> knn5.train <- as.numeric(knn5.train)
> knn10.train <- as.numeric(knn10.train)
> knn21.train <- as.numeric(knn21.train)
>
> knn3.test <- as.numeric(knn3.test)
> knn5.test <- as.numeric(knn5.test)
> knn10.test <- as.numeric(knn10.test)
> knn21.test <- as.numeric(knn21.test)
> #knn 3 AUC curve
> y_obs <- train.data$HeartDisease
> knn3.train <- as.numeric(knn3.train)
> knn3.pred <- prediction(knn3.train, y_obs)
> knn3.perf <- performance(knn3.pred, "tpr", "fpr")
> plot(knn3.perf, colorize=TRUE, main="KNN 3 Training")
> AUC.knn3.train <- performance(knn3.pred, "auc")@y.values[[1]]
> AUC.knn3.train
[1] 0.803159
> y_obs <- train.data$HeartDisease
> knn5.train <- as.numeric(knn5.train)
> knn5.pred <- prediction(knn5.train, y_obs)
> knn5.perf <- performance(knn5.pred, "tpr", "fpr")
> plot(knn5.perf, colorize=TRUE, main="KNN 5 Training")
> AUC.knn5.train <- performance(knn5.pred, "auc")@y.values[[1]]
> AUC.knn5.train
[1] 0.7623132

> y_obs <- train.data$HeartDisease
> knn10.train <- as.numeric(knn10.train)
> knn10.pred <- prediction(knn10.train, y_obs)
> knn10.perf <- performance(knn10.pred, "tpr", "fpr")
> plot(knn10.perf, colorize=TRUE, main="KNN 10 Training")
> AUC.knn10.train <- performance(knn10.pred, "auc")@y.values[[1]]
> AUC.knn10.train
[1] 0.7048234
> y_obs <- train.data$HeartDisease
> knn21.train <- as.numeric(knn21.train)
> knn21.pred <- prediction(knn21.train, y_obs)
> knn21.perf <- performance(knn21.pred, "tpr", "fpr")
> plot(knn21.perf, colorize=TRUE, main="KNN 21 Training")
> AUC.knn21.train <- performance(knn21.pred, "auc")@y.values[[1]]
> AUC.knn21.train
[1] 0.7242697
> #AUC curve on testdation set
> #knn 3 AUC curve
> y_obs2 <- test.data$HeartDisease
> knn3.test <- as.numeric(knn3.test)
> knn3.pred.test <- prediction(knn3.test, y_obs2)
> knn3.perf.test <- performance(knn3.pred.test, "tpr", "fpr")
> plot(knn3.perf.test, colorize=TRUE, main="KNN 3 test")
> AUC.knn3.test <- performance(knn3.pred.test, "auc")@y.values[[1]]
> AUC.knn3.test
[1] 0.6249437
> y_obs2 <- test.data$HeartDisease
> knn5.test <- as.numeric(knn5.test)
> knn5.pred.test <- prediction(knn5.test, y_obs2)
> knn5.perf.test <- performance(knn5.pred.test, "tpr", "fpr")
> plot(knn5.perf.test, colorize=TRUE, main="KNN 5 test")
> AUC.knn5.test <- performance(knn5.pred.test, "auc")@y.values[[1]]
> AUC.knn5.test
[1] 0.6333821
> y_obs2 <- test.data$HeartDisease
> knn10.test <- as.numeric(knn10.test)
> knn10.pred.test <- prediction(knn10.test, y_obs2)
> knn10.perf.test <- performance(knn10.pred.test, "tpr", "fpr")
> plot(knn10.perf.test, colorize=TRUE, main="KNN 10 test")
> AUC.knn10.test <- performance(knn10.pred.test, "auc")@y.values[[1]]
> AUC.knn10.test
[1] 0.647671
> y_obs2 <- test.data$HeartDisease
> knn21.test <- as.numeric(knn21.test)
> knn21.pred.test <- prediction(knn21.test, y_obs2)
> knn21.perf.test <- performance(knn21.pred.test, "tpr", "fpr")
> plot(knn21.perf.test, colorize=TRUE, main="KNN 21 test")
> AUC.knn21.test <- performance(knn21.pred.test, "auc")@y.values[[1]]
> AUC.knn21.test
[1] 0.6397952
> plot(knn3.perf, col='black', main ="KNN Training ROC curve")
> plot(knn5.perf, col='red', main ="ROC curve", add=TRUE)
> plot(knn10.perf, col='blue', main ="ROC curve",add=TRUE)
> plot(knn21.perf, col='green', main ="ROC curve",add=TRUE)
> abline(0,1)
> legend('bottomright', inset=.1, legend=c('K=3','K=5','K=10','K=21')
+        , col=c('black','red','blue','green'), lty=1, lwd=2 )
> plot(knn3.perf.test, col='black', main ="KNN Test ROC curve")
> plot(knn5.perf.test, col='red', main ="ROC curve", add=TRUE)
> plot(knn10.perf.test, col='blue', main ="ROC curve",add=TRUE)
> plot(knn21.perf.test, col='green', main ="ROC curve",add=TRUE)
> abline(0,1)
> legend('bottomright', inset=.1, legend=c('K=3','K=5','K=10','K=21')
+        , col=c('black','red','blue','green'), lty=1, lwd=2 )
>
data3$Age <- as.numeric(data3$Age)
> data3$Sex <- as.numeric(data3$Sex)
> data3$RestingBP <- as.numeric(data3$RestingBP)
> data3$FastingBS <- as.numeric(data3$FastingBS)
> data3$RestingECG <- as.numeric(data3$RestingECG)
> data3$Cholesterol <- as.numeric(data3$Cholesterol)
> data3$ChestPainType <- as.numeric(data3$ChestPainType)
> data3$ExerciseAngina <- as.numeric(data3$ExerciseAngina)
>
> data3$HeartDisease <- as.numeric(data3$HeartDisease)
>
> str(data3)
'data.frame':   629 obs. of  10 variables:
 $ Age           : num  48 67 63 59 49 54 62 39 57 63 ...
 $ Sex           : num  2 2 2 2 2 1 2 2 2 2 ...
 $ ChestPainType : num  2 1 1 1 1 1 1 2 1 1 ...
 $ RestingBP     : num  100 120 126 170 130 138 135 130 95 185 ...
 $ Cholesterol   : num  159 237 0 326 206 274 297 215 0 0 ...
 $ FastingBS     : num  0 0 0 0 0 0 0 0 1 0 ...
 $ RestingECG    : num  2 2 3 1 2 2 2 2 2 2 ...
 $ MaxHR         : num  136 71 136 140 136 ...
 $ ExerciseAngina: num  1 1 1 2 1 2 2 1 1 2 ...
 $ HeartDisease  : num  0 1 0 1 1 1 1 0 1 1 ...
> set.seed(4052)
> n<-dim(data3)
> k = 10
> set.seed(4052)
> folds = createFolds(seq(1:n),k,list=FALSE)
> kcv.error = rep(0,3)
> for (i in 1:k){
+   index = unlist(folds[i],use.names = FALSE)
+   train = data3[-index,]
+   test = data3[index,]
+  winning_class3<-knn(train[,c(1:9)],train[,c(1:9)],train[,10],k=3)
+  winning_class5<-knn(train[,c(1:9)],train[,c(1:9)],train[,10],k=5)
+                                                   winning_class10<-
knn(train[,c(1:9)],train[,c(1:9)],train[,10],k=10)
+                                                   winning_class21<-
knn(train[,c(1:9)],train[,c(1:9)],train[,10],k=21)
+
+                                        kcv.error[1]          =
sum(winning_class3!=train[,10])/length(winning_class3)
+                                        kcv.error[2]          =
sum(winning_class5!=train[,10])/length(winning_class5)
+                                        kcv.error[3]          =
sum(winning_class10!=train[,10])/length(winning_class10)
+                                        kcv.error[4]          =
sum(winning_class21!=train[,10])/length(winning_class21)
+
+ }
```

```
> data.frame(k=c(3,5,10,21),CV_error  =kcv.error)
   k  CV_error
1  3 0.1894904
2  5 0.2356688
3 10 0.2882166
4 21 0.2786624
> set.seed(4052)
> n<-dim(data3)
> k = 10
> folds = createFolds(seq(1:n),k,list=TRUE)
> kcv.error = rep(0,2)
> for (i in 1:k){
+   index = unlist(folds[i],use.names = FALSE)
+   train = data3[-index,]
+   test = data3[index,]
+                                               winning_class.test3<-
knn(train[,c(1:9)],test[,c(1:9)],train[,10],k=3)
+                                               winning_class.test5<-
knn(train[,c(1:9)],test[,c(1:9)],train[,10],k=5)
+                                              winning_class.test10<-
knn(train[,c(1:9)],test[,c(1:9)],train[,10],k=10)
+                                              winning_class.test21<-
knn(train[,c(1:9)],test[,c(1:9)],train[,10],k=21)
+
+                                         kcv.error[1]            =
sum(winning_class.test3!=test[,10])/length(winning_class.test3)
+                                         kcv.error[2]            =
sum(winning_class.test5!=test[,10])/length(winning_class.test5)
+                                         kcv.error[3]            =
sum(winning_class.test10!=test[,10])/length(winning_class.test10)
+                                         kcv.error[4]            =
sum(winning_class.test21!=test[,10])/length(winning_class.test21)
+
+ }
> data.frame(k=c(3,5,10,21),CV_error =kcv.error)
   k  CV_error
1  3 0.3015873
2  5 0.3015873
```

```
3 10 0.3492063
4 21 0.2380952
```

> #Appendix 4 (Iterative regression imputation KNN)

```
> #Read the data

>                              data                              <-
read.table(file="C:/Users/isaac/Desktop/heart10.txt",header   =   T,
fileEncoding="UTF-8-BOM")

> head(data)

  Age Sex ChestPainType RestingBP Cholesterol FastingBS RestingECG
MaxHR ExerciseAngina HeartDisease

1  64   F          <NA>        95           0         1     Normal
145             N            1

2  48   M           ATA       100         159         0     Normal
NA             N            0

3  67   M           ASY       120         237         0     Normal
71             N            1

4  63   M           ASY       126           0         0         ST
NA             N            0

5  59   M           ASY       170         326         0        LVH
140             Y            1

6  49   M           ASY       130         206         0     Normal
NA             N            1

> summary(data)

      Age         Sex     ChestPainType   RestingBP      Cholesterol
FastingBS       RestingECG      MaxHR       ExerciseAngina

 Min.   :28.00   F:169   ASY :364      Min.   :  0.0   Min.   :  0.0
Min.   :0.0000   LVH   :166   Min.   : 60.0   N:482

 1st Qu.:47.00   M:631   ATA :121      1st Qu.:120.0   1st Qu.:172.0
1st Qu.:0.0000   Normal:484   1st Qu.:120.0   Y:318

 Median :54.00           NAP :157      Median :130.0   Median :223.0
Median :0.0000   ST    :150   Median :138.5

 Mean   :53.34           TA  : 34      Mean   :132.5   Mean   :197.9
Mean   :0.2275                Mean   :137.1

 3rd Qu.:60.00           NA's:124      3rd Qu.:140.0   3rd Qu.:267.0
3rd Qu.:0.0000                3rd Qu.:155.0

 Max.   :77.00                         Max.   :200.0   Max.   :603.0
Max.   :1.0000                Max.   :202.0


NA's   :128

  HeartDisease

 Min.   :0.0000

 1st Qu.:0.0000

 Median :1.0000
 Mean   :0.5569  
 3rd Qu.:1.0000  
 Max.   :1.0000  
 NA's   :53      
> str(data)
'data.frame':   800 obs. of  10 variables:
 $ Age           : int  64 48 67 63 59 49 54 58 62 32 ...
 $ Sex            : Factor w/ 2 levels "F","M": 1 2 2 2 2 2 1 2 2
1 ...
 $ ChestPainType : Factor w/ 4 levels "ASY","ATA","NAP",..: NA 2 1 1
1 1 1 1 1 2 ...
 $ RestingBP     : int  95 100 120 126 170 130 138 100 135 105 ...
 $ Cholesterol   : int  0 159 237 0 326 206 274 234 297 198 ...
 $ FastingBS     : int  1 0 0 0 0 0 0 0 0 0 ...
 $ RestingECG    : Factor w/ 3 levels "LVH","Normal",..: 2 2 2 3 1 2
2 2 2 2 ...
 $ MaxHR         : int  145 NA 71 NA 140 NA 105 NA NA NA ...
 $ ExerciseAngina: Factor w/ 2 levels "N","Y": 1 1 1 1 2 1 2 1 2
1 ...
 $ HeartDisease  : int  1 0 1 0 1 1 1 NA 1 NA ...
> 
> #Change the Heartdisease as factor
> data$HeartDisease <- as.factor(data$HeartDisease)
> 
> #How many missing varaibles?
> table(is.na(data))

FALSE  TRUE 
 7695   305 
> colSums(is.na(data))
           Age            Sex  ChestPainType      RestingBP
Cholesterol      FastingBS     RestingECG          MaxHR ExerciseAngina 
             0              0            124              0
0              0              0            128              0 
  HeartDisease 
            53 
> 
```

```
> #Another dataset with Iterative regression impuration

> iter_reg_data=data

> summary(iter_reg_data)

      Age          Sex      ChestPainType   RestingBP       Cholesterol
FastingBS        RestingECG       MaxHR        ExerciseAngina

 Min.   :28.00   F:169   ASY :364      Min.   :  0.0   Min.   :  0.0
Min.   :0.0000   LVH   :166   Min.   : 60.0   N:482

 1st Qu.:47.00   M:631   ATA :121      1st Qu.:120.0   1st Qu.:172.0
1st Qu.:0.0000   Normal:484   1st Qu.:120.0   Y:318

 Median :54.00           NAP :157      Median :130.0   Median :223.0
Median :0.0000   ST    :150   Median :138.5

 Mean   :53.34           TA  : 34      Mean   :132.5   Mean   :197.9
Mean   :0.2275                Mean   :137.1

 3rd Qu.:60.00           NA's:124      3rd Qu.:140.0   3rd Qu.:267.0
3rd Qu.:0.0000                3rd Qu.:155.0

 Max.   :77.00                         Max.   :200.0   Max.   :603.0
Max.   :1.0000                Max.   :202.0


NA's   :128

 HeartDisease

 0   :331

 1   :416

 NA's: 53
```

```
> colSums(is.na(iter_reg_data))

           Age                 Sex   ChestPainType        RestingBP
Cholesterol      FastingBS     RestingECG         MaxHR ExerciseAngina

             0                   0               124                0
0               0                0             128               0

  HeartDisease

            53

>

>
iter_reg_data$MaxHR[is.na(iter_reg_data$MaxHR)]=mean(iter_reg_data$
MaxHR,na.rm=TRUE)

>
```

```
> summary(iter_reg_data$ChestPainType)
 ASY  ATA  NAP   TA NA's
 364  121  157   34  124
>
iter_reg_data$ChestPainType[is.na(iter_reg_data$ChestPainType)]="AS
Y"
>
> summary(iter_reg_data$ChestPainType)
ASY ATA NAP  TA
488 121 157  34
> summary(iter_reg_data$HeartDisease)
   0    1 NA's
 331  416   53
> iter_reg_data$HeartDisease[is.na(iter_reg_data$HeartDisease)]="1"
> summary(iter_reg_data$HeartDisease)
  0   1
331 469
>
> n_iter=20
> for(i in 1:n_iter)
+ {
+   #impute Price give rest
+   m_MaxHR=lm(MaxHR~.,iter_reg_data,subset=!is.na(data$MaxHR))
+   pred_MaxHR=predict(m_MaxHR,iter_reg_data[is.na(data$MaxHR),])
+   iter_reg_data$MaxHR[is.na(data$MaxHR)]=pred_MaxHR
+
+   #impute ChestPainType given rest
+   library(nnet)
+   m_ChestPainType=multinom(ChestPainType~.,iter_reg_data,
+                      subset=!is.na(data$ChestPainType),trace=FALSE)
+
pred_ChestPainType=predict(m_ChestPainType,iter_reg_data[is.na(data
$ChestPainType),])
+
iter_reg_data$ChestPainType[is.na(data$ChestPainType)]=pred_ChestPa
inType
+   #impute HeartDisease given rest
+
m_HeartDisease=glm(HeartDisease~.,iter_reg_data,subset=!is.na(data$
HeartDisease),family="binomial")

+
pred_HeartDisease=predict(m_HeartDisease,iter_reg_data[is.na(data$H
eartDisease),],type="response")

+
iter_reg_data$HeartDisease[is.na(data$HeartDisease)]=ifelse(pred_He
artDisease >0.5, "1","0")

+ }

>

> mean(iter_reg_data$MaxHR)

[1] 137.0224

>

>

> str(iter_reg_data$HeartDisease)

 Factor w/ 2 levels "0","1": 2 1 2 1 2 2 2 2 2 1 ...

> str(data)

'data.frame':   800 obs. of  10 variables:

 $ Age           : int  64 48 67 63 59 49 54 58 62 32 ...

 $ Sex           : Factor w/ 2 levels "F","M": 1 2 2 2 2 2 1 2 2
1 ...

 $ ChestPainType : Factor w/ 4 levels "ASY","ATA","NAP",..: NA 2 1 1
1 1 1 1 1 2 ...

 $ RestingBP     : int  95 100 120 126 170 130 138 100 135 105 ...

 $ Cholesterol   : int  0 159 237 0 326 206 274 234 297 198 ...

 $ FastingBS     : int  1 0 0 0 0 0 0 0 0 0 ...

 $ RestingECG    : Factor w/ 3 levels "LVH","Normal",..: 2 2 2 3 1 2
2 2 2 2 ...

 $ MaxHR         : int  145 NA 71 NA 140 NA 105 NA NA NA ...

 $ ExerciseAngina: Factor w/ 2 levels "N","Y": 1 1 1 1 2 1 2 1 2
1 ...

 $ HeartDisease  : Factor w/ 2 levels "0","1": 2 1 2 1 2 2 2 NA 2
NA ...

>                    iter_reg_data$HeartDisease                    <-
as.numeric(iter_reg_data$HeartDisease) -1

> data$HeartDisease <- as.numeric(data$HeartDisease) -1

>

>  iter_reg_data$HeartDisease <-as.factor(iter_reg_data$HeartDisease)
> data$HeartDisease <-as.factor(data$HeartDisease)
> data3 <- iter_reg_data
> #Final dataset after simple imputation
> str(data3)
'data.frame':   800 obs. of  10 variables:
 $ Age           : int  64 48 67 63 59 49 54 58 62 32 ...
 $ Sex            : Factor w/ 2 levels "F","M": 1 2 2 2 2 2 1 2 2
1 ...
 $ ChestPainType : Factor w/ 4 levels "ASY","ATA","NAP",..: 1 2 1 1
1 1 1 1 1 2 ...
 $ RestingBP     : int  95 100 120 126 170 130 138 100 135 105 ...
 $ Cholesterol   : int  0 159 237 0 326 206 274 234 297 198 ...
 $ FastingBS     : int  1 0 0 0 0 0 0 0 0 0 ...
 $ RestingECG    : Factor w/ 3 levels "LVH","Normal",..: 2 2 2 3 1 2
2 2 2 2 ...
 $ MaxHR         : num  145 149 71 120 140 ...
 $ ExerciseAngina: Factor w/ 2 levels "N","Y": 1 1 1 1 2 1 2 1 2
1 ...
 $ HeartDisease  : Factor w/ 2 levels "0","1": 2 1 2 1 2 2 2 2 2
1 ...
>
> data3$HeartDisease <- as.numeric(data3$HeartDisease) -1
> #Split the dataset No iterative
> set.seed(4052)
> train.index <- sample(1:nrow(data3), 0.7*nrow(data3))
> train.data <- data3[train.index,]
> test.data <- data3[-train.index,]
> train.data$Age <- as.numeric(train.data$Age)
> train.data$Sex <- as.numeric(train.data$Sex)
> train.data$RestingBP <- as.numeric(train.data$RestingBP)
> train.data$FastingBS <- as.numeric(train.data$FastingBS)
> train.data$RestingECG <- as.numeric(train.data$RestingECG)
> train.data$Cholesterol <- as.numeric(train.data$Cholesterol)
> train.data$ChestPainType <- as.numeric(train.data$ChestPainType)
> train.data$ExerciseAngina <- as.numeric(train.data$ExerciseAngina)
> train.data$HeartDisease <- as.numeric(train.data$HeartDisease)
> test.data$Age <- as.numeric(test.data$Age)
> test.data$Sex <- as.numeric(test.data$Sex)
> test.data$RestingBP <- as.numeric(test.data$RestingBP)
> test.data$FastingBS <- as.numeric(test.data$FastingBS)
> test.data$RestingECG <- as.numeric(test.data$RestingECG)
> test.data$Cholesterol <- as.numeric(test.data$Cholesterol)
> test.data$ChestPainType <- as.numeric(test.data$ChestPainType)
> test.data$ExerciseAngina <- as.numeric(test.data$ExerciseAngina)
> test.data$HeartDisease <- as.numeric(test.data$HeartDisease)
>                           knn3.train                          <-
knn(train.data[,1:9],train.data[,1:9],train.data[,10],k=3)
>  ER.knn3.train<-sum(knn3.train!=train.data[,10])/length(knn3.train)
> ER.knn3.train
[1] 0.1857143
>                           knn5.train                          <-
knn(train.data[,1:9],train.data[,1:9],train.data[,10],k=5)
>  ER.knn5.train<-sum(knn5.train!=train.data[,10])/length(knn5.train)
> ER.knn5.train
[1] 0.2017857
>                          knn10.train                          <-
knn(train.data[,1:9],train.data[,1:9],train.data[,10],k=10)
>                                                   ER.knn10.train<-
sum(knn10.train!=train.data[,10])/length(knn10.train)
> ER.knn10.train
[1] 0.2285714
>                          knn21.train                          <-
knn(train.data[,1:9],train.data[,1:9],train.data[,10],k=23)
>                                                   ER.knn21.train<-
sum(knn21.train!=train.data[,10])/length(knn21.train)
> ER.knn21.train
[1] 0.25
> str(data3)
'data.frame':   800 obs. of  10 variables:
 $ Age           : int  64 48 67 63 59 49 54 58 62 32 ...
 $ Sex            : Factor w/ 2 levels "F","M": 1 2 2 2 2 2 1 2 2
1 ...
 $ ChestPainType : Factor w/ 4 levels "ASY","ATA","NAP",..: 1 2 1 1
1 1 1 1 1 2 ...
```

```
 $ RestingBP     : int  95 100 120 126 170 130 138 100 135 105 ...
 $ Cholesterol   : int  0 159 237 0 326 206 274 234 297 198 ...
 $ FastingBS     : int  1 0 0 0 0 0 0 0 0 0 ...
 $ RestingECG    : Factor w/ 3 levels "LVH","Normal",..: 2 2 2 3 1 2
2 2 2 2 ...
 $ MaxHR         : num  145 149 71 120 140 ...
 $ ExerciseAngina: Factor w/ 2 levels "N","Y": 1 1 1 1 2 1 2 1 2
1 ...
 $ HeartDisease  : num  1 0 1 0 1 1 1 1 1 0 ...
> #Best K is sqrt(N) Test set apply
> str(train.data)
'data.frame':   560 obs. of  10 variables:
 $ Age           : num  46 41 52 32 61 52 45 54 55 56 ...
 $ Sex           : num  2 1 2 1 2 2 1 2 1 2 ...
 $ ChestPainType : num  1 2 1 2 1 1 2 2 2 3 ...
 $ RestingBP     : num  140 126 160 105 105 125 180 160 110 130 ...
 $ Cholesterol   : num  311 306 246 198 0 212 295 305 344 221 ...
 $ FastingBS     : num  0 0 0 0 1 0 0 0 0 0 ...
 $ RestingECG    : num  2 2 3 2 2 2 2 2 3 1 ...
 $ MaxHR         : num  120 163 124 166 110 ...
 $ ExerciseAngina: num  2 1 2 1 2 1 1 1 1 1 ...
 $ HeartDisease  : num  1 0 1 0 1 1 0 0 0 0 ...
>
> str(train.data)
'data.frame':   560 obs. of  10 variables:
 $ Age           : num  46 41 52 32 61 52 45 54 55 56 ...
 $ Sex           : num  2 1 2 1 2 2 1 2 1 2 ...
 $ ChestPainType : num  1 2 1 2 1 1 2 2 2 3 ...
 $ RestingBP     : num  140 126 160 105 105 125 180 160 110 130 ...
 $ Cholesterol   : num  311 306 246 198 0 212 295 305 344 221 ...
 $ FastingBS     : num  0 0 0 0 1 0 0 0 0 0 ...
 $ RestingECG    : num  2 2 3 2 2 2 2 2 3 1 ...
 $ MaxHR         : num  120 163 124 166 110 ...
 $ ExerciseAngina: num  2 1 2 1 2 1 1 1 1 1 ...
 $ HeartDisease  : num  1 0 1 0 1 1 0 0 0 0 ...
```

```
>

>                            knn3.test                         <-
knn(train.data[,1:9],test.data[,1:9],train.data[,10],k=3)

> ER.knn3.test<-sum(knn3.test!=test.data[,10])/length(knn3.test)

> ER.knn3.test

[1] 0.3375

knn5.test                                                      <-
knn(train.data[,1:9],test.data[,1:9],train.data[,10],k=5)

> ER.knn5.test<-sum(knn5.test!=test.data[,10])/length(knn5.test)

> ER.knn5.test

[1] 0.316666

>                           knn10.test                         <-
knn(train.data[,1:9],test.data[,1:9],train.data[,10],k=10)

> ER.knn10.test<-sum(knn10.test!=test.data[,10])/length(knn10.test)

> ER.knn10.test

[1] 0.275

>                           knn21.test                         <-
knn(train.data[,1:9],test.data[,1:9],train.data[,10],k=23)

> ER.knn21.test<-sum(knn21.test!=test.data[,10])/length(knn21.test)

> ER.knn21.test

[1] 0.3041667

> data.frame(model=c("k with 3   Training","k with 3 Test set","k
with 5   Training","k with 5 Test set",

+                    "k with 10   Training","k with 10 Test set", "k
with 21   Training","k with 21 Test set"),

+
ER=c(ER.knn3.train,ER.knn3.test,ER.knn5.train,ER.knn5.test,ER.knn10
.train,ER.knn10.test,ER.knn21.train,ER.knn21.test))

                 model        ER

1  k with 3   Training 0.1857143

2    k with 3 Test set 0.3375000

3  k with 5   Training 0.2017857

4    k with 5 Test set 0.3166667

5 k with 10   Training 0.2285714

6   k with 10 Test set 0.2750000

7 k with 21   Training 0.2500000

8   k with 21 Test set 0.3041667

> #ROC curve and AUC of training set
> library(pROC)
> knn3.train <- as.numeric(knn3.train)
> knn5.train <- as.numeric(knn5.train)
> knn10.train <- as.numeric(knn10.train)
> knn21.train <- as.numeric(knn21.train)
>
> knn3.test <- as.numeric(knn3.test)
> knn5.test <- as.numeric(knn5.test)
> knn10.test <- as.numeric(knn10.test)
> knn21.test <- as.numeric(knn21.test)
> #knn 3 AUC curve
> y_obs <- train.data$HeartDisease
> knn3.train <- as.numeric(knn3.train)
> knn3.pred <- prediction(knn3.train, y_obs)
> knn3.perf <- performance(knn3.pred, "tpr", "fpr")
> plot(knn3.perf, colorize=TRUE, main="KNN 3 Training")
> AUC.knn3.train <- performance(knn3.pred, "auc")@y.values[[1]]
> AUC.knn3.train
[1] 0.8088296
>
>
> y_obs <- train.data$HeartDisease
> knn5.train <- as.numeric(knn5.train)
> knn5.pred <- prediction(knn5.train, y_obs)
> knn5.perf <- performance(knn5.pred, "tpr", "fpr")
> plot(knn5.perf, colorize=TRUE, main="KNN 5 Training")
> AUC.knn5.train <- performance(knn5.pred, "auc")@y.values[[1]]
> AUC.knn5.train
[1] 0.7931884
> y_obs <- train.data$HeartDisease
> knn10.train <- as.numeric(knn10.train)
> knn10.pred <- prediction(knn10.train, y_obs)
> knn10.perf <- performance(knn10.pred, "tpr", "fpr")
> plot(knn10.perf, colorize=TRUE, main="KNN 10 Training")
> AUC.knn10.train <- performance(knn10.pred, "auc")@y.values[[1]]
> AUC.knn10.train
[1] 0.7671197

> y_obs <- train.data$HeartDisease
> knn21.train <- as.numeric(knn21.train)
> knn21.pred <- prediction(knn21.train, y_obs)
> knn21.perf <- performance(knn21.pred, "tpr", "fpr")
> plot(knn21.perf, colorize=TRUE, main="KNN 21 Training")
> AUC.knn21.train <- performance(knn21.pred, "auc")@y.values[[1]]
> AUC.knn21.train
[1] 0.7457979
> sqrt(460)
[1] 21.44761
> #AUC curve on testdation set
> #knn 3 AUC curve
> y_obs2 <- test.data$HeartDisease
> knn3.test <- as.numeric(knn3.test)
> knn3.pred.test <- prediction(knn3.test, y_obs2)
> knn3.perf.test <- performance(knn3.pred.test, "tpr", "fpr")
> plot(knn3.perf.test, colorize=TRUE, main="KNN 3 test")
> AUC.knn3.test <- performance(knn3.pred.test, "auc")@y.values[[1]]
> AUC.knn3.test
[1] 0.6573661

> > y_obs2 <- test.data$HeartDisease
> knn5.test <- as.numeric(knn5.test)
> knn5.pred.test <- prediction(knn5.test, y_obs2)
> knn5.perf.test <- performance(knn5.pred.test, "tpr", "fpr")
> plot(knn5.perf.test, colorize=TRUE, main="KNN 5 test")
> AUC.knn5.test <- performance(knn5.pred.test, "auc")@y.values[[1]]
> AUC.knn5.test
[1] 0.6774554
> y_obs2 <- test.data$HeartDisease
> knn10.test <- as.numeric(knn10.test)
> knn10.pred.test <- prediction(knn10.test, y_obs2)
> knn10.perf.test <- performance(knn10.pred.test, "tpr", "fpr")
> plot(knn10.perf.test, colorize=TRUE, main="KNN 10 test")
> AUC.knn10.test <- performance(knn10.pred.test, "auc")@y.values[[1]]
> AUC.knn10.test
[1] 0.7215402
> y_obs2 <- test.data$HeartDisease
> knn21.test <- as.numeric(knn21.test)
> knn21.pred.test <- prediction(knn21.test, y_obs2)
> knn21.perf.test <- performance(knn21.pred.test, "tpr", "fpr")
> plot(knn21.perf.test, colorize=TRUE, main="KNN 21 test")
> AUC.knn21.test <- performance(knn21.pred.test, "auc")@y.values[[1]]
> AUC.knn21.test
[1] 0.6925223
> plot(knn3.perf, col='black', main ="KNN Training ROC curve")
> plot(knn5.perf, col='red', main ="ROC curve", add=TRUE)
> plot(knn10.perf, col='blue', main ="ROC curve",add=TRUE)
> plot(knn21.perf, col='green', main ="ROC curve",add=TRUE)
> abline(0,1)
> legend('bottomright', inset=.1, legend=c('K=3','K=5','K=10','K=21')
+        , col=c('black','red','blue','green'), lty=1, lwd=2 )
> plot(knn3.perf.test, col='black', main ="KNN Test ROC curve")
> plot(knn5.perf.test, col='red', main ="ROC curve", add=TRUE)
> plot(knn10.perf.test, col='blue', main ="ROC curve",add=TRUE)
> plot(knn21.perf.test, col='green', main ="ROC curve",add=TRUE)
> abline(0,1)
> legend('bottomright', inset=.1, legend=c('K=3','K=5','K=10','K=21')
+        , col=c('black','red','blue','green'), lty=1, lwd=2 )
> data3$Age <- as.numeric(data3$Age)
> data3$Sex <- as.numeric(data3$Sex)
> data3$RestingBP <- as.numeric(data3$RestingBP)
> data3$FastingBS <- as.numeric(data3$FastingBS)
> data3$RestingECG <- as.numeric(data3$RestingECG)
> data3$Cholesterol <- as.numeric(data3$Cholesterol)
> data3$ChestPainType <- as.numeric(data3$ChestPainType)
> data3$ExerciseAngina <- as.numeric(data3$ExerciseAngina)
>
> data3$HeartDisease <- as.numeric(data3$HeartDisease)
>
> str(data3)
'data.frame':   800 obs. of  10 variables:
 $ Age           : num  64 48 67 63 59 49 54 58 62 32 ...
 $ Sex           : num  1 2 2 2 2 2 1 2 2 1 ...
 $ ChestPainType : num  1 2 1 1 1 1 1 1 1 2 ...
 $ RestingBP     : num  95 100 120 126 170 130 138 100 135 105 ...
 $ Cholesterol   : num  0 159 237 0 326 206 274 234 297 198 ...
 $ FastingBS     : num  1 0 0 0 0 0 0 0 0 0 ...
 $ RestingECG    : num  2 2 2 3 1 2 2 2 2 2 ...
 $ MaxHR         : num  145 149 71 120 140 ...
 $ ExerciseAngina: num  1 1 1 1 2 1 2 1 2 1 ...
 $ HeartDisease  : num  1 0 1 0 1 1 1 1 1 0 ...
> str(data3)
'data.frame':   800 obs. of  10 variables:
 $ Age           : num  64 48 67 63 59 49 54 58 62 32 ...
 $ Sex           : num  1 2 2 2 2 2 1 2 2 1 ...
 $ ChestPainType : num  1 2 1 1 1 1 1 1 1 2 ...
 $ RestingBP     : num  95 100 120 126 170 130 138 100 135 105 ...
 $ Cholesterol   : num  0 159 237 0 326 206 274 234 297 198 ...
 $ FastingBS     : num  1 0 0 0 0 0 0 0 0 0 ...
 $ RestingECG    : num  2 2 2 3 1 2 2 2 2 2 ...
 $ MaxHR         : num  145 149 71 120 140 ...
 $ ExerciseAngina: num  1 1 1 1 2 1 2 1 2 1 ...
 $ HeartDisease  : num  1 0 1 0 1 1 1 1 1 0 ...
> str(train.data)
'data.frame':   560 obs. of  10 variables:
 $ Age           : num  46 41 52 32 61 52 45 54 55 56 ...
 $ Sex           : num  2 1 2 1 2 2 1 2 1 2 ...
 $ ChestPainType : num  1 2 1 2 1 1 2 2 2 3 ...
 $ RestingBP     : num  140 126 160 105 105 125 180 160 110 130 ...
 $ Cholesterol   : num  311 306 246 198 0 212 295 305 344 221 ...
 $ FastingBS     : num  0 0 0 0 1 0 0 0 0 0 ...
 $ RestingECG    : num  2 2 3 2 2 2 2 2 3 1 ...
 $ MaxHR         : num  120 163 124 166 110 ...
 $ ExerciseAngina: num  2 1 2 1 2 1 1 1 1 1 ...
 $ HeartDisease  : num  1 0 1 0 1 1 0 0 0 0 ...
> sqrt(560)
[1] 23.66432
> set.seed(4052)
> n<-dim(data3)
> k = 5
> set.seed(4052)
> folds = createFolds(seq(1:n),k,list=FALSE)
> kcv.error = rep(0,3)
> for (i in 1:k){
+   index = unlist(folds[i],use.names = FALSE)
+   train = data3[-index,]
+   test = data3[index,]
+  winning_class3<-knn(train[,c(1:9)],train[,c(1:9)],train[,10],k=3)
+  winning_class5<-knn(train[,c(1:9)],train[,c(1:9)],train[,10],k=5)
+                                                  winning_class10<-
knn(train[,c(1:9)],train[,c(1:9)],train[,10],k=10)
+                                                  winning_class23<-
knn(train[,c(1:9)],train[,c(1:9)],train[,10],k=23)
+
+                                         kcv.error[1]           =
sum(winning_class3!=train[,10])/length(winning_class3)
+                                         kcv.error[2]           =
sum(winning_class5!=train[,10])/length(winning_class5)
+                                         kcv.error[3]           =
sum(winning_class10!=train[,10])/length(winning_class10)
+                                         kcv.error[4]           =
sum(winning_class23!=train[,10])/length(winning_class23)
+
+ }
```

```
> data.frame(k=c(3,5,10,23),CV_error  =kcv.error)
   k  CV_error
1  3 0.1764706
2  5 0.2177722
3 10 0.2490613
4 23 0.2565707
> set.seed(4052)
> n<-dim(data3)
> k = 10
> folds = createFolds(seq(1:n),k,list=TRUE)
> kcv.error = rep(0,2)
> for (i in 1:k){
+   index = unlist(folds[i],use.names = FALSE)
+   train = data3[-index,]
+   test = data3[index,]
+                                               winning_class.test3<-
knn(train[,c(1:9)],test[,c(1:9)],train[,10],k=3)
+                                               winning_class.test5<-
knn(train[,c(1:9)],test[,c(1:9)],train[,10],k=5)
+                                              winning_class.test10<-
knn(train[,c(1:9)],test[,c(1:9)],train[,10],k=10)
+                                              winning_class.test21<-
knn(train[,c(1:9)],test[,c(1:9)],train[,10],k=23)
+
+                                         kcv.error[1]            =
sum(winning_class.test3!=test[,10])/length(winning_class.test3)
+                                         kcv.error[2]            =
sum(winning_class.test5!=test[,10])/length(winning_class.test5)
+                                         kcv.error[3]            =
sum(winning_class.test10!=test[,10])/length(winning_class.test10)
+                                         kcv.error[4]            =
sum(winning_class.test21!=test[,10])/length(winning_class.test21)
+
+ }
> data.frame(k=c(3,5,10,21),CV_error =kcv.error)
   k CV_error
1  3   0.3375
2  5   0.3125
```

```
3 10   0.3250
4 21   0.3125

> data.frame(model=c("k with 3   Training","k with 3 Test set","k
with 5   Training","k with 5 Test set",
+                    "k with 10   Training","k with 10 Test set", "k
with 21   Training","k with 21 Test set"),
+
ER=c(ER.knn3.train,ER.knn3.test,ER.knn5.train,ER.knn5.test,ER.knn10
.train,ER.knn10.test,ER.knn21.train,ER.knn21.test))
                 model        ER
1  k with 3   Training 0.1857143
2    k with 3 Test set 0.3375000
3  k with 5   Training 0.2017857
4    k with 5 Test set 0.3166667
5 k with 10   Training 0.2285714
6   k with 10 Test set 0.2750000
7 k with 21   Training 0.2500000
8   k with 21 Test set 0.3041667
> data.frame(model=c("k with 3   Training","k with 3 Test set","k
with 5   Training","k with 5 Test set",
+                    "k with 10   Training","k with 10 Test set", "k
with 21   Training","k with 21 Test set"),
+            AUC=c(AUC.knn3.train,
+                  AUC.knn3.test,AUC.knn5.train,
+                  AUC.knn5.test,AUC.knn10.train,
+                  AUC.knn10.test,AUC.knn21.train,
+                  AUC.knn21.test))
                 model       AUC
1  k with 3   Training 0.8088296
2    k with 3 Test set 0.6573661
3  k with 5   Training 0.7931884
4    k with 5 Test set 0.6774554
5 k with 10   Training 0.7671197
6   k with 10 Test set 0.7215402
7 k with 21   Training 0.7457979
8   k with 21 Test set 0.6925223
```

> #Appendix 5(Simple Imputation Random Forest)

```
> #Read the data

>                              data                              <-
read.table(file="C:/Users/isaac/Desktop/heart10.txt",header   =   T,
fileEncoding="UTF-8-BOM")

> summary(data)

      Age          Sex      ChestPainType    RestingBP       Cholesterol
FastingBS        RestingECG       MaxHR        ExerciseAngina

 Min.   :28.00   F:169   ASY :364      Min.   :  0.0   Min.   :  0.0
Min.   :0.0000   LVH   :166   Min.   : 60.0   N:482

 1st Qu.:47.00   M:631   ATA :121      1st Qu.:120.0   1st Qu.:172.0
1st Qu.:0.0000   Normal:484   1st Qu.:120.0   Y:318

 Median :54.00           NAP :157      Median :130.0   Median :223.0
Median :0.0000   ST    :150   Median :138.5

 Mean   :53.34           TA  : 34      Mean   :132.5   Mean   :197.9
Mean   :0.2275                Mean   :137.1

 3rd Qu.:60.00           NA's:124      3rd Qu.:140.0   3rd Qu.:267.0
3rd Qu.:0.0000                3rd Qu.:155.0

 Max.   :77.00                         Max.   :200.0   Max.   :603.0
Max.   :1.0000                Max.   :202.0


NA's   :128

  HeartDisease

 Min.   :0.0000

 1st Qu.:0.0000

 Median :1.0000

 Mean   :0.5569

 3rd Qu.:1.0000

 Max.   :1.0000

 NA's   :53

> str(data)

'data.frame':   800 obs. of  10 variables:

 $ Age          : int  64 48 67 63 59 49 54 58 62 32 ...

 $ Sex           : Factor w/ 2 levels "F","M": 1 2 2 2 2 2 1 2 2
1 ...

 $ ChestPainType : Factor w/ 4 levels "ASY","ATA","NAP",..: NA 2 1 1
1 1 1 1 1 2 ...
```

```
 $ RestingBP     : int  95 100 120 126 170 130 138 100 135 105 ...
 $ Cholesterol   : int  0 159 237 0 326 206 274 234 297 198 ...
 $ FastingBS     : int  1 0 0 0 0 0 0 0 0 0 ...
 $ RestingECG    : Factor w/ 3 levels "LVH","Normal",..: 2 2 2 3 1 2
2 2 2 2 ...
 $ MaxHR         : int  145 NA 71 NA 140 NA 105 NA NA NA ...
 $ ExerciseAngina: Factor w/ 2 levels "N","Y": 1 1 1 1 2 1 2 1 2
1 ...
 $ HeartDisease  : int  1 0 1 0 1 1 1 NA 1 NA ...
>
> #Change the Heartdisease as factor
> data$HeartDisease <- as.factor(data$HeartDisease)
>
> #How many missing varaibles?
> table(is.na(data))

FALSE  TRUE
 7695   305
> colSums(is.na(data))
           Age            Sex  ChestPainType      RestingBP
Cholesterol     FastingBS    RestingECG         MaxHR ExerciseAngina
             0              0            124              0
0              0             0            128              0
  HeartDisease
            53
>
> #Categorical(ChestPain)
>
> data2 <- data %>% filter(!is.na(data$ChestPainType))
> summary(data2)
      Age         Sex     ChestPainType   RestingBP      Cholesterol
FastingBS       RestingECG      MaxHR       ExerciseAngina
 Min.   :28.00   F:145   ASY:364       Min.   :  0.0   Min.   :  0.0
Min.   :0.0000   LVH   :146   Min.   : 60.0   N:401
 1st Qu.:47.00   M:531   ATA:121       1st Qu.:120.0   1st Qu.:176.8
1st Qu.:0.0000   Normal:399   1st Qu.:118.0   Y:275
 Median :54.00           NAP:157       Median :130.0   Median :228.0
Median :0.0000   ST    :131   Median :135.0
```

```
 Mean   :53.54          TA : 34       Mean   :132.8   Mean   :202.4
Mean   :0.2234               Mean   :136.1

 3rd Qu.:60.00                        3rd Qu.:140.0   3rd Qu.:271.0
3rd Qu.:0.0000               3rd Qu.:154.0

 Max.   :77.00                        Max.   :200.0   Max.   :603.0
Max.   :1.0000               Max.   :202.0


NA's   :108

 HeartDisease

 0   :272

 1   :357

 NA's: 47




> str(data2)

'data.frame':   676 obs. of  10 variables:

 $ Age           : int  48 67 63 59 49 54 58 62 32 39 ...

 $ Sex            : Factor w/ 2 levels "F","M": 2 2 2 2 2 1 2 2 1
2 ...

 $ ChestPainType : Factor w/ 4 levels "ASY","ATA","NAP",..: 2 1 1 1
1 1 1 1 2 2 ...

 $ RestingBP     : int  100 120 126 170 130 138 100 135 105 130 ...

 $ Cholesterol   : int  159 237 0 326 206 274 234 297 198 215 ...

 $ FastingBS     : int  0 0 0 0 0 0 0 0 0 0 ...

 $ RestingECG    : Factor w/ 3 levels "LVH","Normal",..: 2 2 3 1 2 2
2 2 2 2 ...

 $ MaxHR         : int  NA 71 NA 140 NA 105 NA NA NA NA ...

 $ ExerciseAngina: Factor w/ 2 levels "N","Y": 1 1 1 2 1 2 1 2 1
1 ...

 $ HeartDisease  : Factor w/ 2 levels "0","1": 1 2 1 2 2 2 NA 2 NA
1 ...

>

>

> #Continuous (MaxHR)

> mean(data2$MaxHR, na.rm = T)

[1] 136.1303

> data2$MaxHR <- ifelse(is.na(data2$MaxHR), 136.1303, data2$MaxHR)
> table(is.na(data2$MaxHR))

FALSE
  676
>
> #Categorical (HeartDisease)
> data3 <- data2 %>% filter(!is.na(data2$HeartDisease))
> summary(data3)
      Age        Sex      ChestPainType   RestingBP      Cholesterol
FastingBS       RestingECG      MaxHR       ExerciseAngina
 Min.   :29.0   F:137   ASY:340       Min.   :  0.0   Min.   :  0.0
Min.   :0.0000   LVH   :138   Min.   : 60.0   N:367
 1st Qu.:48.0   M:492   ATA:109       1st Qu.:120.0   1st Qu.:176.0
1st Qu.:0.0000   Normal:370   1st Qu.:120.0   Y:262
 Median :54.0           NAP:148       Median :130.0   Median :227.0
Median :0.0000   ST    :121   Median :136.1
 Mean   :53.9           TA : 32       Mean   :134.1   Mean   :202.3
Mean   :0.2321                Mean   :135.5
 3rd Qu.:60.0                         3rd Qu.:142.0   3rd Qu.:271.0
3rd Qu.:0.0000                3rd Qu.:150.0
 Max.   :77.0                         Max.   :200.0   Max.   :603.0
Max.   :1.0000                Max.   :195.0
 HeartDisease
 0:272
 1:357




> str(data3)
'data.frame':   629 obs. of  10 variables:
 $ Age           : int  48 67 63 59 49 54 62 39 57 63 ...
 $ Sex            : Factor w/ 2 levels "F","M": 2 2 2 2 2 1 2 2 2
2 ...
 $ ChestPainType : Factor w/ 4 levels "ASY","ATA","NAP",..: 2 1 1 1
1 1 1 2 1 1 ...
 $ RestingBP     : int  100 120 126 170 130 138 135 130 95 185 ...
 $ Cholesterol   : int  159 237 0 326 206 274 297 215 0 0 ...
 $ FastingBS     : int  0 0 0 0 0 0 0 0 1 0 ...
 $ RestingECG    : Factor w/ 3 levels "LVH","Normal",..: 2 2 3 1 2 2
2 2 2 2 ...
 $ MaxHR         : num  136 71 136 140 136 ...
 $ ExerciseAngina: Factor w/ 2 levels "N","Y": 1 1 1 2 1 2 2 1 1
2 ...
 $ HeartDisease  : Factor w/ 2 levels "0","1": 1 2 1 2 2 2 2 1 2
2 ...
> colSums(is.na(data3))
           Age            Sex  ChestPainType      RestingBP
Cholesterol     FastingBS    RestingECG        MaxHR ExerciseAngina
             0              0              0              0
0             0             0             0             0
  HeartDisease
             0
>
> #Final dataset after simple imputation
> str(data3)
'data.frame':   629 obs. of  10 variables:
 $ Age           : int  48 67 63 59 49 54 62 39 57 63 ...
 $ Sex            : Factor w/ 2 levels "F","M": 2 2 2 2 2 1 2 2 2
2 ...
 $ ChestPainType : Factor w/ 4 levels "ASY","ATA","NAP",..: 2 1 1 1
1 1 1 2 1 1 ...
 $ RestingBP     : int  100 120 126 170 130 138 135 130 95 185 ...
 $ Cholesterol   : int  159 237 0 326 206 274 297 215 0 0 ...
 $ FastingBS     : int  0 0 0 0 0 0 0 0 1 0 ...
 $ RestingECG    : Factor w/ 3 levels "LVH","Normal",..: 2 2 3 1 2 2
2 2 2 2 ...
 $ MaxHR         : num  136 71 136 140 136 ...
 $ ExerciseAngina: Factor w/ 2 levels "N","Y": 1 1 1 2 1 2 2 1 1
2 ...
 $ HeartDisease  : Factor w/ 2 levels "0","1": 1 2 1 2 2 2 2 1 2
2 ...
>
> data3$HeartDisease <- as.numeric(data3$HeartDisease) -1
> #Split the dataset No iterative
> set.seed(4052)
```

```
>
> train.index <- sample(1:nrow(data3), 0.7*nrow(data3))
>
> train.data <- data3[train.index,]
> test.data <- data3[-train.index,]
>
> #Each data structure
> str(train.data)
'data.frame':   440 obs. of  10 variables:
 $ Age           : int  53 58 50 63 55 65 55 74 57 35 ...
 $ Sex            : Factor w/ 2 levels "F","M": 2 1 2 2 1 2 2 2 2
2 ...
 $ ChestPainType : Factor w/ 4 levels "ASY","ATA","NAP",..: 2 2 2 1
1 1 2 4 2 2 ...
 $ RestingBP     : int  120 180 120 185 180 135 140 145 140 150 ...
 $ Cholesterol   : int  181 393 168 0 327 254 196 216 265 264 ...
 $ FastingBS     : int  0 0 0 0 0 0 0 1 0 0 ...
 $ RestingECG    : Factor w/ 3 levels "LVH","Normal",..: 2 2 2 2 3 1
2 2 3 2 ...
 $ MaxHR         : num  132 110 160 98 117 127 150 116 145 168 ...
 $ ExerciseAngina: Factor w/ 2 levels "N","Y": 1 2 1 2 2 1 1 2 2
1 ...
 $ HeartDisease  : num  0 1 0 1 1 1 0 1 1 0 ...
>
> str(test.data)
'data.frame':   189 obs. of  10 variables:
 $ Age           : int  67 39 39 51 39 49 68 59 51 48 ...
 $ Sex            : Factor w/ 2 levels "F","M": 2 2 1 1 2 1 2 2 2
2 ...
 $ ChestPainType : Factor w/ 4 levels "ASY","ATA","NAP",..: 1 2 3 3
3 2 1 1 1 2 ...
 $ RestingBP     : int  120 130 138 120 160 124 135 130 130 140 ...
 $ Cholesterol   : int  237 215 220 295 147 201 0 126 179 238 ...
 $ FastingBS     : int  0 0 0 0 1 0 0 0 0 0 ...
 $ RestingECG    : Factor w/ 3 levels "LVH","Normal",..: 2 2 2 1 2 2
3 2 2 2 ...
 $ MaxHR         : num  71 136 152 136 160 ...
 $ ExerciseAngina: Factor w/ 2 levels "N","Y": 1 1 1 1 1 1 2 1 1
1 ...
 $ HeartDisease  : num  1 0 0 0 0 0 1 1 0 0 ...
> corrgram(data3)
>
corrgram(data3[,1:9],order=F,upper.panel=panel.pie,text.panel=panel
.txt,main='Correlation Plot')
> corrgram(data3[,1:9], upper.panel=panel.conf)
> str(train.data)
'data.frame':   440 obs. of  10 variables:
 $ Age           : int  53 58 50 63 55 65 55 74 57 35 ...
 $ Sex            : Factor w/ 2 levels "F","M": 2 1 2 2 1 2 2 2 2
2 ...
 $ ChestPainType : Factor w/ 4 levels "ASY","ATA","NAP",..: 2 2 2 1
1 1 2 4 2 2 ...
 $ RestingBP     : int  120 180 120 185 180 135 140 145 140 150 ...
 $ Cholesterol   : int  181 393 168 0 327 254 196 216 265 264 ...
 $ FastingBS     : int  0 0 0 0 0 0 0 1 0 0 ...
 $ RestingECG    : Factor w/ 3 levels "LVH","Normal",..: 2 2 2 2 3 1
2 2 3 2 ...
 $ MaxHR         : num  132 110 160 98 117 127 150 116 145 168 ...
 $ ExerciseAngina: Factor w/ 2 levels "N","Y": 1 2 1 2 2 1 1 2 2
1 ...
 $ HeartDisease  : num  0 1 0 1 1 1 0 1 1 0 ...
> str(data3)
'data.frame':   629 obs. of  10 variables:
 $ Age           : int  48 67 63 59 49 54 62 39 57 63 ...
 $ Sex            : Factor w/ 2 levels "F","M": 2 2 2 2 2 1 2 2 2
2 ...
 $ ChestPainType : Factor w/ 4 levels "ASY","ATA","NAP",..: 2 1 1 1
1 1 1 2 1 1 ...
 $ RestingBP     : int  100 120 126 170 130 138 135 130 95 185 ...
 $ Cholesterol   : int  159 237 0 326 206 274 297 215 0 0 ...
 $ FastingBS     : int  0 0 0 0 0 0 0 0 1 0 ...
 $ RestingECG    : Factor w/ 3 levels "LVH","Normal",..: 2 2 3 1 2 2
2 2 2 2 ...
 $ MaxHR         : num  136 71 136 140 136 ...
 $ ExerciseAngina: Factor w/ 2 levels "N","Y": 1 1 1 2 1 2 2 1 1
2 ...
```

```
 $ HeartDisease  : num  0 1 0 1 1 1 1 0 1 1 ...
> train.data$HeartDisease <-as.factor(train.data$HeartDisease)
> ad_rf <- randomForest(HeartDisease~., mtry= 3,train.data)
> ad_rf

Call:
 randomForest(formula = HeartDisease ~ ., data = train.data, mtry =
3)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 3

        OOB estimate of  error rate: 18.64%
Confusion matrix:
    0   1 class.error
0 140  44   0.2391304
1  38 218   0.1484375
>
> plot(ad_rf)
> varImpPlot(ad_rf)
> #Train AUC curve
> y_obs <-train.data[,10]
> yhat_rf <- predict(ad_rf, newdata=train.data, type="prob")[,'1']
> pred_rf <-prediction(yhat_rf, y_obs)
> performance(pred_rf,"auc")@y.values[[1]]
[1] 1
> #Test AUC curve
> y_obs2 <-test.data[,10]
> yhat_rf2 <- predict(ad_rf, newdata=test.data, type="prob")[,'1']
> pred_rf2 <-prediction(yhat_rf2, y_obs2)
> performance(pred_rf2,"auc")@y.values[[1]]
[1] 0.8690932
>
> pred_rf3 <- performance(pred_rf2, "tpr", "fpr")
>
```

```
> plot(pred_rf3, colorize=TRUE, main="RF ROC Test mtry=3" )


rain Error rate

> train_pred1<-predict(ad_rf,train.data,type="response")

> table(train.data$HeartDisease,train_pred1)

   train_pred1

      0   1

  0 184   0

  1   0 256

>                                                           ER1<-1-
sum(diag(table(train.data$HeartDisease,train_pred1)))/sum(table(tra
in.data$HeartDisease,train_pred1))

> ER1

[1] 0

> #Test Error rate

> test_pred4<-predict(ad_rf,test.data,type="response")

> table(test.data$HeartDisease,test_pred4)

   test_pred4

     0  1

  0 63 25

  1 12 89

>                                                           ER4<-1-
sum(diag(table(test.data$HeartDisease,test_pred4)))/sum(table(test.
data$HeartDisease,test_pred4))

> ER4

[1] 0.1957672
```

#Appendix 6 (Iterative regression Imputation Random Forest)

```
> #Read the data

>                              data                              <-
read.table(file="C:/Users/isaac/Desktop/heart10.txt",header   =   T,
fileEncoding="UTF-8-BOM")

> summary(data)

      Age         Sex     ChestPainType   RestingBP      Cholesterol
FastingBS       RestingECG      MaxHR       ExerciseAngina

 Min.   :28.00   F:169   ASY :364      Min.   :  0.0   Min.   :  0.0
Min.   :0.0000   LVH   :166   Min.   : 60.0   N:482

 1st Qu.:47.00   M:631   ATA :121      1st Qu.:120.0   1st Qu.:172.0
1st Qu.:0.0000   Normal:484   1st Qu.:120.0   Y:318

 Median :54.00           NAP :157      Median :130.0   Median :223.0
Median :0.0000   ST    :150   Median :138.5

 Mean   :53.34           TA  : 34      Mean   :132.5   Mean   :197.9
Mean   :0.2275                Mean   :137.1

 3rd Qu.:60.00           NA's:124      3rd Qu.:140.0   3rd Qu.:267.0
3rd Qu.:0.0000                3rd Qu.:155.0

 Max.   :77.00                         Max.   :200.0   Max.   :603.0
Max.   :1.0000                Max.   :202.0


NA's   :128

  HeartDisease

 Min.   :0.0000

 1st Qu.:0.0000

 Median :1.0000

 Mean   :0.5569

 3rd Qu.:1.0000

 Max.   :1.0000

 NA's   :53

> str(data)

'data.frame':   800 obs. of  10 variables:

 $ Age          : int  64 48 67 63 59 49 54 58 62 32 ...

 $ Sex            : Factor w/ 2 levels "F","M": 1 2 2 2 2 2 1 2 2
1 ...

 $ ChestPainType : Factor w/ 4 levels "ASY","ATA","NAP",..: NA 2 1 1
1 1 1 1 1 2 ...

 $ RestingBP     : int  95 100 120 126 170 130 138 100 135 105 ...
 $ Cholesterol   : int  0 159 237 0 326 206 274 234 297 198 ...

 $ FastingBS     : int  1 0 0 0 0 0 0 0 0 0 ...

 $ RestingECG    : Factor w/ 3 levels "LVH","Normal",..: 2 2 2 3 1 2
2 2 2 2 ...

 $ MaxHR         : int  145 NA 71 NA 140 NA 105 NA NA NA ...

 $ ExerciseAngina: Factor w/ 2 levels "N","Y": 1 1 1 1 2 1 2 1 2
1 ...

 $ HeartDisease  : int  1 0 1 0 1 1 1 NA 1 NA ...

>

> #Change the Heartdisease as factor

> data$HeartDisease <- as.factor(data$HeartDisease)

>

> #How many missing varaibles?

> table(is.na(data))


FALSE  TRUE

 7695   305

> colSums(is.na(data))

           Age            Sex  ChestPainType      RestingBP
Cholesterol     FastingBS     RestingECG         MaxHR ExerciseAngina

             0              0            124              0
0              0              0            128              0

  HeartDisease

            53

>

>

> #Another dataset with Iterative regression impuration

> iter_reg_data=data

> summary(iter_reg_data)

      Age          Sex      ChestPainType   RestingBP      Cholesterol
FastingBS        RestingECG       MaxHR        ExerciseAngina

 Min.   :28.00   F:169   ASY :364     Min.   :  0.0   Min.   :  0.0
Min.   :0.0000   LVH   :166   Min.   : 60.0   N:482

 1st Qu.:47.00   M:631   ATA :121     1st Qu.:120.0   1st Qu.:172.0
1st Qu.:0.0000   Normal:484   1st Qu.:120.0   Y:318

 Median :54.00           NAP :157     Median :130.0   Median :223.0
Median :0.0000   ST    :150   Median :138.5

 Mean   :53.34           TA  : 34     Mean   :132.5   Mean   :197.9
Mean   :0.2275                Mean   :137.1

 3rd Qu.:60.00           NA's:124      3rd Qu.:140.0   3rd Qu.:267.0
3rd Qu.:0.0000                3rd Qu.:155.0

 Max.   :77.00                         Max.   :200.0   Max.   :603.0
Max.   :1.0000                Max.   :202.0


NA's   :128

 HeartDisease

 0   :331

 1   :416

 NA's: 53
```

```
> colSums(is.na(iter_reg_data))

           Age                 Sex   ChestPainType         RestingBP
Cholesterol      FastingBS     RestingECG          MaxHR ExerciseAngina

             0                  0               124                0
0               0               0             128              0

  HeartDisease

            53

iter_reg_data$MaxHR[is.na(iter_reg_data$MaxHR)]=mean(iter_reg_data$
MaxHR,na.rm=TRUE)

>

> summary(iter_reg_data$ChestPainType)

 ASY  ATA  NAP   TA NA's

 364  121  157   34  124

>
iter_reg_data$ChestPainType[is.na(iter_reg_data$ChestPainType)]="AS
Y"

>

> summary(iter_reg_data$ChestPainType)

ASY ATA NAP  TA

488 121 157  34

> summary(iter_reg_data$HeartDisease)

   0    1 NA's

 331  416   53
```

```
> iter_reg_data$HeartDisease[is.na(iter_reg_data$HeartDisease)]="1"
> summary(iter_reg_data$HeartDisease)
  0   1
331 469
>
> n_iter=20
> for(i in 1:n_iter)
+ {
+   #impute Price give rest
+   m_MaxHR=lm(MaxHR~.,iter_reg_data,subset=!is.na(data$MaxHR))
+   pred_MaxHR=predict(m_MaxHR,iter_reg_data[is.na(data$MaxHR),])
+   iter_reg_data$MaxHR[is.na(data$MaxHR)]=pred_MaxHR
+
+   #impute ChestPainType given rest
+   library(nnet)
+   m_ChestPainType=multinom(ChestPainType~.,iter_reg_data,
+                      subset=!is.na(data$ChestPainType),trace=FALSE)
+
pred_ChestPainType=predict(m_ChestPainType,iter_reg_data[is.na(data
$ChestPainType),])
+
iter_reg_data$ChestPainType[is.na(data$ChestPainType)]=pred_ChestPa
inType
+   #impute HeartDisease given rest
+
m_HeartDisease=glm(HeartDisease~.,iter_reg_data,subset=!is.na(data$
HeartDisease),family="binomial")
+
pred_HeartDisease=predict(m_HeartDisease,iter_reg_data[is.na(data$H
eartDisease),],type="response")
+
iter_reg_data$HeartDisease[is.na(data$HeartDisease)]=ifelse(pred_He
artDisease >0.5, "1","0")
+ }
>
> mean(iter_reg_data$MaxHR)
[1] 137.0224
> str(iter_reg_data$HeartDisease)
```

```
 Factor w/ 2 levels "0","1": 2 1 2 1 2 2 2 2 2 1 ...
> str(data)
'data.frame':   800 obs. of  10 variables:
 $ Age           : int  64 48 67 63 59 49 54 58 62 32 ...
 $ Sex            : Factor w/ 2 levels "F","M": 1 2 2 2 2 2 1 2 2
1 ...
 $ ChestPainType : Factor w/ 4 levels "ASY","ATA","NAP",..: NA 2 1 1
1 1 1 1 1 2 ...
 $ RestingBP     : int  95 100 120 126 170 130 138 100 135 105 ...
 $ Cholesterol   : int  0 159 237 0 326 206 274 234 297 198 ...
 $ FastingBS     : int  1 0 0 0 0 0 0 0 0 0 ...
 $ RestingECG    : Factor w/ 3 levels "LVH","Normal",..: 2 2 2 3 1 2
2 2 2 2 ...
 $ MaxHR         : int  145 NA 71 NA 140 NA 105 NA NA NA ...
 $ ExerciseAngina: Factor w/ 2 levels "N","Y": 1 1 1 1 2 1 2 1 2
1 ...
 $ HeartDisease  : Factor w/ 2 levels "0","1": 2 1 2 1 2 2 2 NA 2
NA ...
>                   iter_reg_data$HeartDisease                   <-
as.numeric(iter_reg_data$HeartDisease) -1
> data$HeartDisease <- as.numeric(data$HeartDisease) -1
>
>  iter_reg_data$HeartDisease <-as.factor(iter_reg_data$HeartDisease)
> data$HeartDisease <-as.factor(data$HeartDisease)
>
> data3 <- iter_reg_data
>
> data3$HeartDisease <- as.numeric(data3$HeartDisease) -1
> #Split the dataset No iterative
> set.seed(4052)
>
> train.index <- sample(1:nrow(data3), 0.7*nrow(data3))
>
> train.data <- data3[train.index,]
> test.data <- data3[-train.index,]
>
> #Each data structure
> str(train.data)
'data.frame':   560 obs. of  10 variables:
 $ Age           : int  46 41 52 32 61 52 45 54 55 56 ...
 $ Sex            : Factor w/ 2 levels "F","M": 2 1 2 1 2 2 1 2 1
2 ...
 $ ChestPainType : Factor w/ 4 levels "ASY","ATA","NAP",..: 1 2 1 2
1 1 2 2 2 3 ...
 $ RestingBP     : int  140 126 160 105 105 125 180 160 110 130 ...
 $ Cholesterol   : int  311 306 246 198 0 212 295 305 344 221 ...
 $ FastingBS     : int  0 0 0 0 1 0 0 0 0 0 ...
 $ RestingECG    : Factor w/ 3 levels "LVH","Normal",..: 2 2 3 2 2 2
2 2 3 1 ...
 $ MaxHR         : num  120 163 124 166 110 ...
 $ ExerciseAngina: Factor w/ 2 levels "N","Y": 2 1 2 1 2 1 1 1 1
1 ...
 $ HeartDisease  : num  1 0 1 0 1 1 0 0 0 0 ...
>
> str(test.data)
'data.frame':   240 obs. of  10 variables:
 $ Age           : int  64 48 55 39 39 58 59 62 50 58 ...
 $ Sex            : Factor w/ 2 levels "F","M": 1 2 2 2 1 2 2 2 2
2 ...
 $ ChestPainType : Factor w/ 4 levels "ASY","ATA","NAP",..: 1 2 1 2
3 1 3 1 1 1 ...
 $ RestingBP     : int  95 100 140 130 138 120 130 138 140 116 ...
 $ Cholesterol   : int  0 159 0 215 220 0 318 204 231 0 ...
 $ FastingBS     : int  1 0 0 0 0 0 0 0 0 0 ...
 $ RestingECG    : Factor w/ 3 levels "LVH","Normal",..: 2 2 2 2 2 1
2 3 3 2 ...
 $ MaxHR         : num  145 149 83 158 152 ...
 $ ExerciseAngina: Factor w/ 2 levels "N","Y": 1 1 1 1 1 2 2 2 2
1 ...
 $ HeartDisease  : num  1 0 1 0 0 1 0 1 1 1 ...
> corrgram(data3)
>
corrgram(data3[,1:9],order=F,upper.panel=panel.pie,text.panel=panel
.txt,main='Correlation Plot')
> corrgram(data3[,1:9], upper.panel=panel.conf)
```

```
>
> str(train.data)
'data.frame':   560 obs. of  10 variables:
 $ Age           : int  46 41 52 32 61 52 45 54 55 56 ...
 $ Sex            : Factor w/ 2 levels "F","M": 2 1 2 1 2 2 1 2 1
2 ...
 $ ChestPainType : Factor w/ 4 levels "ASY","ATA","NAP",..: 1 2 1 2
1 1 2 2 2 3 ...
 $ RestingBP     : int  140 126 160 105 105 125 180 160 110 130 ...
 $ Cholesterol   : int  311 306 246 198 0 212 295 305 344 221 ...
 $ FastingBS     : int  0 0 0 0 1 0 0 0 0 0 ...
 $ RestingECG    : Factor w/ 3 levels "LVH","Normal",..: 2 2 3 2 2 2
2 2 3 1 ...
 $ MaxHR         : num  120 163 124 166 110 ...
 $ ExerciseAngina: Factor w/ 2 levels "N","Y": 2 1 2 1 2 1 1 1 1
1 ...
 $ HeartDisease  : num  1 0 1 0 1 1 0 0 0 0 ...
> str(data3)
'data.frame':   800 obs. of  10 variables:
 $ Age           : int  64 48 67 63 59 49 54 58 62 32 ...
 $ Sex            : Factor w/ 2 levels "F","M": 1 2 2 2 2 2 1 2 2
1 ...
 $ ChestPainType : Factor w/ 4 levels "ASY","ATA","NAP",..: 1 2 1 1
1 1 1 1 1 2 ...
 $ RestingBP     : int  95 100 120 126 170 130 138 100 135 105 ...
 $ Cholesterol   : int  0 159 237 0 326 206 274 234 297 198 ...
 $ FastingBS     : int  1 0 0 0 0 0 0 0 0 0 ...
 $ RestingECG    : Factor w/ 3 levels "LVH","Normal",..: 2 2 2 3 1 2
2 2 2 2 ...
 $ MaxHR         : num  145 149 71 120 140 ...
 $ ExerciseAngina: Factor w/ 2 levels "N","Y": 1 1 1 1 2 1 2 1 2
1 ...
 $ HeartDisease  : num  1 0 1 0 1 1 1 1 1 0 ...
> train.data$HeartDisease <-as.factor(train.data$HeartDisease)
> ad_rf <- randomForest(HeartDisease~., mtry= 3,train.data)
> ad_rf


Call:
 randomForest(formula = HeartDisease ~ ., data = train.data, mtry =
3)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 3

        OOB estimate of  error rate: 15.36%
Confusion matrix:
    0   1 class.error
0 196  48   0.1967213
1  38 278   0.1202532
>
> plot(ad_rf)
>
> varImpPlot(ad_rf)
> #Train AUC curve
> y_obs <-train.data[,10]
> yhat_rf <- predict(ad_rf, newdata=train.data, type="prob")[,'1']
> pred_rf <-prediction(yhat_rf, y_obs)
> performance(pred_rf,"auc")@y.values[[1]]
[1] 1
> #Test AUC curve
> y_obs2 <-test.data[,10]
> yhat_rf2 <- predict(ad_rf, newdata=test.data, type="prob")[,'1']
> pred_rf2 <-prediction(yhat_rf2, y_obs2)
> performance(pred_rf2,"auc")@y.values[[1]]
[1] 0.8844866
>
> pred_rf3 <- performance(pred_rf2, "tpr", "fpr")
> plot(pred_rf3, colorize=TRUE, main="RF ROC Test mtry=3" )
> #Train Error rate
> train_pred1<-predict(ad_rf,train.data,type="response")
> table(train.data$HeartDisease,train_pred1)
   train_pred1
      0   1
```

```
  0 244   0
  1   0 316
> ER1<-1-sum(diag(table(train.data$HeartDisease,train_pred1)))/sum(table(train.data$HeartDisease,train_pred1))
> ER1
[1] 0
>
>
> #Test Error rate
> test_pred4<-predict(ad_rf,test.data,type="response")
> table(test.data$HeartDisease,test_pred4)
   test_pred4
      0   1
  0  90  22
  1  22 106
> ER4<-1-sum(diag(table(test.data$HeartDisease,test_pred4)))/sum(table(test.data$HeartDisease,test_pred4))
> ER4
[1] 0.1833333
```