USE CASE STUDY REPORT

Group No.: Group 09

Student Names: Chenghan Yue and Zijian Zhao

Executive Summary:

This report focuses on analyzing the likelihood on getting physical condition data gathered from hospital. The dataset contains 14 variables including age, type of chest pain, maximum heart rate, resting blood pressure etc. The dataset was split into training (80%) and validating part (20%). The goal is to predict once the patient's physical information is recorded, whether heart attack will occur. We expect to run several models on the dataset and find out the model with the best accuracy. The techniques includes classification tree, logistic regression, KNN model and SVM model. The result indicates that logistic regression model has the best accuracy on this dataset. We recommend the one who fits the following three condition to check for heart attack risk: 1. Experiencing chest pain. 2. Over age of 60. 3. High maximum heart rate.

I. Background and Introduction

Heart disease is the leading cause of death in our world. As CDC reported, In the United States, 1 in every 4 deaths is caused by heart disease, that is about 610,000 people who die from the heart condition each year. To prevent heart disease occur, people always take heart health examination in the hospital so that there are enormous data to analyze heart disease throughout the country.

This study focuses on a heart examination dataset and build a Machine Learning model that predicts whether the heart disease will occur after the patient taken a heart examination. The prediction analysis based on heart examination report.

The possible solution might be derived from the as much as algorithm that will classify the report request into two classes: 1. Occurred 2. Didn't occur. We intended to use as many models as possible to analyze the dataset.

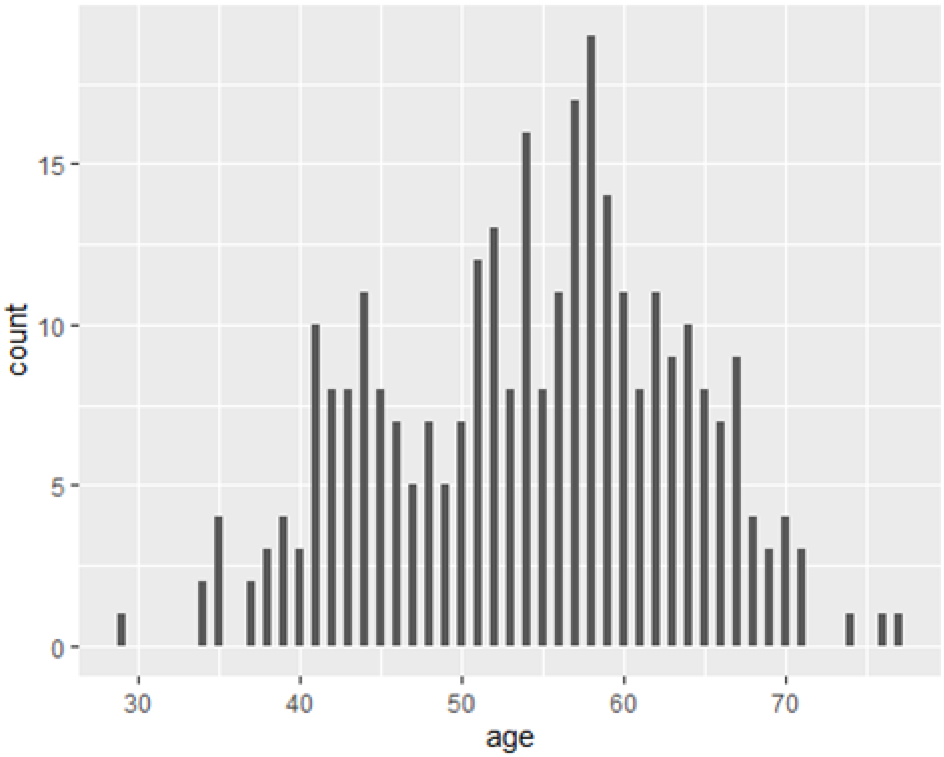
II. Data Exploration and Visualization

Variables of the dataset includes:

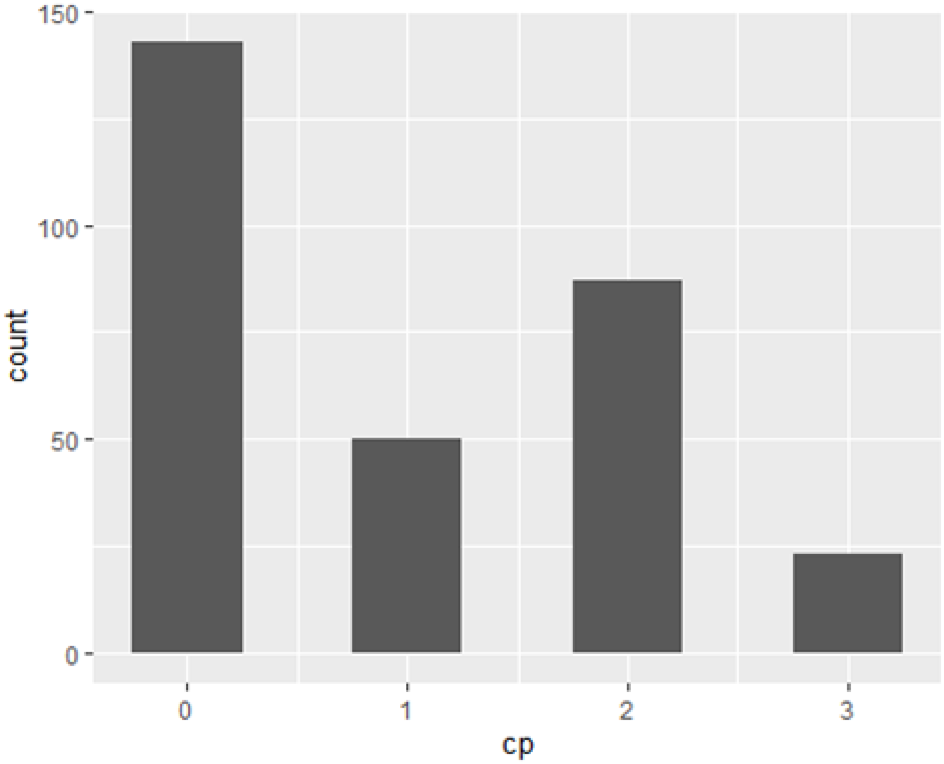
1. age
2. sex: 1 denotes male and 0 denotes female
3. chest pain type (4 values): the values between 0 to 3. If the value is higher, the more probability of heart attack to occurr
4. resting blood pressure: the blood pressure without exercise
5. serum cholesterol in mg/dl: the blockage for blood supply in the blood vessels
6. fasting blood sugar > 120 mg/dl: blood sugar taken after a long gap between a mean and the test
7. resting electrocardiographic results (values 0,1,2): ECG values taken while person is on rest which means no exercise and normal functioning of heart is happening
8. maximum heart rate achieved
9. exercise induced angina: chest pain while exercising or doing any physical activity
10. oldpeak = ST depression induced by exercise relative to rest
11. the slope of the peak exercise ST segment
12. number of major vessels (0-3) colored by fluoroscopy
13. thal (3 = normal; 6 = fixed defect; 7 = reversable defect): the types of thalassemia
14. target: 1 denotes Heart attack occurred and 0 where it didn't occur

Data Visualization:

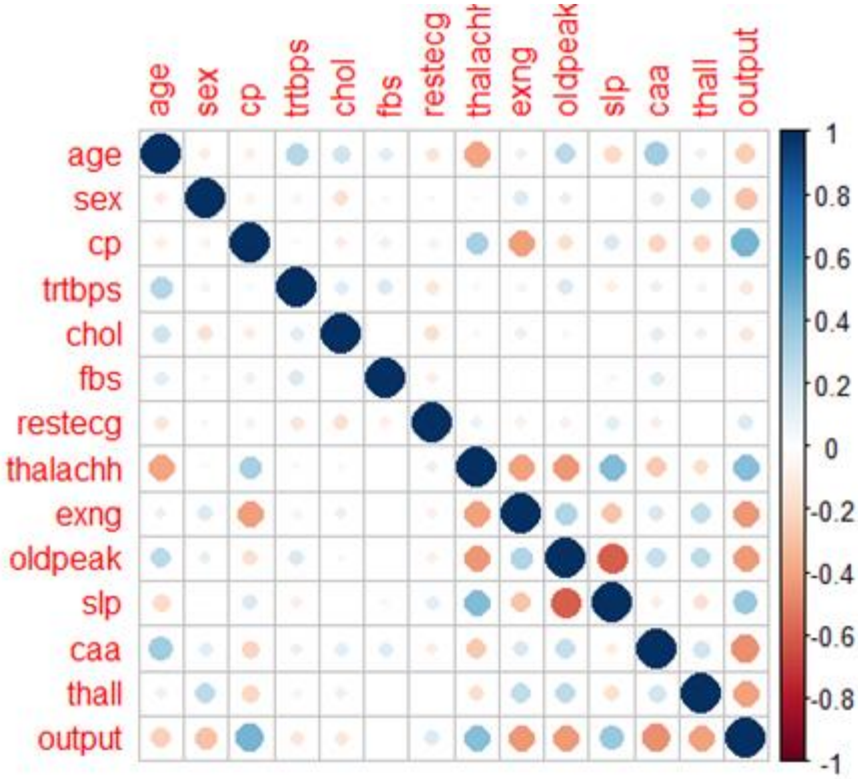
1. Histogram of Heart Attack vs. Age



2. Histogram of Heart Attack vs. Chest Pain



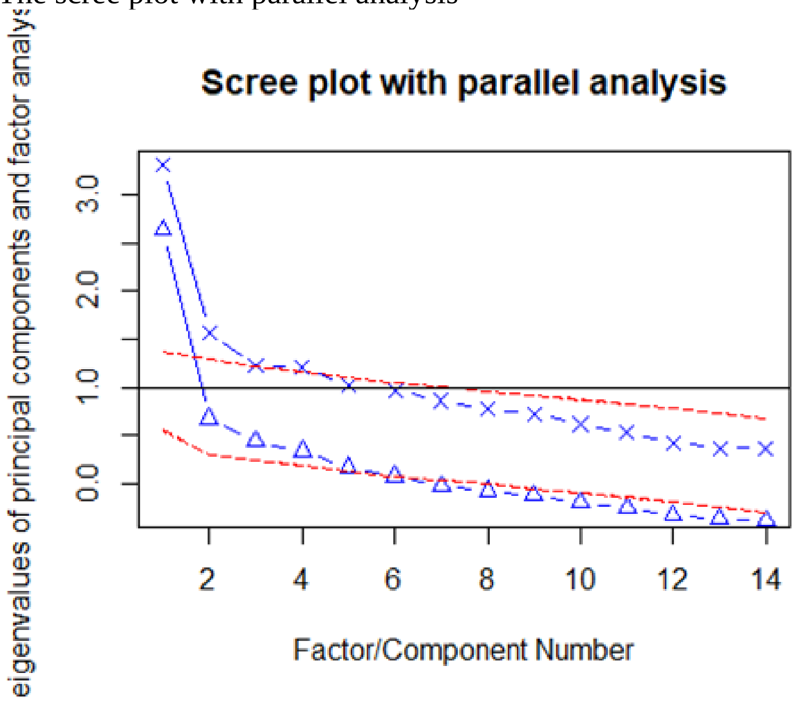
3. Correlations



III. Data Preparation and Preprocessing

Parallel analysis

The scree plot with parallel analysis



Parallel analysis suggests that the number of factors is 4 and the number of components is 2.

PCA analysis can help us find importance of component.

Table of Importance of component

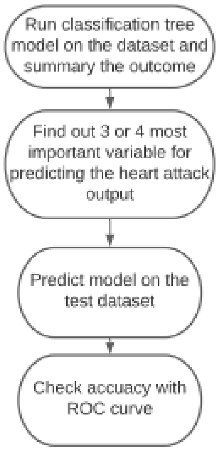
	PC1	PC2	PC3	PC4	PC5	PC6	PC7
Standard deviation	1.8170	1.2539	1.1100	1.09847	1.0110	0.9850	0.92910
Proportion of Variance	0.2358	0.1123	0.0880	0.08619	0.0730	0.0693	0.06166
Cumulative Proportion	0.2358	0.3481	0.4361	0.52231	0.5953	0.6646	0.72627
	PC8	PC9	PC10	PC11	PC12	PC13	PC14
Standard deviation	0.88096	0.85393	0.78913	0.73103	0.65577	0.60982	0.60658
Proportion of Variance	0.05544	0.05209	0.04448	0.03817	0.03072	0.02656	0.02628
Cumulative Proportion	0.78170	0.83379	0.87827	0.91644	0.94716	0.97372	1.00000

IV. Data Mining Techniques and Implementation

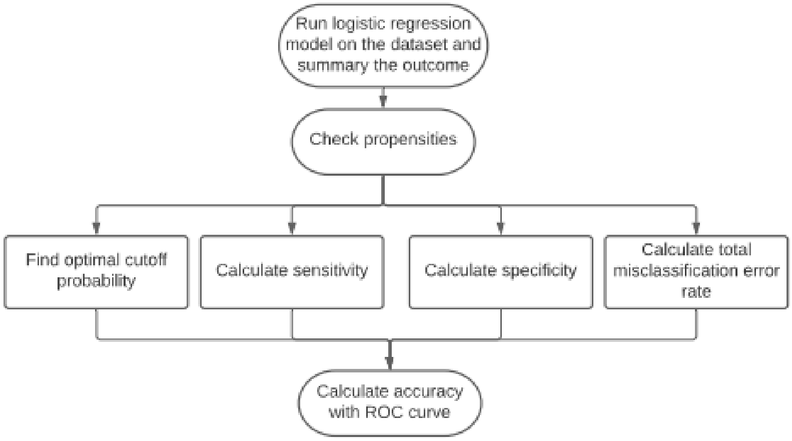
The dataset is split into 2 parts. 80% of the data is used for training and 20% is used for validation. In different model, the selection of variables might be different.

Flow charts of all possible model to use in this analysis:

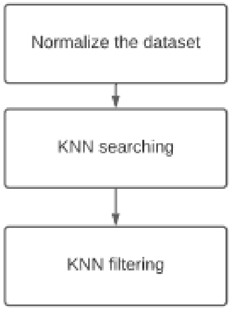
Flow Chart of Classification Tree Model


Run classification tree model on the dataset and summary the outcome
Find out 3 or 4 most important variable for predicting the heart attack output
Predict model on the test dataset
Check accuacy with ROC curve

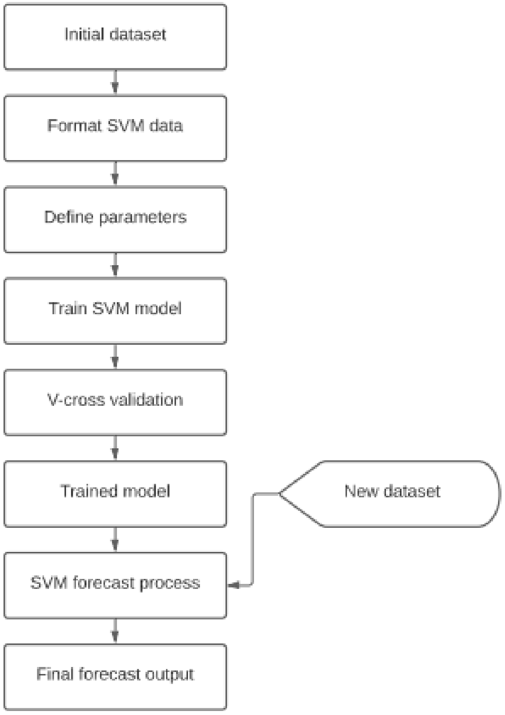
Flow Chart of Logistic Regression Model


Run logistic regression model on the dataset and summary the outcome
Check propensities
Find optimal cutoff probability
Calculate sensitivity
Calculate specificity
Calculate total misclassification error rate
Calculate accuracy with ROC curve

Flow Chart of KNN Model

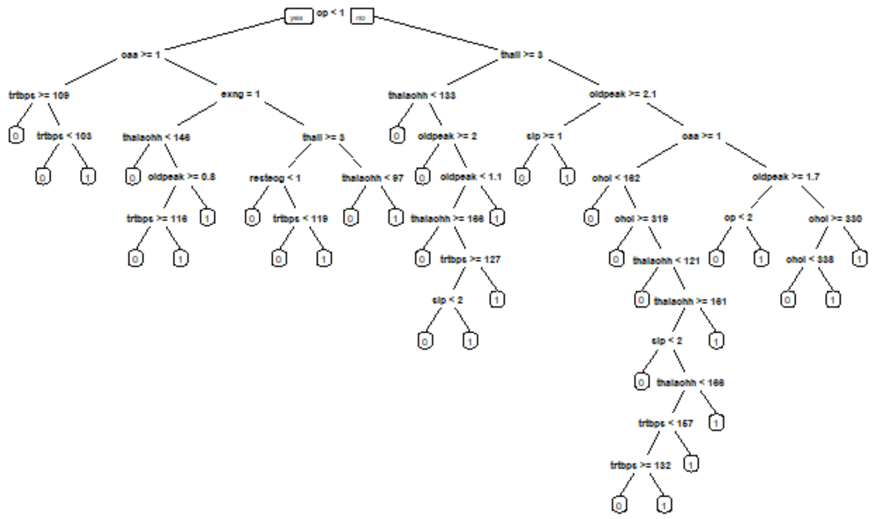


Flow Chart of SVM Model



1. Classification tree model

   2.1 Plot of classification tree

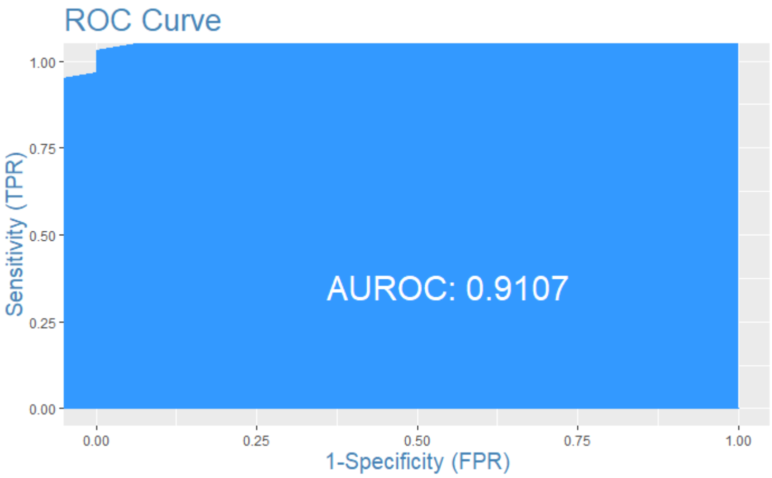


2.2 The important specifications

cp	32.90058209
thalachh	30.96112021
oldpeak	25.55918489
thall	24.60120894
exng	23.23695835
slp	17.09906921
caa	15.94304334
trtbps	13.81436673
chol	13.41792713
restecg	2.34761905
fbs	0.04370581
cp	32.90058209

From the table we can see that the 4 most important specifications are cp, age, thalachh and old peak.

1.3 ROC curve



2. Logistic regression model

2.1 Summary of the outcome of logistic regression model

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

Deviance Residuals: 
    Min       1Q   Median       3Q      Max  
-2.8335  -0.4074   0.2428   0.6196   2.2072  

Coefficients:
            Estimate Std. Error z value Pr(>|z|)    
(Intercept)  0.09537    0.19337   0.493 0.621864    
cp           0.77049    0.20455   3.767 0.000165 ***
trtbps      -0.32147    0.19534  -1.646 0.099830 .  
chol         0.01841    0.20540   0.090 0.928584    
fbs         -0.16910    0.20436  -0.827 0.407966    
restecg      0.25348    0.19380   1.308 0.190889    
thalachh     0.37265    0.21672   1.720 0.085523 .  
exng        -0.47179    0.20417  -2.311 0.020843 *  
oldpeak     -0.80167    0.26789  -2.992 0.002767 ** 
slp          0.38434    0.22137   1.736 0.082525 .  
caa         -0.76441    0.20482  -3.732 0.000190 ***
thall       -0.69948    0.19349  -3.615 0.000300 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 333.10  on 241  degrees of freedom
Residual deviance: 181.75  on 230  degrees of freedom
AIC: 205.75

Number of Fisher Scoring iterations: 6

(Intercept)          cp      trtbps        chol         fbs     restecg    thalachh        exng 
oldpeak         slp         caa       thall 
  1.1000707   2.1608301   0.7250855   1.0185795   0.8444240   1.2885054 
1.4515801   0.6238816   0.4485798   1.4686471   0.4656083   0.4968455 
```

2.2 Propensities

Find out the optimal cutoff probability used for maximization accuracy

```
##    0  1
## 0 22  2
## 1  7 29
```

Calculate sensitivity

```
## [1] 0.9354839
```

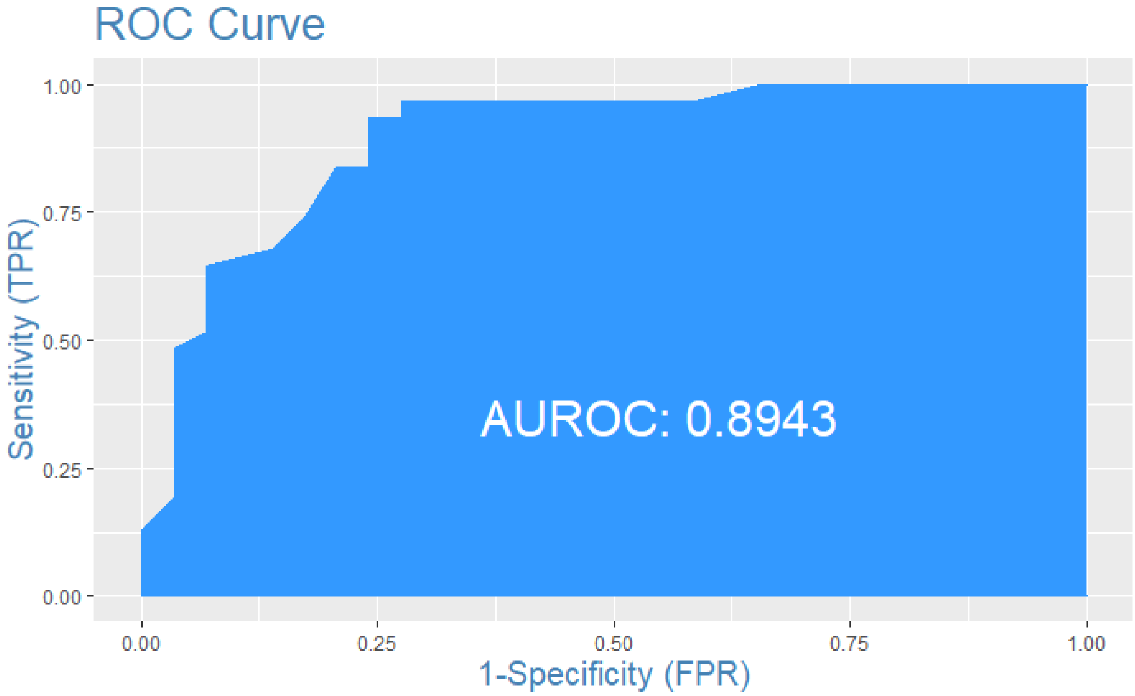
Calculate specificity

```
## [1] 0.7586207
```

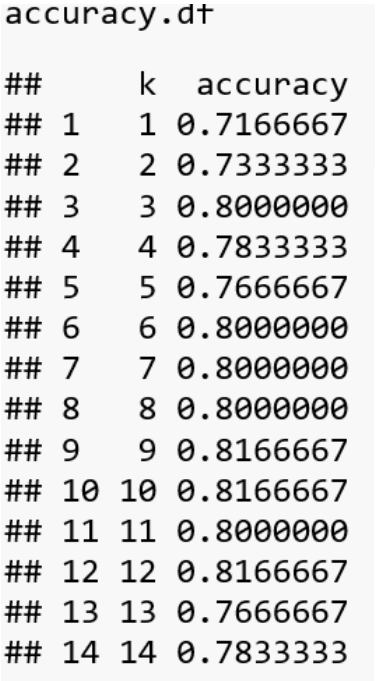
Calculate total misclassification error rate

```
## [1] 0.15
```

ROC curve

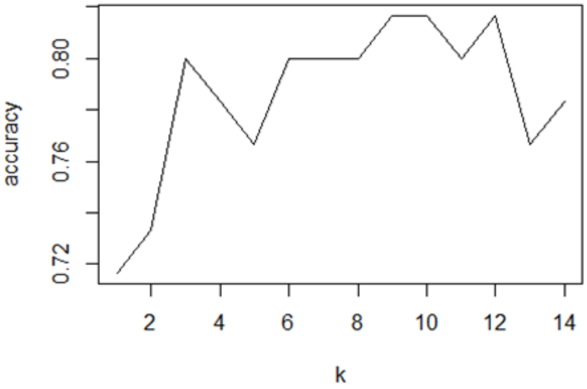


3. KNN model
   In KNN model, we first normalize the dataset then run KNN model to the dataset.

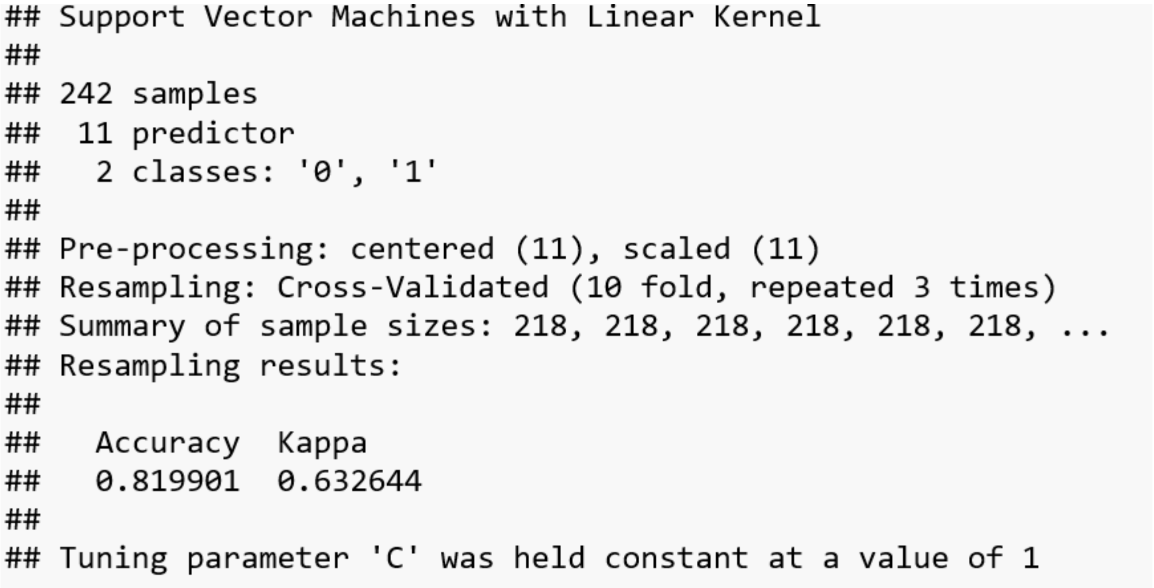


```
accuracy.df

##     k  accuracy
## 1   1 0.7166667
## 2   2 0.7333333
## 3   3 0.8000000
## 4   4 0.7833333
## 5   5 0.7666667
## 6   6 0.8000000
## 7   7 0.8000000
## 8   8 0.8000000
## 9   9 0.8166667
## 10 10 0.8166667
## 11 11 0.8000000
## 12 12 0.8166667
## 13 13 0.7666667
## 14 14 0.7833333
```



4. SVM model



```
## Support Vector Machines with Linear Kernel
##
## 242 samples
##  11 predictor
##   2 classes: '0', '1'
##
## Pre-processing: centered (11), scaled (11)
## Resampling: Cross-Validated (10 fold, repeated 3 times)
## Summary of sample sizes: 218, 218, 218, 218, 218, 218, ...
## Resampling results:
##
##   Accuracy  Kappa
##   0.819901  0.632644
##
## Tuning parameter 'C' was held constant at a value of 1
```

V. Performance Evaluation

In the KNN model, the best accuracy is 0.816667 when K=9.
The performance evaluation will be measured by how accurate a model is. Using ROC curve would be a preferable method to compare accuracy between different models.
One other thing to mention is that we also preformed analysis using Random forest model SVM model. But there is no output from these two models. We would put the code in R in the Appendix part.

The Area Under the Receiver Operating Characteristics of the classification tree model and logistic regression model are 0.8679 and 0.9249. The accuracy of SVM model is 0.819901. The result indicate that the logistic regression tree has a better accuracy on prediction.

VI. Discussion and Recommendation

The advantage of using logistic regression model is that it is very convenient to implement and understand. It also shows efficiency in training. Secondly, it can reach to multinomial regression easily, the class predictions' natural probabilistic view. Third, logistic regression model can output coefficient size and the direction of association, whether positive or negative. Moreover, it has good accuracy dealing with linear separable dataset.

However, there are few disadvantages come with logistic regression. First, it can't solve non-linear problems. Second, the model is assuming there is a linear relationship between the dependent and independent variables. Third, the model has constructs linear boundaries.

In this analysis, variables in this dataset are most categorical variables and they show strong linear relationship, thus the logistic regression model is a good choice.

For patients who have high level of chest pain, who is older than 60 and high maximum heart rate should be aware of danger of heart attack.

If given more time and space, we could run more algorithms on analyzing the dataset and dig more into data mining with R.

Moreover, we would further improve each model, to fix problem like no prediction score to output ROC plot in KNN model and SVM model.

VII. Summary

This report analyzes a dataset related to heart health, including 14 variables recorded by hospital. The purpose is to choose a model which can better predict if a patient would get heat attack with the information provided to the hospital. After comparing several model performances, the classification tree model works the best in this problem.

Appendix: R Code for use case study

~~~~
---
title: "IE 7275 Project"
author: "Chenghan Yue"
date: "4/17/2021"
output: word_document
---



```{r}
library(dummies)
library(dplyr)
library(caret)
~~~~

~~~
library(rpart)
library(rpart.plot)
library(forecast)
library(ggplot2)
library(InformationValue)
library(ISLR)
library("Hmisc")
library(corrplot)
library(psych)
library(reshape2)
library(grid)
library(gridExtra)
library(randomForest)
library(pROC)
library(e1071)
library(ROCR)
library(FNN)
```

```{r}
## import the data
heart = read.csv("C:/IE 7275/Project/heart.csv", header = T)

## Create dummy variables for the categorical predictors (Sex and chest pain type)
heart.dummy = dummy.data.frame(heart, sep = ".", dummy.classes = "factor")
heart.dummy = heart.dummy[,-c(1,2)]



```



```{r}
## Visualized data
### heart attack output vs. age
ggplot(heart) + geom_histogram(aes(x = age), binwidth = 0.5)
### heart attack output vs. chest pain
ggplot(heart) + geom_histogram(aes(x = cp), binwidth = 0.5)

## Data correlation Plot
heart.cor = cor(heart)
heart.cor = cor(heart, method = c("spearman"))
corrplot(heart.cor)

## Parallel Analysis Scree Plots\
fa.parallel(heart, n.iter = 100,show.legend = F, main = "Scree plot with parallel analysis")
~~~

~~~
## PCA
pca_heart = prcomp(heart, center = T, scale. = T)
summary(pca_heart)

```




```{r}
## Split the data into training (80%), validation(20%)
set.seed(7275)
train_index = sample(nrow(heart.dummy), 0.8*dim(heart))
valid_index = sample(setdiff(rownames(heart.dummy), train_index), 0.2*dim(heart)[1])
valid_index = as.numeric(valid_index)
train.df = heart.dummy[train_index,]
valid.df = heart.dummy[valid_index,]

```



```{r}
## fitting decision tree classification model

## Run classifictation tree
rt = rpart(output~ cp + trtbps + chol + fbs + restecg + thalachh + exng + oldpeak + slp +
caa + thall, data = train.df, method = "class", minbucket = 1, maxdepth = 30, cp = 0.001)
prp(rt)

## find the three or four most important car specifications for predicting the heart attack
output
t(t(rt$variable.importance))


```


```{r}
## Predicting Model on Test Data Set
predrt = predict(rt, newdata = valid.df, type = "prob")

#plot the ROC curve
plotROC(valid.df$output, predrt)


```
~~~

~~~~
```{r}
## Logistic Regression
logistic = glm(output~., data = train.df, family = "binomial")
summary(logistic)
exp(coef(logistic))
## propensities
logistic_pred = predict(logistic, valid.df, type = "response")
#find optimal cutoff probability to use to maximize accuracy
optimal  = optimalCutoff(valid.df$output, logistic_pred)[1]
confusionMatrix(valid.df$output, logistic_pred)
#calculate sensitivity
(sensitivity(valid.df$output, logistic_pred))
#calculate specificity
(specificity(valid.df$output, logistic_pred))
#calculate total misclassification error rate
(misClassError(valid.df$output, logistic_pred, threshold = optimal))
#plot the ROC curve
plotROC(valid.df$output, logistic_pred)
```
```{r}
## SVM
train.df$output = as.factor(train.df$output)
trctrl <- trainControl(method = "repeatedcv", number = 10, repeats = 3)

svm_Linear <- train(output ~., data = train.df, method = "svmLinear", trControl=trctrl,
preProcess = c("center", "scale"), tuneLength = 10)
svm_Linear ## Therefore, it just tested at value “C” =1.





```



```{r}
## KNN
## Normalized
norm.values <- preProcess(train.df[, -c(12)], method=c("center", "scale"))
train.df[, -c(12)] <- predict(norm.values, train.df[, -c(12)])
valid.df[, -c(12)] <- predict(norm.values, valid.df[, -c(12)])
heart.dummy[, -c(12)] = predict(norm.values, heart.dummy[, -c(12)])

cl = train.df$output
~~~~

~~~
accuracy.df = data.frame(k = seq(1,14,1), accuracy = rep(0,14))
for (i in 1:14) {
 KNN_b = knn(train = train.df[,-12], test = valid.df[,-12], cl, k = i, prob
= T )
## accuracy.df[i,2] = confusionMatrix(KNN_b, as.factor(valid.df[,12]))$
 accuracy.df[i,2] = sum(KNN_b==valid.df[,12])/nrow(valid.df)
}
accuracy.df
plot(accuracy.df,type = "l")




```



```{r}
predict_data = data.frame(cp = 0.05, trtbps = 0.7, chol = 1, fbs = 1.578, restecg = 0.975,
thalachh = 1.02594, exng = 1.41598, oldpeak = 0.375, slp = 0.5783, caa = 1.19735, thall
= 1.14536)
(output_pred = predict(rt, predict_data))

```
~~~