Classification_methods

Biswajit Chowdhury

28/05/2020

Load the required library

```r
library(tidyverse) # Data manipulation and visualization
```

```
## -- Attaching packages ------------------------------------------------------------- tidyverse 1.3.0

## v ggplot2 3.3.0     v purrr   0.3.3
## v tibble  3.0.0     v dplyr   0.8.5
## v tidyr   1.0.2     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0

## -- Conflicts ----------------------------------------------------------------- tidyverse_conflicts()
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
library (caret)   # Machine learning workflow
```

```
## Loading required package: lattice

##
## Attaching package: 'caret'

## The following object is masked from 'package:purrr':
##
##     lift
```

```r
library(klaR)   # Naive Bayes classifier
```

```
## Loading required package: MASS

##
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':
##
##     select
```

```r
library(MASS) # Discriminant analysis
theme_set(theme_bw())
```

Load the data and visualize the summary of the dataset

```r
heart_data<-read.csv("heart.csv")
# to add the colonm names in the dataset
names(heart_data) <- c("age", "sex", "chest_pain", "resting_bp","cholestrol",
                       "fasting_sugar", "resting_ECG", "max_heart_rate", "exercise_agina",
                       "oldpeak", "slope", "number_major_vessels", "thal", "target")

# visualize the sumary of the dataset
head(heart_data,3)
```

```
##   age sex chest_pain resting_bp cholestrol fasting_sugar resting_ECG
## 1  63   1          3        145        233             1           0
## 2  37   1          2        130        250             0           1
## 3  41   0          1        130        204             0           0
##   max_heart_rate exercise_agina oldpeak slope number_major_vessels thal target
## 1            150              0     2.3     0                    0    1      1
## 2            187              0     3.5     0                    0    2      1
## 3            172              0     1.4     2                    0    2      1
```

```r
str(heart_data)
```

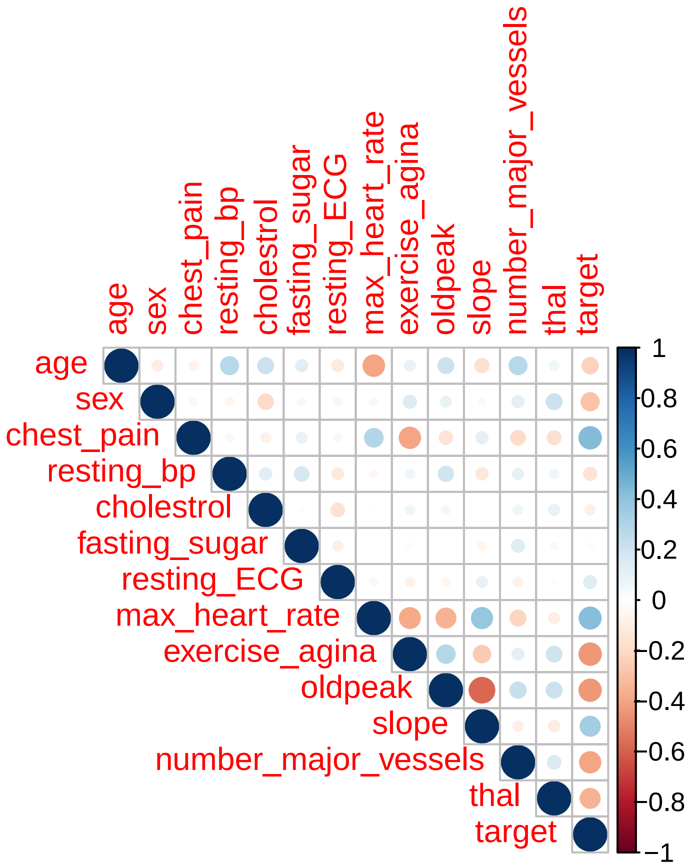
```
## 'data.frame':    303 obs. of  14 variables:
##  $ age                 : int  63 37 41 56 57 57 56 44 52 57 ...
##  $ sex                 : int  1 1 0 1 0 1 0 1 1 1 ...
##  $ chest_pain          : int  3 2 1 1 0 0 1 1 2 2 ...
##  $ resting_bp          : int  145 130 130 120 120 140 140 120 172 150 ...
##  $ cholestrol          : int  233 250 204 236 354 192 294 263 199 168 ...
##  $ fasting_sugar       : int  1 0 0 0 0 0 0 0 1 0 ...
##  $ resting_ECG         : int  0 1 0 1 1 1 0 1 1 1 ...
##  $ max_heart_rate      : int  150 187 172 178 163 148 153 173 162 174 ...
##  $ exercise_agina      : int  0 0 0 0 1 0 0 0 0 0 ...
##  $ oldpeak             : num  2.3 3.5 1.4 0.8 0.6 0.4 1.3 0 0.5 1.6 ...
##  $ slope               : int  0 0 2 2 2 1 1 2 2 2 ...
##  $ number_major_vessels: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ thal                : int  1 2 2 2 2 1 2 3 3 2 ...
##  $ target              : int  1 1 1 1 1 1 1 1 1 1 ...
```

Relationship between the variables and distribution of quantitative variables

```r
library(corrplot)
```

```
## corrplot 0.84 loaded
```

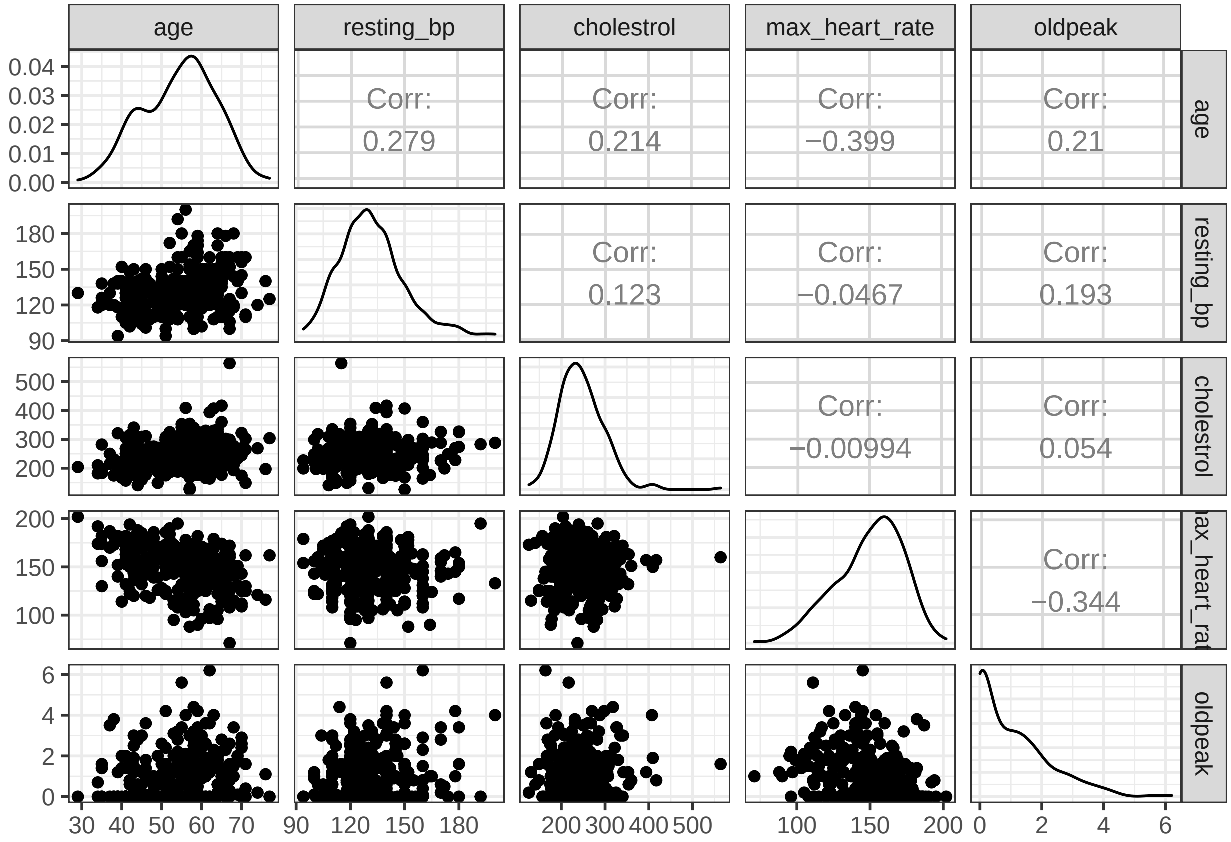
```
my_data <- heart_data

corrplot(cor(my_data), type="upper")
```



```
# Create a scatter plot matrix and density curve for quantitatve variables
library(GGally)
```

```
##
## Attaching package: 'GGally'

## The following object is masked from 'package:dplyr':
##
##     nasa
```

```
library(ggplot2)

ggpairs(my_data[, c(1, 4, 5, 8, 10)])
```



Positive correlations are displayed in blue and negative correlations in red color. The scale for correlationa color intensity are spread from -1 to 1 at the right. Color intensity and the size of the circle are proportional to the correlation coefficients.

Changed a few predictor variables from integer to factors for regression analysis

```
my_data[, 2]<-factor(my_data[, 2])
my_data[, 3]<-factor(my_data[, 3])
my_data[, 6]<-factor(my_data[, 6])
my_data[, 7]<-factor(my_data[, 7])
my_data[, 9]<-factor(my_data[, 9])
my_data[, 11]<-factor(my_data[, 11])
my_data[, 12]<-factor(my_data[, 12])
my_data[, 13]<-factor(my_data[, 13])
my_data[, 14]<-factor(my_data[, 14])

# Check the new format of predictor variables
str(my_data)
```

```
## 'data.frame':    303 obs. of  14 variables:
##  $ age                   : int  63 37 41 56 57 57 56 44 52 57 ...
##  $ sex                   : Factor w/ 2 levels "0","1": 2 2 1 2 1 2 1 2 2 2 ...
##  $ chest_pain          : Factor w/ 4 levels "0","1","2","3": 4 3 2 2 1 1 2 2 3 3 ...
##  $ resting_bp          : int  145 130 130 120 120 140 140 120 172 150 ...
##  $ cholestrol          : int  233 250 204 236 354 192 294 263 199 168 ...
##  $ fasting_sugar       : Factor w/ 2 levels "0","1": 2 1 1 1 1 1 1 1 2 1 ...
##  $ resting_ECG         : Factor w/ 3 levels "0","1","2": 1 2 1 2 2 2 1 2 2 2 ...
##  $ max_heart_rate      : int  150 187 172 178 163 148 153 173 162 174 ...
##  $ exercise_agina      : Factor w/ 2 levels "0","1": 1 1 1 1 2 1 1 1 1 1 ...
##  $ oldpeak             : num  2.3 3.5 1.4 0.8 0.6 0.4 1.3 0 0.5 1.6 ...
##  $ slope               : Factor w/ 3 levels "0","1","2": 1 1 3 3 3 2 2 3 3 3 ...
##  $ number_major_vessels: Factor w/ 5 levels "0","1","2","3",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ thal                : Factor w/ 4 levels "0","1","2","3": 2 3 3 3 3 2 3 4 4 3 ...
##  $ target              : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 2 2 2 ...
```

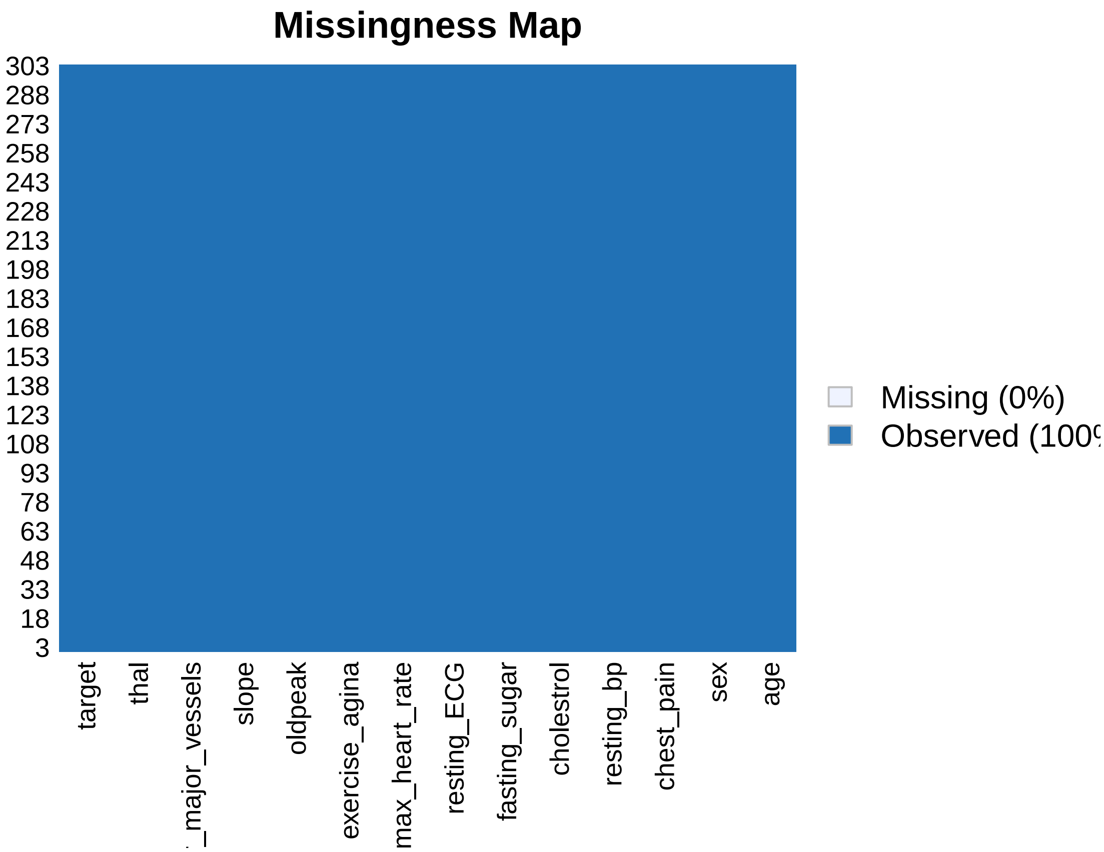
Visualize the missing value

```
library(Amelia)
```

```
## Loading required package: Rcpp
```

```
## ##
## ## Amelia II: Multiple Imputation
## ## (Version 1.7.6, built: 2019-11-24)
## ## Copyright (C) 2005-2020 James Honaker, Gary King and Matthew Blackwell
## ## Refer to http://gking.harvard.edu/amelia/ for more information
## ##
```

```
missmap(my_data)
```



There is no missing values

Split the data set for the machine learning algorithms

```
set.seed(123)

training_samples<- my_data$target%>%
        createDataPartition(p=0.7, list = FALSE)
train_data<-my_data[training_samples, ]
test_data<- my_data[-training_samples, ]
```

1. Logistic regression model (LGM)

```
set.seed(123)
# Fit the model
model <- glm(target ~., data = train_data, family = binomial)

# Summarize the final output of the model
summary(model)
```

```
##
```

```
## Call:
## glm(formula = target ~ ., family = binomial, data = train_data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.7258  -0.3690   0.1163   0.4438   3.2122
##
## Coefficients:
##                        Estimate Std. Error z value Pr(>|z|)
## (Intercept)            1.137833   3.752860   0.303 0.761744
## age                    0.017083   0.030570   0.559 0.576284
## sex1                  -1.801517   0.668380  -2.695 0.007031 **
## chest_pain1            1.112029   0.692340   1.606 0.108232
## chest_pain2            1.538037   0.618804   2.485 0.012937 *
## chest_pain3            2.341734   0.871905   2.686 0.007236 **
## resting_bp            -0.025159   0.015277  -1.647 0.099577 .
## cholestrol            -0.011915   0.005909  -2.016 0.043759 *
## fasting_sugar1         0.903331   0.706105   1.279 0.200786
## resting_ECG1           0.442854   0.474269   0.934 0.350427
## resting_ECG2          -0.266805   2.480812  -0.108 0.914355
## max_heart_rate         0.018944   0.013615   1.391 0.164123
## exercise_agina1       -0.704483   0.557405  -1.264 0.206280
## oldpeak               -0.134014   0.286224  -0.468 0.639631
## slope1                 0.037735   0.999392   0.038 0.969880
## slope2                 1.960043   1.111106   1.764 0.077724 .
## number_major_vessels1 -2.241621   0.589912  -3.800 0.000145 ***
## number_major_vessels2 -4.421049   1.200705  -3.682 0.000231 ***
## number_major_vessels3 -2.031186   1.205220  -1.685 0.091926 .
## number_major_vessels4  1.003685   1.695395   0.592 0.553846
## thal1                  2.558072   2.361130   1.083 0.278627
## thal2                  2.343203   2.219684   1.056 0.291129
## thal3                  1.274155   2.241273   0.568 0.569698
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 293.58  on 212  degrees of freedom
## Residual deviance: 133.14  on 190  degrees of freedom
## AIC: 179.14
##
## Number of Fisher Scoring iterations: 6
```

```r
# Calculate the model predictions and accuracy
probabilities <- model %>% predict(test_data, type = "response")
predicted.classes <- ifelse(probabilities > 0.5, "0", "1")

# Model accuracy
accuracy_LR<- mean(predicted.classes==test_data$target)

accuracy_LR
```

```
## [1] 0.1444444
```

As we can see from the above output that all the variables are not significantly associated with the outcome variables. So we should taken out the non siginificant variables and run the algorithm to make the best model

```
# Fit the model
model <- glm(target ~ sex + chest_pain + cholestrol+ number_major_vessels, data = train_data, family = 
# Summarize the final output of the model
summary(model)
```

```
## 
## Call:
## glm(formula = target ~ sex + chest_pain + cholestrol + number_major_vessels, 
##     family = binomial, data = train_data)
## 
## Deviance Residuals: 
##     Min       1Q   Median       3Q      Max  
## -2.0744  -0.6950   0.2729   0.6204   2.3561  
## 
## Coefficients:
##                        Estimate Std. Error z value Pr(>|z|)    
## (Intercept)            3.020415   1.118708   2.700 0.006936 ** 
## sex1                  -1.486784   0.431319  -3.447 0.000567 ***
## chest_pain1            2.209272   0.532277   4.151 3.32e-05 ***
## chest_pain2            2.100986   0.456662   4.601 4.21e-06 ***
## chest_pain3            1.990189   0.694859   2.864 0.004181 ** 
## cholestrol            -0.008685   0.004102  -2.117 0.034250 *  
## number_major_vessels1 -1.882232   0.451282  -4.171 3.03e-05 ***
## number_major_vessels2 -2.942099   0.807890  -3.642 0.000271 ***
## number_major_vessels3 -2.221298   0.886313  -2.506 0.012203 *  
## number_major_vessels4  0.244848   1.314035   0.186 0.852184    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for binomial family taken to be 1)
## 
##     Null deviance: 293.58  on 212  degrees of freedom
## Residual deviance: 186.01  on 203  degrees of freedom
## AIC: 206.01
## 
## Number of Fisher Scoring iterations: 5
```

```
# Calculate the model predictions and accuracy
probabilities <- model %>% predict(test_data, type = "response")
predicted.classes <- ifelse(probabilities > 0.5, "0", "1")
# Model accuracy
accuracy_LR<- mean(predicted.classes==test_data$target)

accuracy_LR
```

```
## [1] 0.2
```

The accuracy of the model has been improved from the global methods (from 14% to 20%). However this is not the best predition model. We should try other methods

2. Stepwise logistic regression (SLR)

This method autometically removes nonsignificant preditable variables for building the best regression model.

```r
library(MASS)
# Fit the model
model_SLR <- glm(target ~., data = train_data, family = binomial) %>%
  stepAIC(direction = "both", trace = FALSE)

# Summarize the final output of the model
summary(model_SLR)
```

```
##
## Call:
## glm(formula = target ~ sex + chest_pain + resting_bp + cholestrol +
##     fasting_sugar + max_heart_rate + exercise_agina + slope +
##     number_major_vessels, family = binomial, data = train_data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.5817  -0.4276   0.1171   0.4518   3.2550
##
## Coefficients:
##                        Estimate Std. Error z value Pr(>|z|)
## (Intercept)            4.690432   2.805914   1.672 0.094599 .
## sex1                  -2.200867   0.581424  -3.785 0.000154 ***
## chest_pain1            1.399916   0.656867   2.131 0.033072 *
## chest_pain2            1.635078   0.577837   2.830 0.004660 **
## chest_pain3            2.393525   0.853335   2.805 0.005033 **
## resting_bp            -0.029687   0.012817  -2.316 0.020541 *
## cholestrol            -0.012704   0.005625  -2.259 0.023910 *
## fasting_sugar1         0.973421   0.665731   1.462 0.143691
## max_heart_rate         0.019901   0.012234   1.627 0.103787
## exercise_agina1       -0.861969   0.547637  -1.574 0.115493
## slope1                 0.337303   0.868855   0.388 0.697857
## slope2                 2.439848   0.901745   2.706 0.006816 **
## number_major_vessels1 -2.205808   0.552670  -3.991 6.57e-05 ***
## number_major_vessels2 -4.402394   1.060105  -4.153 3.28e-05 ***
## number_major_vessels3 -2.476234   1.139150  -2.174 0.029723 *
## number_major_vessels4  0.490423   1.539184   0.319 0.750011
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 293.58  on 212  degrees of freedom
## Residual deviance: 139.79  on 197  degrees of freedom
## AIC: 171.79
##
## Number of Fisher Scoring iterations: 6
```

```r
# Calculate the model predictions and accuracy
probabilities <- model_SLR %>% predict(test_data, type = "response")
predicted.classes <- ifelse(probabilities > 0.5, "0", "1")
# Model accuracy
accuracy_SLR<- mean(predicted.classes==test_data$target)

accuracy_SLR
```

```
## [1] 0.1666667
```

3. Support Vector Machine (SVM) Model

Support vector machine methods can handle both linear and non-linear class boundaries. Prior building the model, variables are normalized to It standardized the variables to make can be used for both two-class and multi-class classification problems.

```r
# Variables are normalized to make their scale comparable. This is automatically done before building t

set.seed(123)
model_SVM<- train(
  target~., data = train_data, method="svmRadial",
  trControl=trainControl("cv", number=10),

  preProcess = c("center", "scale"),

  tuneLength=10
  )
# Summarize the final output of the model
summary (model_SVM)
```

```
## Length  Class   Mode
##      1   ksvm     S4
```

```r
# Calculate the model predictions and accuracy

predicted_classes<- model_SVM%>% predict(test_data)

observed_classes<- test_data$target

# compute the accuracy rate
accuracy_SVM<- mean(predicted_classes==test_data$target)
accuracy_SVM
```

```
## [1] 0.8666667
```

```r
# model performance test - Confusion matrix
table(observed_classes, predicted_classes)
```

```
##                 predicted_classes
## observed_classes  0  1
##                0 38  3
##                1  9 40
```

Quadratic discriminant analysis (QDA)

```
model_QDA <- qda(target~., data = train_data)
model
```

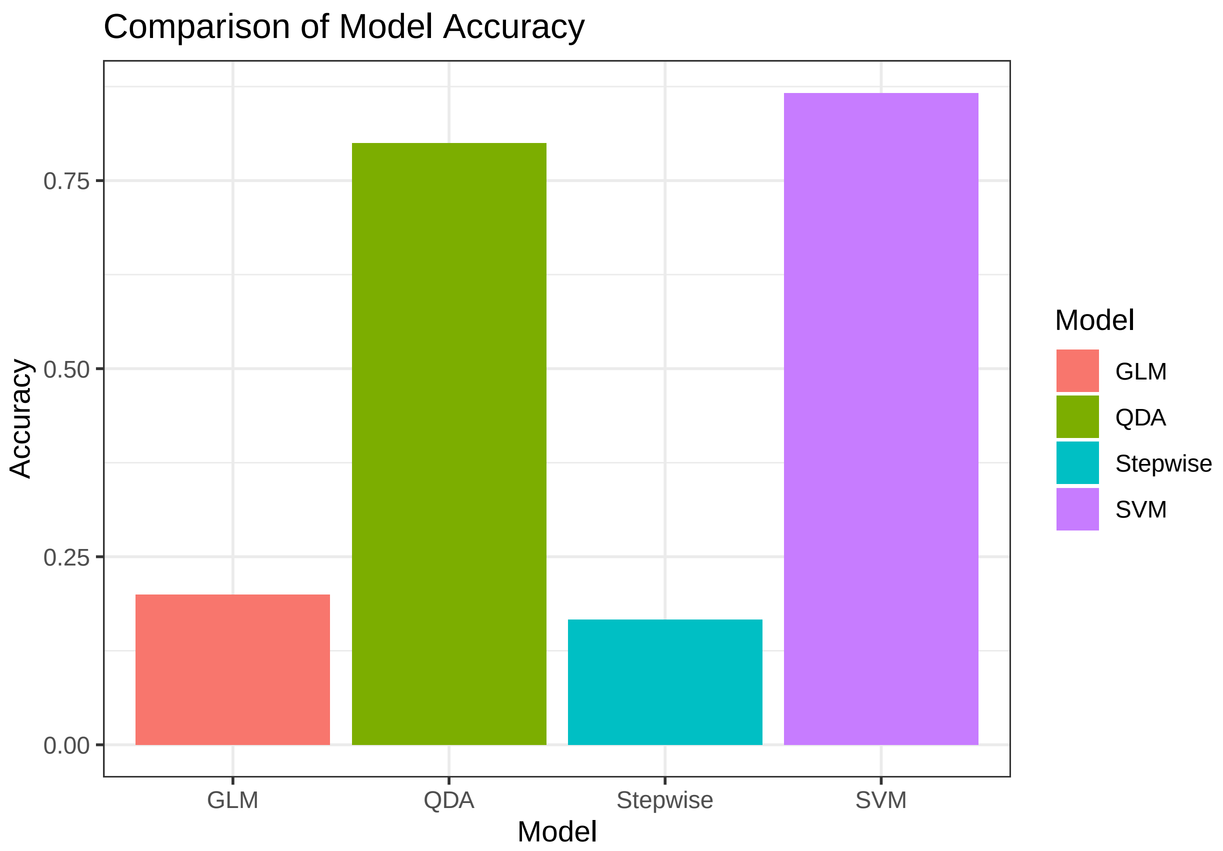
```
##
## Call:  glm(formula = target ~ sex + chest_pain + cholestrol + number_major_vessels,
##     family = binomial, data = train_data)
##
## Coefficients:
##           (Intercept)                   sex1            chest_pain1
##              3.020415              -1.486784               2.209272
##           chest_pain2            chest_pain3             cholestrol
##              2.100986               1.990189              -0.008685
## number_major_vessels1  number_major_vessels2  number_major_vessels3
##             -1.882232              -2.942099              -2.221298
## number_major_vessels4
##              0.244848
##
## Degrees of Freedom: 212 Total (i.e. Null);  203 Residual
## Null Deviance:       293.6
## Residual Deviance: 186   AIC: 206
```

```
# Compute the predictions and model accuracy
predicted_classes <- model_QDA %>% predict(test_data)


# Model accuracy
accuracy_QDA<- mean(predicted_classes$class == test_data$target)
accuracy_QDA
```

```
## [1] 0.8
```

Comparison of Model Accuracy

```
accuracy <- data.frame(Model=c("GLM", "Stepwise", "SVM", "QDA"),
  Accuracy=c(accuracy_LR, accuracy_SLR, accuracy_SVM, accuracy_QDA))

ggplot(accuracy,aes(x=Model,y=Accuracy, fill=Model)) + geom_bar(stat='identity') +
  ggtitle('Comparison of Model Accuracy')
```



Based on the above prediction models, SVM has the higisest prediction accuarcy rate (86%) compare to rest. Therefore SVM model is highly appropriate for this dataset.