PROJECT REPORT

STA3241: Statistical Learning

Zheng, Zhi

Zhizheng0889@floridapoly.edu

Table of Contents

a. Introduction **2**

b. Literature Review **2**

c. Data **3**

d. Explanatory Data Analysis **4**

e. Methodology **6**

f. Results 7

1. Logistic Regression 7
2. Random Forest 8
3. K-Nearest Neighbor 8

g. Conclusion **9**

h. Executive Summary **10**

Work Cited **11**

Appendix **12**

a. Introduction

Death is the irreversible ending of all vital functions, especially as indicated by the permanent stoppage of the heart, respiration, and brain activity. Most people instinctively want to avoid a situation to the most extended length as possible in the game we call life. Thus, death prevention has been a highly prioritized goal of most humans from the beginning of civilization to present humanity and probably to the end of society. Heart disease is one of the leading causes of death for people of most racial and ethnic groups in the United States. Based on CDC, about 659,000 people in the United States die from heart disease each year—that's 1 in every four deaths [1]. Therefore, the topic I picked for my project is predicting heart disease with machine learning.

My project's dataset comes from Kaggle.com (Heart Disease Dataset), provided by David Lapp, last updated three years ago. This dataset dates from 1988 and consists of four databases: Cleveland, Hungary, Switzerland, and Long Beach V. It contains 76 attributes, including the predicted attribute, but all published experiments refer to using a subset of 14 of them. The "target" field refers to heart disease in the patient. It is integer-valued 0 = no disease and 1 = disease [2].

b. Literature Review

As previously mentioned, heart disease is one of the leading causes of death for people of most racial and ethnic groups in the United States. Thus, using data science to predict heart disease is commonly explored and researched in both data science and medical science communities. Almost annually, there is new research on how to expect heart disease and related fields more accurately. Some recently published works are "Data Science Framework - Heart Disease Predictions, Variant Models and Visualizations," published in the 2020 International Conference on Computer Science, Engineering and Applications (ICCSEA), and "An Automated Diagnostic System for Heart Disease Prediction Based on } Statistical Model and Optimally Configured Deep Neural Network" published in 2019 on IEEE Xplore All Journals (Liaqat et al.; Chittampalli Sai et al.).

The "Data Science Framework - Heart Disease Predictions, Variant Models and Visualizations" research's goal is to develop a data science framework to address the how to discover the chances of the existence of heart disease by applying different classification algorithms, influence and distribution of various parameters are playing a significant role in disease prediction along with visualizations on Cleveland cardiovascular medical records (Chittampalli Sai et al.). The authors mainly aim to find the optimal classification algorithm for the heart disease affected health records and majorly influencing parameters. They used and tested various data science algorithms such as Random Forest, Vector support, Logistic regression, and XG-Boost to build the heart disease prediction model and evaluate the model's performance in their research (Chittampalli Sai et al.).

The "An Automated Diagnostic System for Heart Disease Prediction Based on } Statistical Model and Optimally Configured Deep Neural Network" study's goal is to refine features of automated decision support systems based on artificial neural network

(ANN) and eliminate the problems posed by the predictive model to predict heart disease better. The authors managed to create a proposed model achieving the prediction accuracy of 93.33%. The study suggests that physicians can accurately use the proposed diagnostic system to predict heart disease (Liaqat et al.).

Both pieces of research mentioned above show how researchers are still slowly improving and expanding on predicting heart disease with modern technologies. However, my study will be much simpler due to the time constraint and limitations of my current skill in data science.

c. Data – description of the dataset, including the variables, their types, etc.

My project's dataset comes from Kaggle.com (Heart Disease Dataset), provided by David Lapp, last updated three years ago. It contains 14 variables and 1025 observations:

Variable Name	Data Type	Description
age	numerical	age in years
sex	categorical	(1 = male; 0 = female)
cp	categorical	chest pain type
trestbps	numerical	resting blood pressure (in mm Hg on admission to the hospital)
chol	numerical	serum cholesterol in mg/dl
fbs	categorical	(fasting blood sugar & gt; 120 mg/dl) (1 = true; 0 = false)
restecg	categorical	resting electrocardiographic results
thalach	numerical	maximum heart rate achieved
exang	categorical	Exercise-induced angina (1 = yes; 0 = no)
Old peak	numerical	ST depression induced by exercise relative to rest
slope	categorical	the slope of the peak exercise ST segment
ca	categorical	number of major vessels (0-3) colored by fluoroscopy

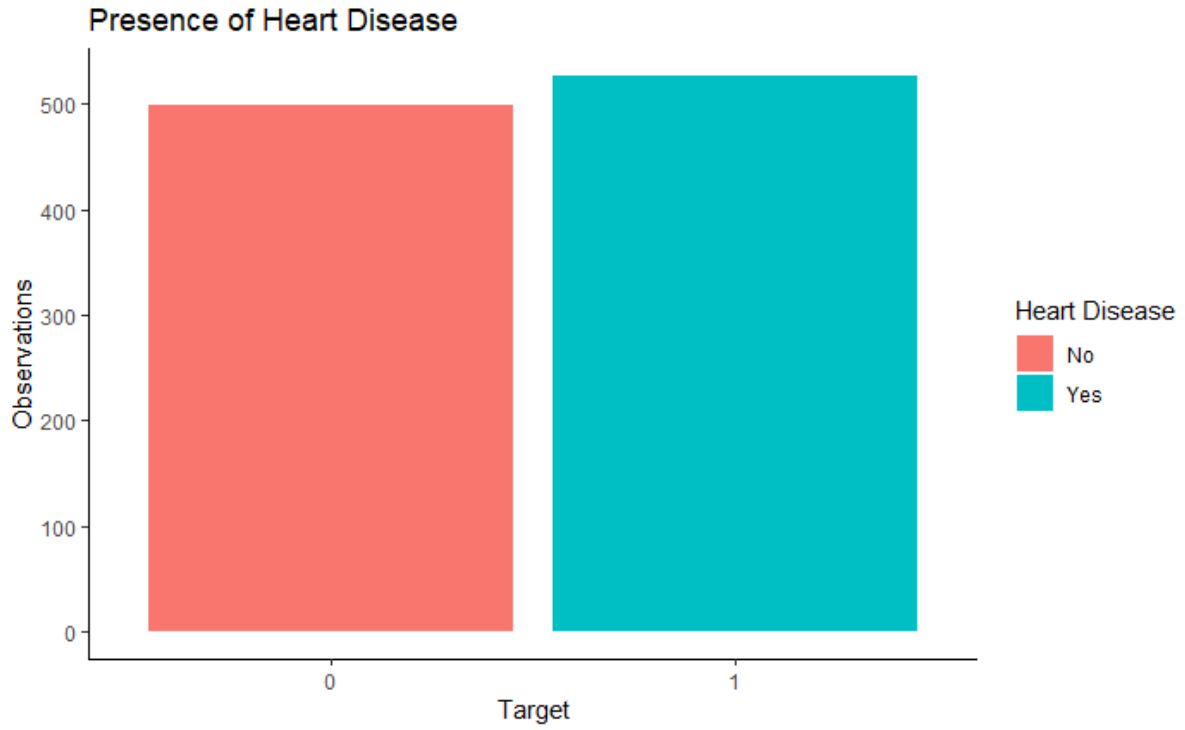
thal	categorical	Thalium stress result: 1 = normal; 2 = fixed defect; 3 = reversable defect
target	categorical	Do the patient have heart disease: 0 = no disease and 1 = disease

Table 1: Data Dictionary

d. Exploratory Data Analysis

The project aims to predict if a patient might or might not have heart disease through machine learning. Hence, the model's dependent variable will be the target variable, a categorical variable of "1=having disease" and "0=having no disease". The first step conducted of the EDA is to check all the variables to ensure there are no missing values in the dataset. Through the usage of the "which(is. na(df$variable), arr.ind=TRUE)" method to check if there are any missing values in each variable column, there is no missing value discovered in the dataset. Then I need to convert the categorical variables that are misread and read as numeric variables into factor variables.

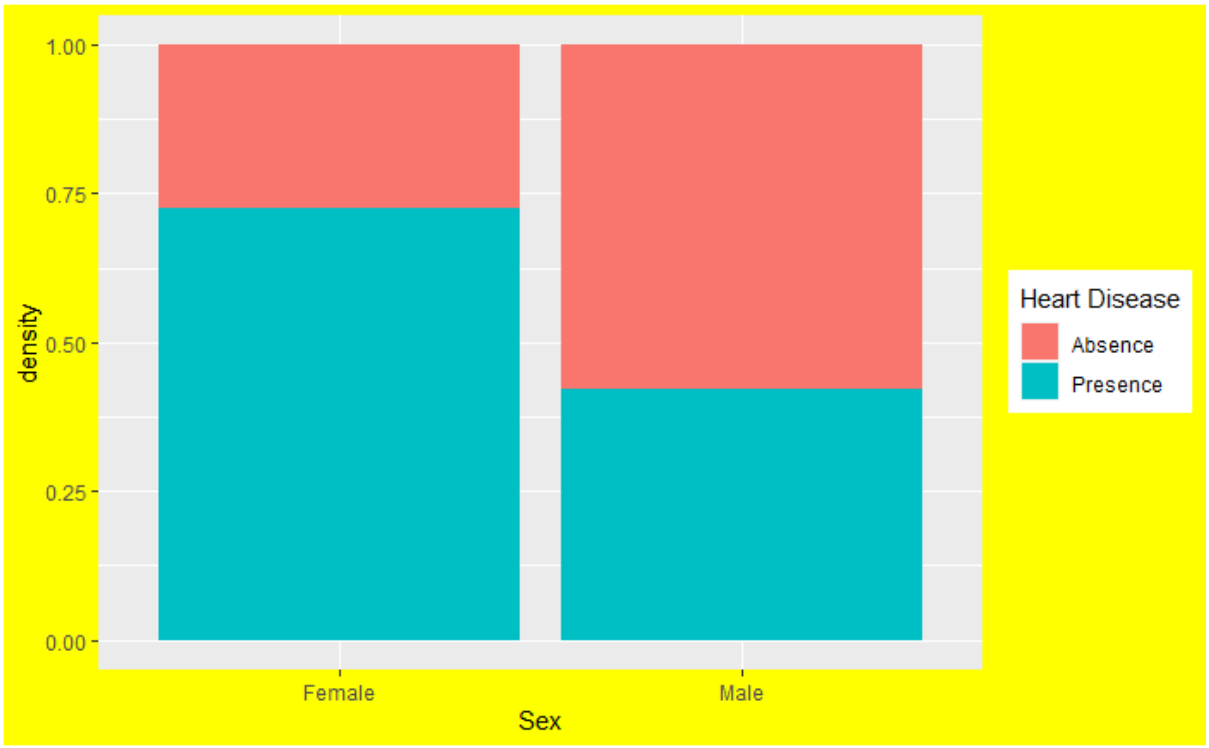
The second step of EDA is checking the "target" variable's observations to confirm it has a balanced amount of 0s and 1s comments. As the bar chart below shows:



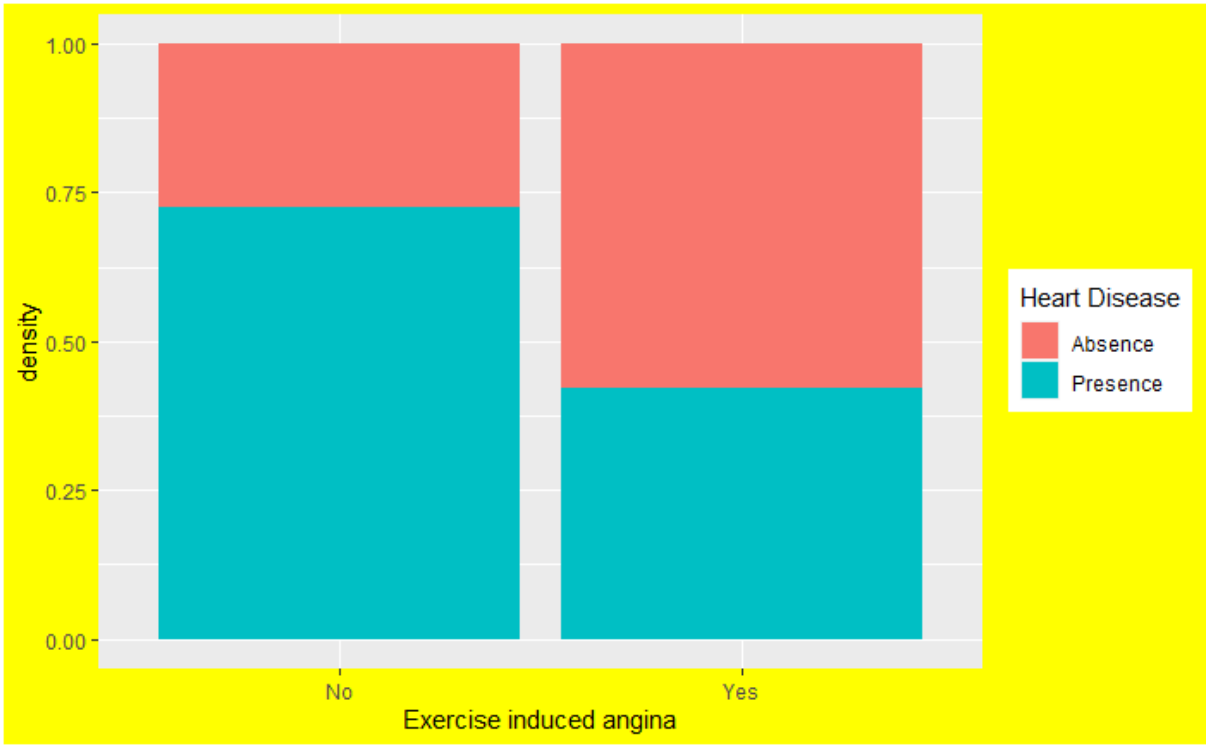
Graph 1: Target Variable Bar Graph

The 0s and 1s observations for the "target" variable are quite closely balanced, meaning the prediction result using this dataset as training will less likely be biased. Finally, the last step

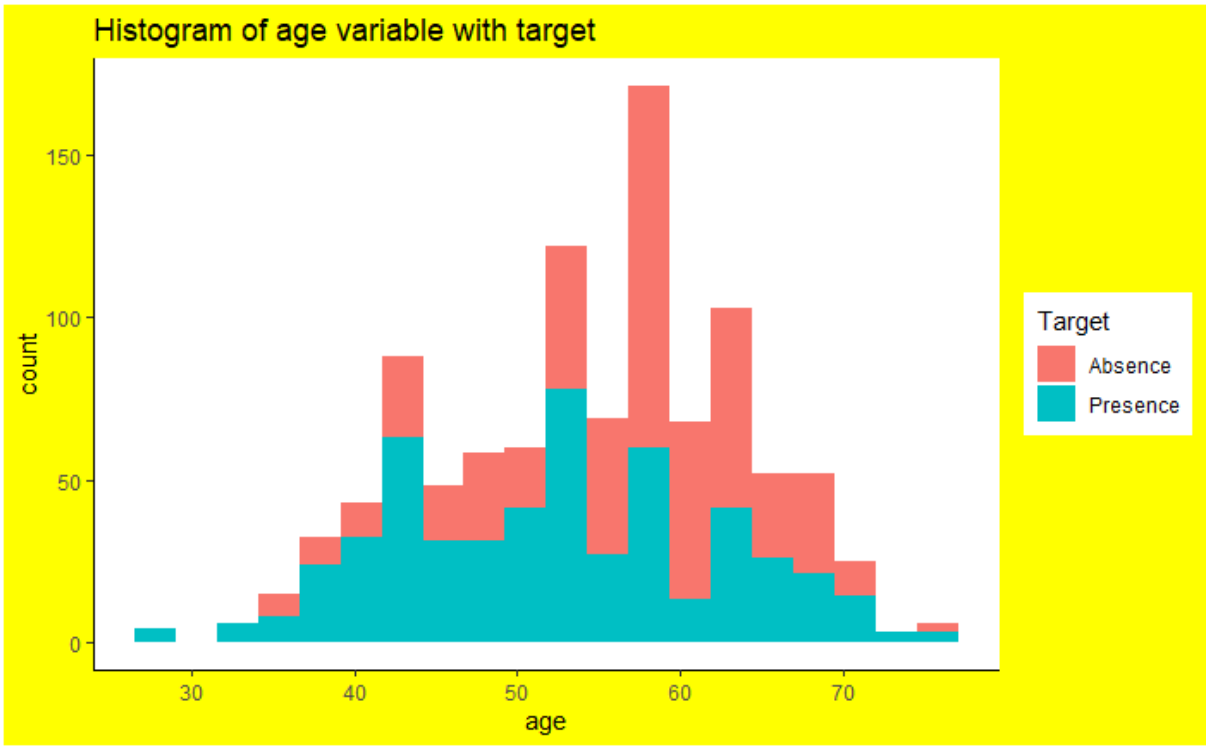
of EDA conducted is to check the tabular and visual statistics of the correlation between the "target" variable and other variables. The plots below show some of the results from the graphical statistics:



Graph 2: Sex VS Target



Graph 3: Exang Vs. Target



Graph 4: Age Vs. Target

Based on the three graphs above shows, that target is highly correlated to the sex, exang, and age variables. Graph 2 shows the female gender has a higher presence of heart diseases than the male gender. In Graph 3, patients without exercise-induced angina tend to have heart disease more likely. In Graph 4, patients between 42.5 to 60 are more likely to have heart disease than other age groups. These are just parts of the visual statistics EDA on the data, but it shows patterns/correlations between the target variable and other variables.

e. Methodology

The response variable "target" is a categorical variable. Thus classification-related methods like logistic regression can be used to produce predictions. The Stepwise Logistic Regression, Random Forest, and the KNN method are used to predict if a patient has heart disease or not in this project.

The stepwise logistic regression consists of automatically selecting a reduced number of predictor variables to build the best-performing logistic regression model. This is done through iteration, going through repeated rounds or cycles of analysis to arrive at the model with the best results. I ran this method by first logistic regressing a full model targeting the "target" variable. Then I use the stepAIC() function to find the best model with a stepwise process based on the AIC (Akaike information criterion). Afterward, I logistically regress the best fit model result from the stepAIC() with the train dataset (train_set). Finally, plot the result onto a ROC curve and calculate the prediction accuracy with a confusion matrix based on the test dataset (test_set).

Random forest is a supervised machine learning algorithm used for classification and regression modeling. It builds decision trees on different samples and takes their majority vote for classification. The method can handle categorical features very well. I ran the model with the randomForest() approach with mtry=6 and ntree=12 with the training dataset (train_set). Then calculate the probability and prediction with the test dataset (test_set). Finally, plot the result onto a ROC curve and calculate the prediction accuracy with a confusion matrix.

K Nearest-Neighbor (KNN) classifiers are non-model-based supervised machine learning algorithm classifiers, assuming that similar things exist in close proximity. KNN tries to predict the correct class for the test data by calculating the distance between the test data and all the training points. I ran the model with the kknn() method with k=13 and distance=2 with the training dataset (train_set). Then plot the result onto a ROC curve and calculate the prediction accuracy with a confusion matrix based on the test dataset (test_set).

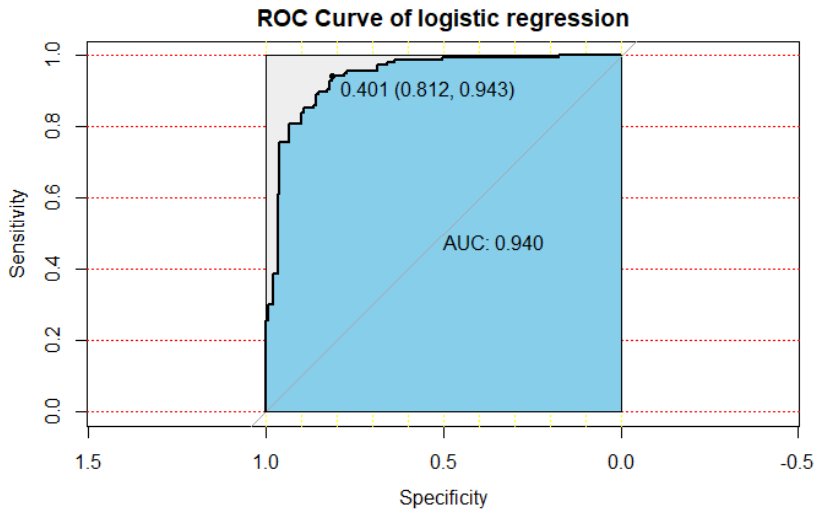
f. Results

1. Stepwise Logistic Regression

After using the stepAIC() function on the full logistic regress model to find the best fit model, I find the model "target ~ age + sex + cp + trestbps + chol + thalach + exang + oldpeak + slope + ca + thal" has the lowest AIC result, thus indicating it is the best fit model. Then I logistic regressed the best fit model and used it to predict on the test_set. Finally, I find the model has an AUC of .940 and prediction accuracy of 81.373%.

		predicted value	
actual value		0	1
	0	143	6
	1	51	106

Matrix 1: Logistic Regression's Confusion Matrix



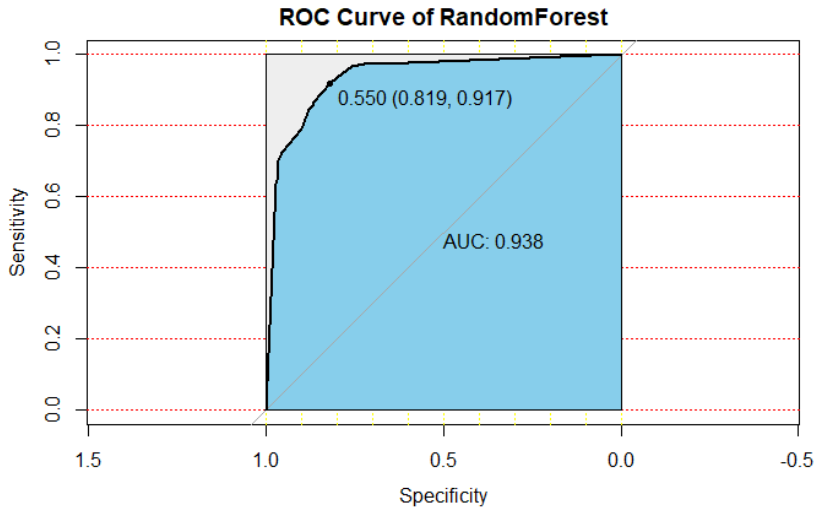
Graph 5: Logistic Regression's ROC Curve

2. Random Forest

After running the model with the randomForest() method with mtry=6 and ntree=12 with the training dataset (train_set), calculate the probability and prediction with the test dataset (test_set). I find the model has an AUC of .938 and a prediction accuracy of 86.275%.

	predicted value	
actual value	0	1
0	129	20
1	22	135

Matrix 2: Random Forest's Confusion Matrix



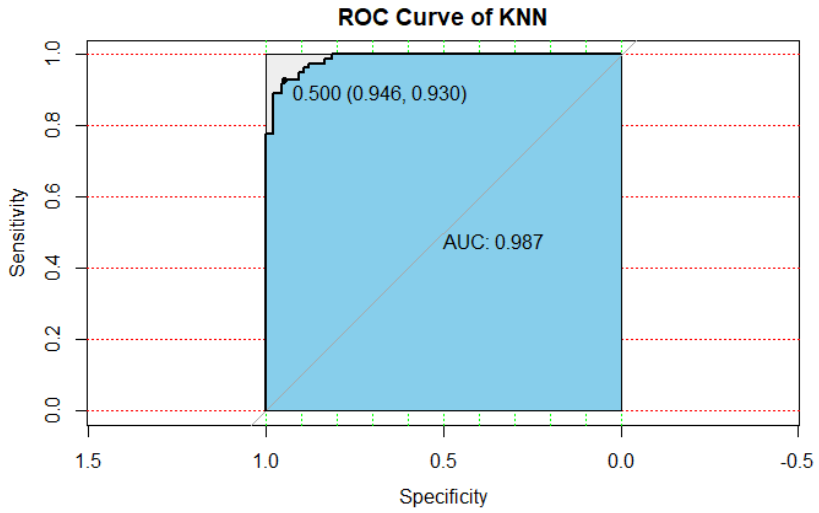
Graph 6: Random Forest's ROC Curve

3. KNN(K-Nearest Neighbors)

After running the model with the kknn() method with k=13 and distance=2 with the training dataset (train_set), calculate the probability and prediction with the test dataset (test_set). I find the model has an AUC of .987 and prediction accuracy of 89.869%

	predicted value	
actual value	0	1
0	146	3
1	28	129

Matrix 3: KNN's Confusion Matrix



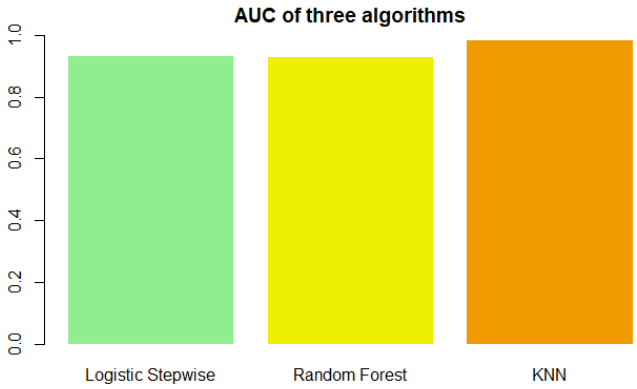
Graph 7: KNN's ROC Curve

g. Conclusions

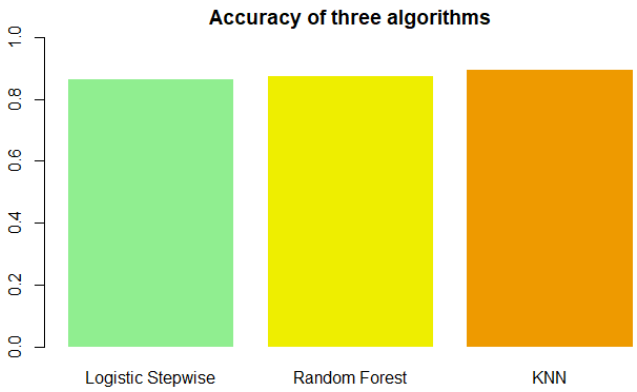
After getting the results from the three different machine learning methods for predicting heart disease, I compared the results. I concluded that KNN produced the best result with AUC=0.987 and 89.869% of accuracy. The study has two significant limitations that can be improved in future works. The first limitation is that the dataset was last updated three years ago, which is not as up-to-date as possible, and the number of observations is a bit lower than what I would like. The second limitation of the study is my skillset; I hope to improve and expand on this study in the future when I strengthen my skillset even further.

h. Executive Summary

The goal of the study is to find the best machine learning method to predict the presence of heart disease. Three different machine learning methods that deal with classification have been used in the study to predict the presence of heart disease. The methods are Stepwise Logistic Regression, Random Forest, and KNN (K-nearest neighbor). The results show that the KNN method produces the better AUC and accuracy results compared to the two other methods.



Graph 8: AUC Comparison



Graph 9: Accuracy Comparison

Works Cited

[1] "Heart Disease Facts." *Centers for Disease Control and Prevention*, Centers for Disease Control and Prevention, 7 Feb. 2022, https://www.cdc.gov/heartdisease/facts.htm.

[2] Ali, Liaqat, et al. "An Automated Diagnostic System for Heart Disease Prediction Based on } Statistical Model and Optimally Configured Deep Neural Network." IEEE Access 7 (2019): 34938–34945. Web.

[3] Lapp, David. "Heart Disease Dataset." *Kaggle*, 6 June 2019, https://www.kaggle.com/datasets/johnsmith88/heart-disease-dataset.

[4] Prakash, Chittampalli Sai, et al. "Data Science Framework - Heart Disease Predictions, Variant Models and Visualizations." 2020 International Conference on Computer Science, Engineering and Applications (ICCSEA). IEEE, 2020. 1–4. Web.

Appendix A

```
---
title: "Project"
output:
  pdf_document: default
  html_notebook: default
---
```r
#Packages
library(car)
library(caret)
library(class)
library(corrplot)
library(dplyr)
library(ggplot2)
library(gridExtra)
library(h2o)
library(kknn)
library(MASS)
library(plyr)
library(pROC)
library(ROCR)
library(rpart.plot)
library(rpart)
library(randomForest)

```



```r
#Importing the data
heart <- read.csv("C:/Users/zhizh/Desktop/Statistical Learning/Project/heart.csv")

heart$sex <- as.factor(heart$sex)
heart$cp <- as.factor(heart$cp)
heart$fbs <- as.factor(heart$fbs)
heart$restecg <- as.factor(heart$restecg)
heart$exang <- as.factor(heart$exang)
heart$slope <- as.factor(heart$slope)
heart$ca <- as.factor(heart$ca)
heart$target <- as.factor(heart$target)
heart$thal <- c(scale(heart$thal))

str(heart)
```
```

```
```

```{r}
#EDA

#Check if there are any missing values
which(is.na(heart$age), arr.ind=TRUE)
which(is.na(heart$sex), arr.ind=TRUE)
which(is.na(heart$cp), arr.ind=TRUE)
which(is.na(heart$trestbps), arr.ind=TRUE)
which(is.na(heart$chol), arr.ind=TRUE)
which(is.na(heart$fbs), arr.ind=TRUE)
which(is.na(heart$restecg), arr.ind=TRUE)
which(is.na(heart$thalach), arr.ind=TRUE)
which(is.na(heart$exang), arr.ind=TRUE)
which(is.na(heart$oldpeak), arr.ind=TRUE)
which(is.na(heart$slope), arr.ind=TRUE)
which(is.na(heart$ca), arr.ind=TRUE)
which(is.na(heart$thal), arr.ind=TRUE)
which(is.na(heart$target), arr.ind=TRUE)
which(is.na(heart$age), arr.ind=TRUE)
which(is.na(heart$age), arr.ind=TRUE)

```

```{r}
#Splitting data into train and test dataset
set.seed(666)
intrain <-createDataPartition(y=heart$target, p=0.7, list=FALSE)
train_set <- heart[intrain, ]
test_set <- heart[-intrain, ]

```

```{r}
#target
ggplot(heart, aes(factor(target), fill = factor(target))) +
  geom_bar() +
  theme_classic() +
  labs(x = "Target", y = "Observations") +
  ggtitle("Presence of Heart Disease") +
  scale_fill_discrete(name = "Heart Disease", labels = c("No", "Yes"))

#cp
```

```
ggplot(heart, aes(factor(cp), fill = factor(cp))) +
  geom_bar() +
  theme_classic() +
  labs(x = "Chest Pain", y = "Observations") +
  ggtitle("Bar graph of Chest Pain variable") +
  scale_fill_discrete(name = "Chest Pain", labels = c("Asymptomatic", "Typical Angina","Atypical Angina","Non-anginal Pain"))

#sex
ggplot(heart, aes(factor(sex), fill = factor(sex))) +
  geom_bar() +
  theme_classic() +
  labs(x = "Sex", y = "Count") +
  ggtitle("Bar graph of Sex") +
  scale_fill_discrete(name = "Sex", labels = c("Female","Male"))

#fbs
ggplot(heart, aes(factor(fbs), fill = factor(fbs))) +
  geom_bar() +
  theme_classic() +
  labs(x = "fbs", y = "Count") +
  ggtitle("Bar graph of fasting blood sugar") +
  scale_fill_discrete(name = "Fbs > 120 mg/dl", labels = c("No","Yes"))


```

```{r}
#slope
ggplot(heart, aes(factor(restecg), fill = factor(restecg))) +
  geom_bar() +
  theme_classic() +
  labs(x = "slope", y = "Count") +
  ggtitle("The slope of the peak exercise ST segment") +
  scale_fill_discrete(name = "Resting elec", labels = c("Up","Flat","Down"))

#restecg
ggplot(heart, aes(factor(restecg), fill = factor(restecg))) +
  geom_bar() +
  theme_classic() +
  labs(x = "restecg", y = "Count") +
  ggtitle("Bar graph of resting electrocardiographic results") +
  scale_fill_discrete(name = "Resting elec", labels = c("Normal","Abnormal","Probable or definite"))
```

```
#ca
ggplot(heart, aes(factor(ca), fill = factor(ca))) +
  geom_bar() +
  theme_classic() +
  labs(x = "ca ", y = "Count") +
  ggtitle("Number of major vessels (0-3) colored by fluoroscopy") +
  scale_fill_discrete(name = "Vessels Colors", labels = c("Color 0","Color 1","Color 2", "Color 3", "Unknown"))

#thal
ggplot(heart, aes(factor(thal), fill = factor(thal))) +
  geom_bar() +
  theme_classic() +
  labs(x = "thal", y = "Count") +
  ggtitle("Thal") +
  scale_fill_discrete(name = "thal", labels = c("Unknown", "Normal", "Fixable Defect", "Reversable Defect"))

#exang
ggplot(heart, aes(factor(exang), fill = factor(exang))) +
  geom_bar() +
  theme_classic() +
  labs(x = "exang ", y = "Count") +
  ggtitle("Exercise-induced angina") +
  scale_fill_discrete(name = "(1 = yes; 0 = no)", labels = c("No", "Yes"))


```


```{r}
#age
plot1 <- ggplot(heart, aes(age))+geom_histogram()

#trestbps
plot2 <- ggplot(heart, aes(trestbps))+geom_histogram()

#chol
plot3 <- ggplot(heart, aes(chol))+geom_histogram()

#thalach
plot4 <-ggplot(heart, aes(thalach))+geom_histogram()
```

```
#oldpeak
plot5 <- ggplot(heart, aes(oldpeak))+
   geom_histogram(aes(y =..density..))+
   labs(x="ST depression Oldpeak",y="")



grid.arrange(plot1, plot2, plot3, plot4, plot5, ncol=2)
```

```{r}
#Sex with Target
sex1 = factor(heart$sex,labels=c("Female","Male"),levels=0:1)
target1 = factor(heart$target,labels=c("Absence","Presence"),levels=0:1)


ggplot(heart, aes(x = sex1, fill = target1))+geom_bar(position = "fill")+
        theme(plot.background = element_rect(fill = "yellow"))+
        scale_fill_discrete(name = "Heart Disease")+
        labs(x="Sex",y="density")

#Exang with Target
exang1 = factor(heart$sex,labels=c("No","Yes"),levels=0:1)

ggplot(heart, aes(x = exang1, fill = target1))+geom_bar(position = "fill")+
        theme(plot.background = element_rect(fill = "yellow"))+
        scale_fill_discrete(name = "Heart Disease")+
        labs(x="Exercise induced angina",y="density")

#Age with Target
ggplot(heart, aes(age, fill = factor(target)))+
   geom_histogram(bins = 20)+
   theme_classic()+
   theme(plot.background = element_rect(fill = "yellow"))+
   labs(title = "Histogram of age variable with target", x = "age", y = "count")+
   scale_fill_discrete(name = "Target", labels = c("Absence","Presence"))

```


```{r}
pairs(heart)
```

```{r}
#Logistic Regression Stepwise Modeling

log.model <- glm(target~.,family="binomial",data = train_set)
summary(log.model)

step.model <- stepAIC(log.model)
summary(step.model)

```
```{r}
# Logistic Regressing the best fit model base on the AIC score of stepAIC

glm2 <- glm(target ~ age + sex + cp + trestbps + chol + thalach + exang +
      oldpeak + slope + ca + thal, family = "binomial", data = train_set)
summary(glm2)

exp(coef(glm2))

vif(glm2)

glm.probs <- predict(glm2,newdata = data.frame(test_set),type = "response")
glm.pred <- ifelse( glm.probs > 0.827, "1", "0")
glm.roc <- roc(test_set$target,glm.probs)

plot(glm.roc, print.auc=TRUE, auc.polygon=TRUE, grid=c(0.1, 0.2), grid.col=c("yellow",
"red"), max.auc.polygon=TRUE, auc.polygon.col="skyblue", print.thres=TRUE, main='ROC
Curve of logistic regression')

table(test_set[,"target"], glm.pred, dnn=c("actual value","predicted value"))

#Accuracy = (143+106)/(143+6+51+106) = .81373 = 81.373% accuracy
```


```{r}
#Random Forest
rtree <- randomForest(target ~ age + sex + cp + trestbps + chol + thalach + exang +
      oldpeak + slope + ca + thal, data=test_set, mtry=6, importance = TRUE, ntree=12)

rtree

yhat_rf_heart <- as.data.frame(predict(rtree, data = test_set, type = "prob"))
```

```
rtree.roc <- roc(test_set$target,yhat_rf_heart[,1])


plot(rtree.roc, print.auc=TRUE, auc.polygon=TRUE, grid=c(0.1, 0.2), grid.col=c("yellow", "red"), max.auc.polygon=TRUE, auc.polygon.col="skyblue", print.thres=TRUE, main='ROC Curve of RandomForest')

predtree<-ifelse(yhat_rf_heart[, 2]>=0.6,1,0)
table(test_set$target, predtree, dnn=c("actual value","predicted value"))

#Accuracy = (129+135)/(129+20+22+135) = 0.86275 = 86.275% accuracy
```

```{r}
#KNN K Nearest Neighbor
#k=13,optimal
heart_knn <- kknn(target~.,train_set,test_set,k=13,distance=2)
knn_roc <- roc(test_set$target,heart_knn$prob[,2])

plot(knn_roc, print.auc=TRUE, auc.polygon=TRUE, grid=c(0.1, 0.2),grid.col=c("green", "red"), max.auc.polygon=TRUE,auc.polygon.col="skyblue", print.thres=TRUE,main='ROC Curve of KNN')

pre_knn =ifelse(heart_knn$prob[,2]>0.699,1,0)
table(test_set$target,pre_knn,dnn=c("actual value","predicted value"))

# Accuracy = (146+129)/(146+3+28+129) = .89869 =89.869% accuracy
```

```{r}
#Visualizing AUC result comparison

auc1 = c(0.940, 0.938, 0.987)
names = c("Logistic Stepwise","Random Forest","KNN")
barplot(auc1,ylim=c(0,1),names.arg = names,border="white",col=c("lightgreen","yellow2","orange2"),main="AUC of three algorithms")

#Visualizing accuracy result comparison

acc1 = c(0.81373, 0.86275, 0.89869)
names = c("Logistic Stepwise","Random Forest","KNN")
barplot(acc1,ylim=c(0,1),names.arg = names,border="white",col=c("lightgreen","yellow2","orange2"),main="Accuracy of three
```

algorithms")

```
```