Examining Heart Disease

Table of Contents

Authors: Shivani Mehta

Metin Senturk

Pooja Umathe

This project is published at this website and open sources at GitHub.

Executive Summary

Early diagnosis of the factors that increase lead to the heart disease can help in taking precautions in timely manner. The research revolves around predicting the most influential factors leading to the heart disease using data mining techniques.

In this report, we tried to construct a model to predict the heart disease using two of well-known the classification methdods, logistic regression and support vector machines. We used "Chol", "MaxHR", "Oldpeak" out of all contuinuous type variables to predict the heart disease (AHD). According to our analysis with confusion matrices, we predict **74%** of the patients accurately in terms of their heart disease with logistic regression. Similarly, **73.6%** of patients are a also predicted with the use of support vector machines.

Technical Analysis

There are two different approach used for making the predictions:

1) Logistic Regression:

   - AHD , MaxHR, Oldpeak, RestBP are significant factors that can be used for prediction of a heart disease.
   - Accuracy of the model is 80%
   - The confusion matrix shows 72% accuracy.

2) Support Vector Machines:

   - AHD, Age, RestBP, Chol, MaxHR, Oldpeak give the best result for prediction of the heart disease
   - 'C-classification' type model with kernel 'radial' is used as classifier.
   - The confusion matrix shows 80% accuracy.

Introduction

We are investigating the Heart.csv which contains of various factor that can help in prediction of heart disease. Our objective to find a linear regression and svm model that is best for prediction of a heart disease. Here we will consider only the continuous variables to predict models.

```
# importing dataset
data_raw = read.csv("./Dataset/Heart.csv")
head(data_raw[ , c(2, 5, 6, 9, 11, 15)])
```

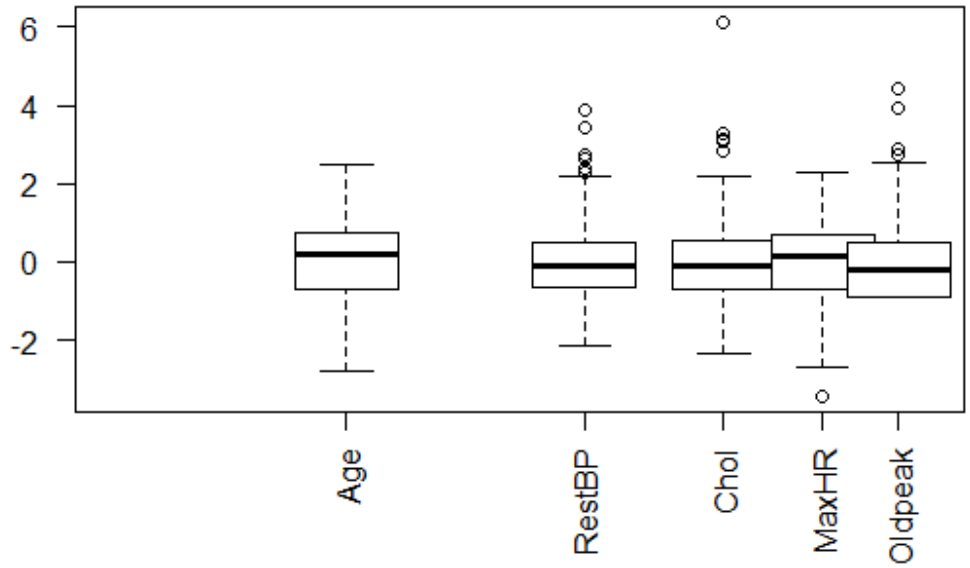
```
##   Age RestBP Chol MaxHR Oldpeak AHD
## 1  63    145  233   150     2.3  No
## 2  67    160  286   108     1.5 Yes
## 3  67    120  229   129     2.6 Yes
## 4  37    130  250   187     3.5  No
## 5  41    130  204   172     1.4  No
## 6  56    120  236   178     0.8  No
```

Data Analysis

To meet the requirements of the prediction the data to be analysed should be analyzed, processed and cleaned. Here we examine the data, then further the outliers are removed and lastly the dataset is divided for testing and training.

```
# assigning 0 and 1 to predictor
data_raw$AHD = as.factor(ifelse(data_raw$AHD == "Yes", 1, 0))
```

```
# examining NA values
data_raw[, names(cont_list)] %>%
  select_if(is.numeric)  %>%
  summarise_if(is.atomic, funs(sum(is.na(.))))

##   Age RestBP Chol MaxHR Oldpeak
## 1   0      0    0     0       0

# normalization
data_raw[, names(cont_list)] <- data.frame(sapply(cont_list, scale))
```



Outlier Detection with Boxplot

```
# outlier removals on detected variables
data_raw$Chol = outlier_handler(data_raw$Chol)
data_raw$RestBP = outlier_handler(data_raw$RestBP)
data_raw$Oldpeak = outlier_handler(data_raw$Oldpeak)

# train and test datasets
data_raw = data_raw[, c(2, 5, 6, 9, 11, 15)]
data_train = data_raw[ 1:250, ]
data_test = data_raw[251:303, ]
```

After removal of outliers and operations applied to our dataset in above, the following is the structure of the dataset that we will be using for analysis.

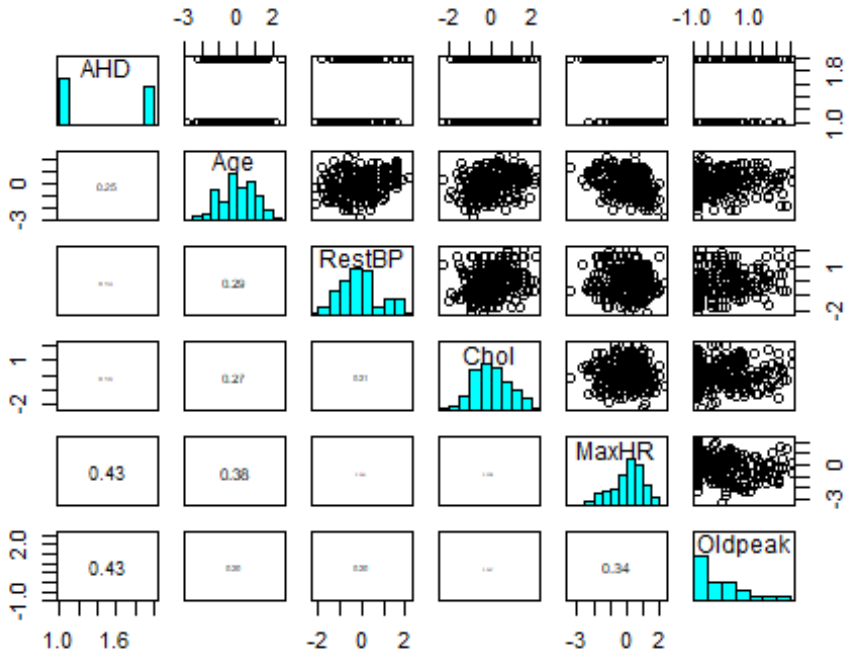
```
# examining variables
str(data_raw)

## 'data.frame':    303 obs. of  6 variables:
##  $ Age    : num  0.947 1.39 1.39 -1.929 -1.487 ...
##  $ RestBP : num  0.756 1.609 -0.664 -0.096 -0.096 ...
##  $ Chol   : num  -0.2645 0.7592 -0.3417 0.0639 -0.8246 ...
##  $ MaxHR  : num  0.0172 -1.8189 -0.9009 1.6347 0.9789 ...
```

```
##  $ Oldpeak: num  1.086 0.397 1.344 2.119 0.31 ...
##  $ AHD    : Factor w/ 2 levels "0","1": 1 2 2 1 1 1 2 1 2 2 ...
```

Variables' Analysis

Co-Relation between variables is an important aspect to be considered while predicting a model. Therefore, below in the graph, we represented the correlation values and graphs of variables with each other.

```
# correlation between predictor or variables
pairs(AHD ~ Age + RestBP + Chol + MaxHR + Oldpeak,
      data = data_train,
      diag.panel = panel.hist,
      lower.panel = panel.cor)
```




```
# assuming all having the disease
t = table(rep(1, 250), data_train$AHD)

# chances
data.frame(all_diseased  = t[1,2] / sum(t),
           none_diseased = t[1,1] / sum(t))

##   all_diseased none_diseased
## 1        0.448         0.552
```

Models

In order to understand patient's illness, we will construct two different two classification model. We will use logistic regression model and support vector machines (SVM) in order to classify the disease.

Logistic Regression Approach

Logistic regression is a method to understand more about a predictor which has only two outcomes from a set of independent variables. Here in this dataset, we evaluated many variables together and presented the output in below. From many of our model, we will be presenting the following models.

1. LGM with all variables.
2. GLM with "MaxHR", "Oldpeak" and "Chol" variables.

A Glance of All Variables

We started with using all the continuous type variables in the dataset.

```
# f_1 <- AHD ~ MaxHR + Oldpeak + RestBP
lgm_model = glm(AHD ~ Age + RestBP + Chol + MaxHR + Oldpeak, data = data_train, family = binom
ial)
```

In order to understand about variables, we evaluated their variances.

```
# variance importance of variables
varImp(lgm_model)
```

```
##           Overall
## Age     0.4593191
## RestBP  0.4957732
## Chol    1.8548966
## MaxHR   4.5201632
## Oldpeak 4.6422079
```

Since we have multiple independent variables, we run chi square test to understand the relationship between predictor and each of the independent variables. From our analysis in below, we find MaxHR, Oldpeak and Age are of importance.

```
# analysis of variance of all variables
invisible(anova(lgm_model, test = "Chisq"))
```

However, further analysis on variables indicate age is not statistically significant as others. The following two models become a candidate to predict the presense of an heart disease. From these two, we implemented likelihood ratio test to observe the importance of "RestBP" and "Chol".

```
# best
lgm_model_1 = glm(AHD ~ MaxHR + Oldpeak,          data = data_train, family = binomial)
lgm_model_2 = glm(AHD ~ MaxHR + Oldpeak + RestBP, data = data_train, family = binomial)
lgm_model_3 = glm(AHD ~ MaxHR + Oldpeak + Chol,   data = data_train, family = binomial)

# Anova of two best models selected
anova(lgm_model_1, lgm_model_2, test = "LRT")
```

```
## Analysis of Deviance Table
##
## Model 1: AHD ~ MaxHR + Oldpeak
## Model 2: AHD ~ MaxHR + Oldpeak + RestBP
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1       247     266.52
## 2       246     265.43  1    1.095   0.2954
```

```
# Anova of two best models selected
anova(lgm_model_1, lgm_model_3, test = "LRT")
```

```
## Analysis of Deviance Table
##
## Model 1: AHD ~ MaxHR + Oldpeak
## Model 2: AHD ~ MaxHR + Oldpeak + Chol
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1       247     266.52
## 2       246     261.54  1   4.9844  0.02558 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

According to the likelihood ratio test results of anova, "Chol" values are statistically more significant than "RestBP" values, therefore we picked "MaxHR", "Oldpeak" and "Chol" to construct our model in classification of heart diseases.

More Analysis on the Best Model

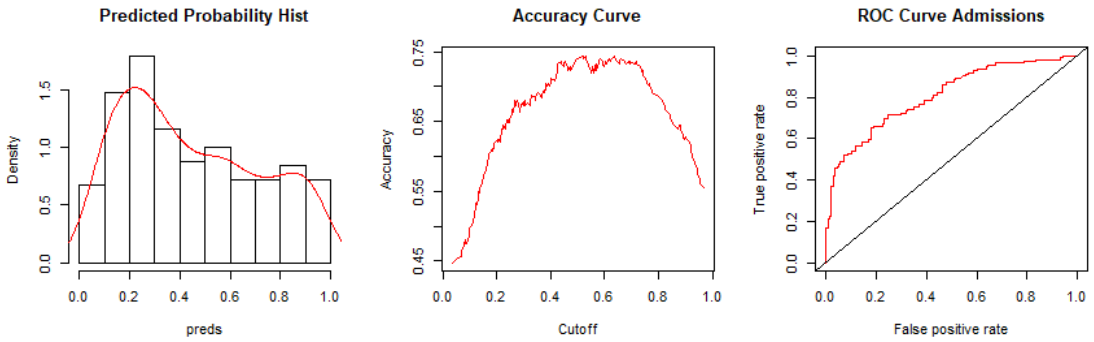
To examine our model we applied some statistical tests for further analysis on the model. Below, the odds of having a heart disease (AHD) is 2.40 times higher in one unit increase of "Oldpeak". Likewise, 1 unit of "Chol" increase in a patient can increase the odds of having a heart disease 1.49 times.

```
# odds of variables in the model
ort <- cbind(exp(confint(lgm_model)),
             Coefficients = coef(lgm_model),
             'Odds Ratio' = exp(coef(lgm_model)))

## Waiting for profiling to be done...

round(ort, 2)

##             2.5 % 97.5 % Coefficients Odds Ratio
## (Intercept)  0.61   1.11        -0.19       0.82
## MaxHR        0.30   0.59        -0.86       0.42
## Oldpeak      1.70   3.47         0.87       2.40
## Chol         1.05   2.13         0.40       1.49
```

Below, the comparison of null deviance and residual deviance test the significance of variables in the model. The result indicate that each variable is statistically significant to present the heart disease.

```
# chi square goodness-of-fit test
anova(lgm_model, update(lgm_model, ~ 0), test="Chisq")

## Analysis of Deviance Table
##
## Model 1: AHD ~ MaxHR + Oldpeak + Chol
## Model 2: AHD ~ 1 - 1
##   Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
## 1       246     261.54
## 2       250     346.57 -4  -85.037 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Reciever Operating Characteristic (ROC) curve is used to find a good decision boundary point. The predictions from the dataset are evaluated in terms of true positive (tpr) and false positive (fpr) rates are plotted in below. A curve in the upper left side of the ROC curve represents our model is presenting fruitful outcome.



Cutoff Value Plots

According to the graph above, we determined the following cutoff value with respect to it's accuracy for our model.

```
# accuracy and cutoff value
cutoff_roc(probs_lgm, data_train$AHD)

##                  [,1]
## sensitivity 0.7142857
## specificity 0.7536232
## cutoff      0.4381594
```

We evaluate our model performance with a confusion matrix. According to confusion matrix, the misclassification error in our model is promising with a value of 0.26. Likewise, the model's accuracy for predicting correct results is 0.74.

Additionally, we provide some informative raitos of guessing all true and false in order to compare accuracy and mse.

```
# decision boundary value from cutoff analysis
db <- unname(cutoff_acc(probs_lgm, data_train$AHD)[2,1])

# confusion matrix for train data
probs = predict(lgm_model, data_train, type = "response")
confmatrix(probs_lgm, data_train$AHD, db)

## $matrix
##           actual
## predicted   0   1
##         0 128  55
##         1  10  57
##
## $information
##      tpr    fpr  mse accuracy if_all_false if_all_true
## 1 0.9649 0.4297 0.26     0.74        0.552       0.448
```

Lastly, in order to eliminate the chance of picking the best subset in the dataset for our selected model, we did a k-fold cross validation test with k = 10 for our analysis. For that purpose, we created train and test datasets in below. Hence, training the model with 80% of the dataset, which is roughly the same amount of observations with comparison to our initial analysis.

```
# creating training and test dataset
train_index <- createDataPartition(data_raw$AHD, p=0.8, list=F)
training <- data_raw[ train_index, ]
testing <- data_raw[ -train_index, ]
```

We constructed a k-fold cross validation with 10 repeats and 10 folds. According to the results of k-fold cross validation test, we found the model we picked presents similiar to our initial analysis.

The result of k-fold cv model is also indicating the same independent variables as from our presented model's variables. Therefore, we proved that the presented model is behaving as our first analysis.

```
# k-fold control
ctrl <- trainControl(method = "repeatedcv",
                     number = 10, repeats = 10,
                     savePredictions = TRUE)

# k-fold training
mod_fit <- train(AHD ~ MaxHR + Oldpeak + Chol, data=training,
                 method="glm", family="binomial",
                 trControl = ctrl,
                 tuneLength = 10)
```

The confision matrix for k-fold cross validation is used to examine accuracy and its significance value. The accuracy observed with k-fold cross validation is 0.72, which is similiar with our initial model's results.

```
# confusion matrix results for training data
pred <- predict(mod_fit, newdata=training)
confusionMatrix(data = pred, training$AHD)

## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 106  42
##          1  26  70
##
##                Accuracy : 0.7213
##                  95% CI : (0.6605, 0.7766)
##     No Information Rate : 0.541
##     P-Value [Acc > NIR] : 5.731e-09
##
##                   Kappa : 0.4327
##  Mcnemar's Test P-Value : 0.06891
##
##             Sensitivity : 0.8030
##             Specificity : 0.6250
##          Pos Pred Value : 0.7162
##          Neg Pred Value : 0.7292
##              Prevalence : 0.5410
##          Detection Rate : 0.4344
##    Detection Prevalence : 0.6066
##       Balanced Accuracy : 0.7140
##
##        'Positive' Class : 0
##
```

Model Validation

To validate our model, we used test dataset in the presented model that we subsetted to present model's success. It is observed that our model acts slightly less accurate with test data than in training.

```
# confusion matrix for test data (Presented Model)
probs = predict(lgm_model, data_test, type = "response")
confmatrix(probs, data_test$AHD, db)
```

```
## $matrix
##          actual
## predicted  0  1
##         0 22 12
##         1  4 15
##
## $information
##   tpr    fpr    mse accuracy if_all_false if_all_true
## 1 0.8 0.5455 0.3019   0.6981       0.4906      0.5094
```

Lastly, the testing dataset created from k-fold cross validation subset is analyzed with our model. Here, k-fold cross validation results with the test dataset is presents better accuracy than our presented model in the test dataset.

```
# k-fold validation for test data (k-fold analysis Model)
pred <- predict(mod_fit, newdata=testing)
confusionMatrix(data = pred, testing$AHD)

## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 25  6
##          1  7 21
##
##                Accuracy : 0.7797
##                  95% CI : (0.6527, 0.8771)
##     No Information Rate : 0.5424
##     P-Value [Acc > NIR] : 0.0001366
##
##                   Kappa : 0.5574
##  Mcnemar's Test P-Value : 1.0000000
##
##             Sensitivity : 0.7812
##             Specificity : 0.7778
##          Pos Pred Value : 0.8065
##          Neg Pred Value : 0.7500
##              Prevalence : 0.5424
##          Detection Rate : 0.4237
##    Detection Prevalence : 0.5254
##       Balanced Accuracy : 0.7795
##
##        'Positive' Class : 0
##
```

Support Vector Machine Approach

Model Analysis

Another approach we used to to predict the presense of the heart disease is support vector machines (SVM). SVM divides the dataset into classes with the use of hyperplanes. Below, we used svm to create a model for our dataset. Since we implemented normalization, we turned down the scale option for svm. Additionally, we did k-fold cross validation with 10 folds. We found the best SVM with the tune function.

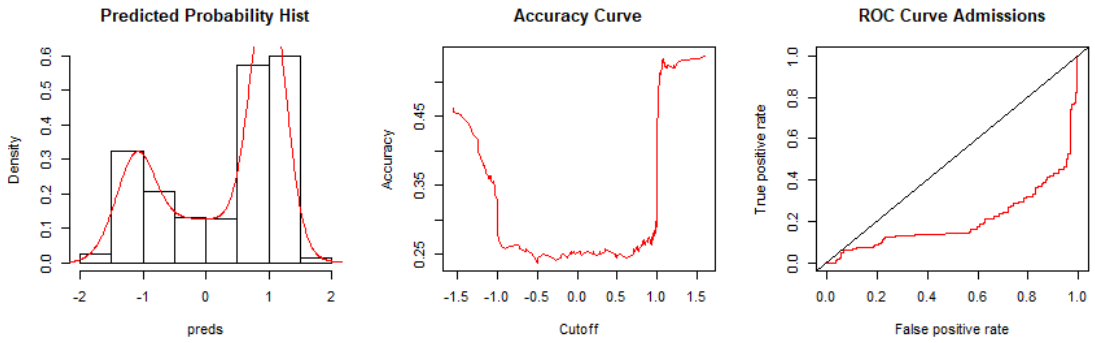
```
# finding best model with svm tune
svm_tune <- tune(svm, AHD ~ Chol + MaxHR + Oldpeak, data = data_raw,
                 type = "C-classification",
                 kernel = "radial",
                 decision.values = T,
                 scale = F,
                 tunecontrol = tune.control(cross = 10, nrepeat = 2),
```

```
                tuneLength = 8,
                ranges = list(gamma = 2^(-1:2), cost = 2^(-1:10)))


# first six performances
# head(svm_tune$performances)
```

According to our analysis, we found the best cost and gamma values as 0.5 and 0.5, respectively. Therefore, we are selecting the best model out of the tune function.

```
# getting the best model
svm_model <- svm_tune$best.model

# a brief summary of the best model
summary(svm_model)

##
## Call:
## best.tune(method = svm, train.x = AHD ~ Chol + MaxHR + Oldpeak,
##     data = data_raw, ranges = list(gamma = 2^(-1:2), cost = 2^(-1:10)),
##     tunecontrol = tune.control(cross = 10, nrepeat = 2), type = "C-classification",
##     kernel = "radial", decision.values = T, scale = F, tuneLength = 8)
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  radial
##        cost:  2
##       gamma:  0.5
##
## Number of Support Vectors:  197
##
##  ( 102 95 )
##
##
## Number of Classes:  2
##
## Levels:
##  0 1
```

ROC curve for the SVM Model is used to examine an optimal cutoff point. At left, we present probability distribution graph, the accuracy curve at the center and on the right is the ROC curve.



SVM ROC for Train Data

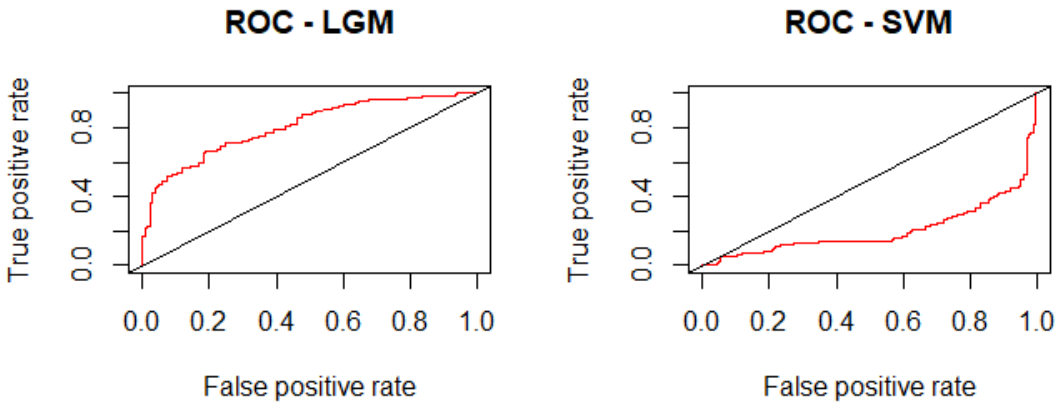
Model Validation

We will validate our model's accuracy with the test dataset. Below, confusion matrix presents the spesifity and sensitivity of our model on test data.

```
# confusion matrix for svm on test data
preds <- predict(svm_model, data_raw)
confusionMatrix(preds, data_raw$AHD)

## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 143  56
##          1  21  83
##
##                Accuracy : 0.7459
##                  95% CI : (0.6929, 0.7939)
##     No Information Rate : 0.5413
##     P-Value [Acc > NIR] : 1.653e-13
##
##                   Kappa : 0.4783
##  Mcnemar's Test P-Value : 0.0001068
##
##             Sensitivity : 0.8720
##             Specificity : 0.5971
##          Pos Pred Value : 0.7186
##          Neg Pred Value : 0.7981
##              Prevalence : 0.5413
##          Detection Rate : 0.4719
##    Detection Prevalence : 0.6568
##       Balanced Accuracy : 0.7345
##
##        'Positive' Class : 0
##
```

Model Comparisions

Lastly, we compared our results of two models the ROC curve. It is observed that logistic regression presents better results than svm as a classifier for our dataset.



Comparison between ROC Curves

Conclusion

In this report, we examined the presense of a heart disease from a set of variables including chollesterol, Age and others provided in the dataset. We found logistic regression to be a better model for prediction.

According to our model, the odds of one patient having the disease is highly effected by "Chol", "MaxHR" and "Oldpeak". The odds of having AHD disease is very high if a patient's oldpeak is increases by one unit.