Heart Disease Prediction using Logistic Regression

Comparative discussion using Python and R

Khandokar Faisal Islam
Bac ID: 2020262002

Semester: Fall 2020
School of communication, Business,
and Law
Leeds Trinity University, UK
Email: faisal2619@gmail.com

Abstract— The heart is the most important or vital organ of the human body. It is used to maintain conjugate blood in the human body. Every day there are many heart disease cases recorded all over the world. It is one of the main causes of death. Data science plays a significant role in processing massive data produces by the healthcare industry. Predicting heart disease is a very complex task. Conventional heart disease prediction is time-consuming. Thus, it is necessary to automate the prediction process to avoid the risks related to it and notify the patient in advance. This study will use the Cleveland heart disease dataset available in the UCI machine learning repository. The proposed system predicts the chances of heart disease by using Logistic machine learning techniques on both python and R data science platforms. Further, the outcome of the study will be analyzed through a comparative analysis using Python and R.

Keywords—Heart disease, logistic regression, Machine learning, data science, python, R

I. INTRODUCTION

A. Project aim and objectives:

The main aim of this project is to predict heart disease using a machine learning algorithm and compare predicted data with the analysis tool. Heart disease is a major cause of death all over the world. The incidence of heart disease is increasing dramatically. Symptoms of heart disease differ from gender to gender. The most common heart disease symptoms are chest pain, nausea, indigestion, heartburn stomach ache, etc. [1]. In a traditional method, doctors can't detect heart disease instantly. they need a physical examination and medical image to know the current state of the patient. Moreover, it needs experience and comprehensive knowledge of prediction. [2]. Machine learning has been broadly used in the healthcare industry for detecting and predicting various diseases using different data models. As we know if early detection is possible, it can save a life, time, and money. Predicting heart disease is a classification problem. Hence, logistic regression has been selected, it will produce a better result for data analysis. Two popular data science platforms or tools Python and R will be used. Cleveland heart disease dataset will be collected, prepared, visualized, and analyzed for the logistic regression prediction model. The outcome of the prediction will lead to a comparative discussion between python and R data analysis tools.

B. Dataset:

The name of the Dataset is “Cleveland Heart disease” data set, it is collected from the UCI repository website (https://archive.ics.uci//ml/datasets/heart+disease) from an ongoing study on heart disease conducted by the Hungarian Institute of Cardiology and the University of Zurich. Dataset was collected from UCI because it is one of the largest and most reliable sources for collecting datasets. Students, Educators, and Researchers all over the world consider UCI or UC Irvine machine learning repository as a primary source of machine learning datasets. UCI is continuously updating its repository. Due to its authenticity and reliability, most people prefer UCI for collecting datasets.
This heart dataset consists of 303 separate data, 13 features, and one target variable. To visualize the dataset as a data frame, see the appendix Figure:1

C. Business Value of the Project:

Heart disease is one of the main causes of death all over the world [2]. It is a complex task for doctors to detect heart disease accurately and efficiently. This machine learning heart diseases prediction system has been developed to help doctors, medical administrators to detect heart disease. The analytical data will help to decrease the number of tests that need to be taken by the patient. This system can add value to the medical and health care industry.

D. Analytical Technique and Platforms:

The whole data preparation, exploration, visualization, and analysis has been carried out on both Python and R environments using the Logistic Regression machine learning technique. Predicting heart disease is an instance of supervised machine learning and it is also a classification problem. Hence, the Logistic regression supervised machine learning technique is appropriate for this project to predict heart disease. Python and R are excellent open-source programming languages for a data science project. Moreover, they have rich libraries and functionality for data preparation, visualization, and analysis. Machine learning techniques like logistic regression works perfectly on these platforms. Thus, it provides more accuracy than other techniques.

II. Theoretical Feasibility of Logistic Regression

A. *Machine learning:*

Machine learning is the idea that enables computer programs or systems to learn and improve automatically without human involvement. There is no need for explicit programming [5]. Machine learning algorithms use historical data to learn and improve and predict outcomes more accurately Machine learning is mainly of four types:

Supervised Machine learning: In this machine learning algorithm, only labeled data used for training and predicting the accurate outcome [6].

Unsupervised machine learning: This algorithm involves training unlabeled data. It analyzes the whole dataset to draw a meaningful connection.

Semi-Supervised machine learning: Both labeled and unlabeled used for training, usually a small amount of labeled data and large amounts of unlabeled data [7].

Reinforcement Learning: This algorithm is about taking appropriate action to maximize reward in a specific situation. It is used to find the best possible behavior or approach for a particular situation through different software and machines

Logistic Regression:

Logistic regression is a very well-known machine learning algorithm borrowed from the statistic. It is a supervised machine learning technique used to analyze a dataset with one or more independent variables that predict the outcome of the categorical dependent variables. It converts outcomes using the logistic or sigmoid function to return a probability value and this value can be mapped into two or more discreet groups. The logistic regression can be classified into three types Binary logistic regression, Multinomial logistics regression, Ordinal Logistic regression [8].

How logistic regression works:

In general, logistic regression is a supervised classification algorithm. For a classification problem, the target variable (or output) is y, it can only use discrete values for a certain set of functions, X. The logistic regression model uses a more complex cost function which is also known as the sigmoid function or logistic function. The sigmoid function or logistic function is a function that is identical to an "S"-shaped curve when graphically drawn. It takes values from 0 to 1 and "cuts" them to the top and bottom edges, and sticks them to 0 or 1. Logistic regression restricts the cost function between 0 and 1. Like Linear regression, it uses an equation as the representation.

y = e^(b0 + b1*x) / (1 + e^(b0 + b1*x))

The function above contains variables b0 and b1. These variables are called weights or coefficient values. b0 describes the bias or intercept and b1 represents the coefficient. These weights are learned and trained from the collected dataset. The outcome of the formula will generate a percentage or probability and it will be mapped through discrete classes. The specified separation of these two groups is known as the boundary of decisions [9]. You can see figure 11.

Advantages of Logistic Regression:

Logistic Regression is a very simple executable machine learning algorithm. It provides excellent training efficiency. The training model using logistic regression doesn’t require high computational power**.** Unlike decision trees and support vector machines, the logistic regression enables models to be updated easily to represent new data [10]. The predicted parameters (trained weights) show the significance of each characteristic. There is also a positive or negative direction of the association. Hence, we can use logistic regression to figure out the relationship within the features. It produces well-calibrated predictions and outcomes and this is an advantage over other models that only produce final classification as outcomes. Moreover, Logistic regression is less likely to overfitting into a low dimensional dataset with adequate numbers of training instances. It is very quick and easy to execute thus it is considered as a benchmark model to measure performance. If the dataset has features that can be linearly separable then logistic regression provides excellent accuracy. It has a close association with neural networks. Due to a simple probabilistic explanation, It takes less time for training. It is less complex than any other complex algorithms such as the artificial neural network. It is also known as multinomial logistic regression [10].

Limitations of Logistic Regression:

Logistic regression should not be used when the number of observations is lesser than the number of features. Therefore, it can lead to overfitting. It creates linear boundaries therefore Solving Non-linear cannot be possible. It is hard to found data in the real-world that linearly separable. Moreover, Logistic Regression cannot capture complex relationships. Hence powerful and complex algorithms like Neural networks can easily outperform logistic regression. It needs average or no multicollinearity within independent variables. Furthermore, Logistic regression requires the log odds to be connected linearly with the independent variables (log(p/(1-p)) [11].

B. Framework

Factors	Why it matters
Ease of Use	It ensures efficiency when performing the task. As it makes sure simple syntax that easy to understand and handle the error.
Available Library	By using different libraries different job or task can be accomplished
Visualization capability	Visualization capability enables information and data into visual context including the map, graph. It makes it easy to understand and analyze the data.
Computational speed	Good computational speed lets you accomplish your task within a short time. It makes it easy to work on a big dataset.
Community support	Community support is an advantage, it helps to clear knowledge and learn fundamentals and you can get a solution to any problem regarding your project
Cost	Less cost provides flexibility to the developer
Statistical Correctness	It helps to produce accurate statistics during data analysis.
System Integration	It brings together all the subsystem for accurate data analysis
Consistency	It enables developers to do more assumption and predictions

III. Practical Feasibility of analytical tool

Ease of Use:

R is not easy to learn and understand because it takes more time to understand the syntax and handle the error. If one has no or less experience in programming will find it very difficult to learn and implement. Whereas, Python is a very simple programming language with an easy syntax that anyone can learn, understand and implement easily, even those who have just begin programming can grasp it easily and learn quickly [12].

Available Library:

A library is a collection of functions that can be included in the program. It is very essential just like any other function. Python has extensive libraries for every task that ease the development process and saves time. However, R contains more library that is useful for data science and statistics, Python is mainly focused on machine learning, deep learning, and artificial intelligence. There are many power libraries for both Python and R, these libraries update regularly [13].

In this study, To prepare and process the dataset in Python, different libraries like Pandas, Numpy has been used. Whereas, tidyverse and Dplyr are used in the R environment. For dataset exploration, visualization, and analysis matplotlib, seaborn libraries are used in python, On the contrary, the ggplot2 library has been used in R to explore and visualize the dataset. To apply a machine learning model in python scikitlearn library used in python, whereas, caret and Randomforest libraries have been imported in the R environment [14].

You can visualize the code screenshot in Appendix Figure 2 and 3.

Visualization Capability:

Data visualizations are one of the fundamental parts of data analysis. It describes complex information in a manner that everyone can easily identify patterns and correlations. R has many advanced and powerful packages like ggplot2, Lattice for data visualization. These packages make complex raw data set understandable, informative, and eye-pleasing. Whereas, Python has also a large number of interactive packages like Matplotlib, seaborn, pydot, bokeh, etc. for visualizing the data. However, choosing the best and most relevant package can be a very intricate task for the data scientist. R provides better data visualization with less complexity in comparison to python [14].

The collected dataset has been visualized through both python and R. First, look at the data frames generated from the dataset by Python and R. Figure 1 and 4. As we can see that python and R has a different way to visualize the data for exploratory data analysis. If we look at the relationship graph figures 4, 5, 6,7, 8 with the target variable in both python and R.

Computational speed:

Both Python and R are used for data analytics but R is slower than python. You can analyze a large dataset through python with less code and less time. R was mainly created for the statistical task; it has many packages to improve performance. R needs more code to do a single task that reduces computational speed when handling a large dataset [15].

Community support:

R has big community support with more than two million users. There are thousands of developers and programmers around the world who are part of this R community. The programmer and member of stack overflow are also contributing to growing R. You can get more statistical solution on the R community. Popular community support for R such as R-studio community, R-statistic community, stack overflow community [16].

However, Python has also rich community support like R, it has comprehensive online community support without customer service support. The biggest advantage of python, there are millions of developers who can support you when you run into trouble or facing any code error. Python community provides strong support for machine learning, Artificial Intelligence. Popular python communities like pycon, pyslackers, StackOverflow, and so on available to support your work [17].

Statistical correctness:

R provides better support and libraries for statistics as it is specially developed for data statistics. Whereas, Python is best for deep learning, machine learning, and application development and deployment. So it can be assumed that R has more statistical correctness than python because the libraries of R implement a wide range of statistical and graphical techniques for data analysis [18].

However, predicting heart disease applying logistic regression in python provides more accuracy than R. After doing the confusion matrix the accuracy score of the Python logistic regression model is 0.9230, whereas R logistic regression model accuracy is 0.7444. Please see figures 9 and 10

Cost:

Both Python and R is an open-source programming language, most of the times the IDE and the environment used for python and R are comes with free of cost. However, there is some paid version of IDE can add additional cost while developing with Python and R. Pycharm is one of such IDE that offers both community and professional version. The community version comes free of cost whereas the professional version costs. On the contrary, R studio is a popular IDE for development in R, it also offers a paid and free version [18].

System Integration:

Python can be integrated with other languages it supports R functionality via the RPy2 package. For easy development, the python programs are integrated with web apps. On the other hand, R runs programs locally and integration can be very difficult. As R has complex syntax, particularly for the new users it is quite impossible to integrate with another language [19].

Consistency:

To add more functionality to the program, R heavily depends on third party code for many packages. It often creates an inconsistency within the available algorithms. Whereas Python is very flexible and versatile in this respect. The method of producing large packages with fewer features means that the code is consistent across the platform. As we know python provides great community support [18].

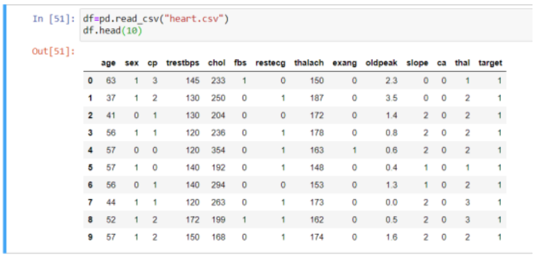
IV. Conclusion and Recommendations

The main intent of this study is to predict heart disease through machine learning data analysis techniques and platforms. Further, this study proposes a comparative study on the accuracy of the technique for multiple platforms. Logistic Regression has been used to analyze the Cleveland heart dataset with the help of data analysis tools or platforms python and R. The dataset contains 303 records with 14 necessary attributes with some missing values. At first, the raw dataset has been prepared to visualize and analyze the dataset through raw coding with the help of different libraries and packages imported on both python and R. As it is a classification problem, the outcome has been classified into two classes 1(one) and 0(zero). “1” indicates having heart diseases. Further work involves the development of the system using the mentioned technique and hence training and test the system. It produces a satisfactory accuracy score in both python and R platforms. As we know if the ROC curve is over 0.50 then it can be considered a good result. In this study logistic regression produces an accuracy score over 0.50 for both python and R. However, when the logistic regression is applied with python, it produces a more accuracy score of 0.9230 than R accuracy score of 0.7444. Moreover, if more data has been used then the model could have improved more. The result of the logistic regression shows that men have more chance to have heart disease than women. Age, number of cigarettes each day, and systolic blood pressure are the main risk of heart disease.

Though the Logistic Regression technique has produced a significant-good result, this heart disease prediction can be done using other machine learning techniques like Artificial Neural Network, Random Forest. The Artificial Neural networks perform well with the ensemble method to this dataset. And the Random forest can perform better with decision trees. However, an Artificial neural network can produce more precise results than any other technique. Moreover, instead of using confusion matrix validation, this study could have used k-fold cross-validation to find better accuracy in the trained and testing dataset.

V. Appendix

1. Figure: Dataframe Python

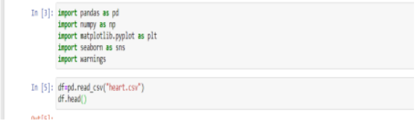


```
In [51]: df=pd.read_csv("heart.csv")
         df.head(10)
```

Out[51]:

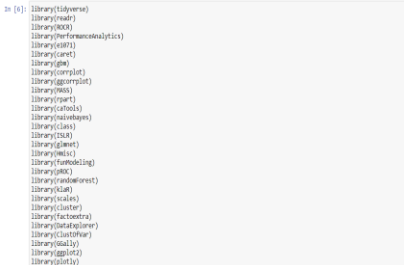
	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
1	37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
2	41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
3	56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
4	57	0	0	120	354	0	1	163	1	0.6	2	0	2	1
5	57	1	0	140	192	0	1	148	0	0.4	1	0	1	1
6	56	0	1	140	294	0	0	153	0	1.3	1	0	2	1
7	44	1	1	120	263	0	1	173	0	0.0	2	0	3	1
8	52	1	2	172	199	1	1	162	0	0.5	2	0	3	1
9	57	1	2	150	168	0	1	174	0	1.6	2	0	2	1

2. Figure: Python Libraries



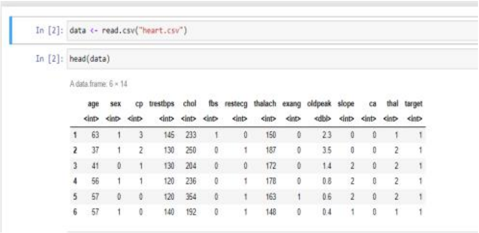
```
In [3]: import pandas as pd
        import numpy as np
        import matplotlib.pyplot as plt
        import seaborn as sns
        import warnings

In [5]: df=pd.read_csv("heart.csv")
        df.head()

Out[5]:
```

3. Figure: R Libraries:



```
In [6]: library(tidyverse)
        library(readr)
        library(ROCR)
        library(PerformanceAnalytics)
        library(e1071)
        library(caret)
        library(gbm)
        library(corrplot)
        library(ggcorrplot)
        library(MASS)
        library(rpart)
        library(caTools)
        library(naivebayes)
        library(class)
        library(ISLR)
        library(glmnet)
        library(Hmisc)
        library(funModeling)
        library(pROC)
        library(randomForest)
        library(klaR)
        library(scales)
        library(cluster)
        library(factoextra)
        library(DataExplorer)
        library(ClustOfVar)
        library(GGally)
        library(ggplot2)
        library(plotly)
```

4. Figure: Dataframe R

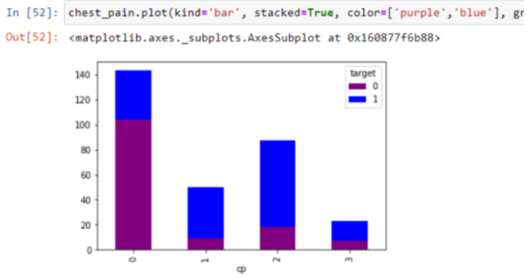


```
In [2]: data <- read.csv("heart.csv")

In [2]: head(data)
```

A data frame: 6 × 14

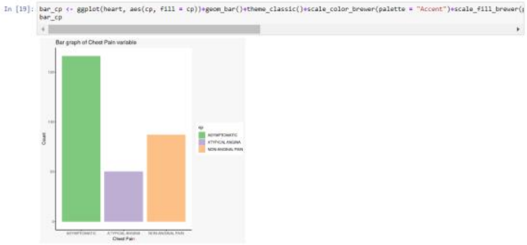
	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<dbl>	<int>	<int>	<int>	<int>
1	63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
2	37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
3	41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
4	56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
5	57	0	0	120	354	0	1	163	1	0.6	2	0	2	1
6	57	1	0	140	192	0	1	148	0	0.4	1	0	1	1

5. Chest Pain bar Python



```
In [52]: chest_pain.plot(kind='bar', stacked=True, color=['purple','blue'], gr
Out[52]: <matplotlib.axes._subplots.AxesSubplot at 0x160877f6b88>
```

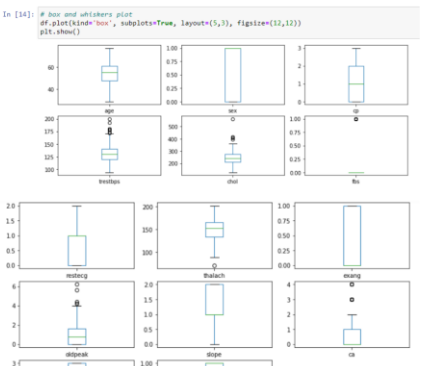
6. Figure: chest pain bar in R



```
In [19]: bar_cp <- ggplot(heart, aes(cp, fill = cp))+geom_bar()+theme_classic()+scale_color_brewer(palette = "Accent")+scale_fill_brewer(
         bar_cp
```

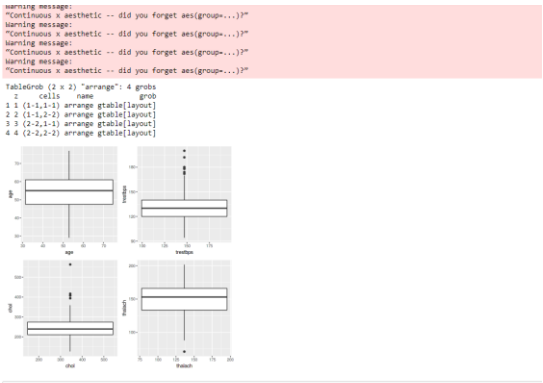
7. Figure: Boxplot python

with Histograms we can see the shape of each feature and provides the count of number of observations in each bin.



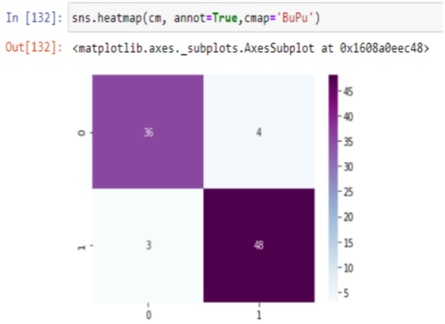
```
In [14]: # box and whiskers plot
         df.plot(kind='box', subplots=True, layout=(5,3), figsize=(12,12))
         plt.show()
```

8. Figure: Boxplot R

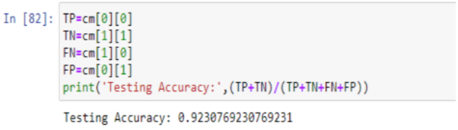


```
Warning message:
"Continuous x aesthetic -- did you forget aes(group=...)?"
Warning message:
"Continuous x aesthetic -- did you forget aes(group=...)?"
Warning message:
"Continuous x aesthetic -- did you forget aes(group=...)?"
Warning message:
"Continuous x aesthetic -- did you forget aes(group=...)?"
TableGrob (2 x 2) "arrange": 4 grobs
  z     cells    name           grob
1 1 (1-1,1-1) arrange gtable[layout]
2 2 (1-1,2-2) arrange gtable[layout]
3 3 (2-2,1-1) arrange gtable[layout]
4 4 (2-2,2-2) arrange gtable[layout]
```

9. Figure: Python confusion matrix

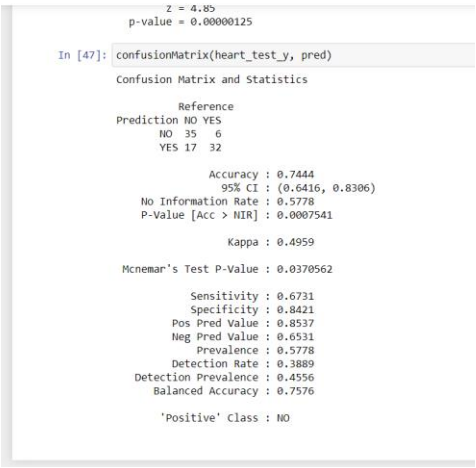


```
In [132]: sns.heatmap(cm, annot=True,cmap='BuPu')
Out[132]: <matplotlib.axes._subplots.AxesSubplot at 0x1608a0eec48>
```



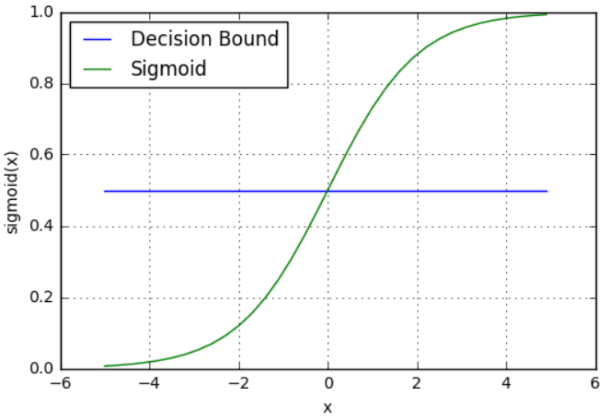
```
In [82]: TP=cm[0][0]
         TN=cm[1][1]
         FN=cm[1][0]
         FP=cm[0][1]
         print('Testing Accuracy:',(TP+TN)/(TP+TN+FN+FP))

         Testing Accuracy: 0.9230769230769231
```

10. Figure: R confusion Matrix



```
             z = 4.85
       p-value = 0.00000125

In [47]: confusionMatrix(heart_test_y, pred)

Confusion Matrix and Statistics

          Reference
Prediction NO YES
       NO  35   6
       YES 17  32

               Accuracy : 0.7444
                 95% CI : (0.6416, 0.8306)
    No Information Rate : 0.5778
    P-Value [Acc > NIR] : 0.0007541

                  Kappa : 0.4959

 Mcnemar's Test P-Value : 0.0370562

            Sensitivity : 0.6731
            Specificity : 0.8421
         Pos Pred Value : 0.8537
         Neg Pred Value : 0.6531
             Prevalence : 0.5778
         Detection Rate : 0.3889
   Detection Prevalence : 0.4556
      Balanced Accuracy : 0.7576

       'Positive' Class : NO
```

11. Figure Logistic Regression



12. Code Python:

```
import pandas as PD
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings

df=pd.read_csv("heart.csv")
df.head(10)

#checkingtheNulValues
df.isnull().sum()

print(df.info())

#checkingtheCorrelationAmongtheAttributes
plt.figure(figsize=(30,15))
sns.heatmap(df.corr(), annot=True, cmap='terrain')
sns.pairplot(data=df)
df.hist(figsize=(12,12), layout=(5,3));
#box and whiskers plot
df.plot(kind='box', subplots=True, layout=(5,3), figsize=(12,12))
plt.show()
#visualizethefeatures and their relation with target(heart disease or no heart disease)
sns.catplot(data=df, x='sex', y='age', hue='target', palette='husl')
df['sex'].value_counts() #207males and 96females
df['cp'].value_counts() #chestPainType
sns.countplot(x='cp', hue='target', data=df, palette='rocket')
#crossTables
gen=pd.crosstab(df['sex'], df['target'])
print(gen)
gen.plot(kind='bar', stacked=True, color=['Skyblue', 'yellow'], grid=False)
chest_pain=pd.crosstab(df['cp'], df['target'])
chest_pain
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
StandardScaler=StandardScaler()
columns_to_scale=['age', 'trestbps', 'chol', 'thalach', 'oldpeak']
df[columns_to_scale]= StandardScaler.fit_transform(df[columns_to_scale])
df.head()
x=df.drop(['target'], axis=1)
y=df['target']
x_train, x_test, y_train, y_test= train_test_split(x,y, test_size=0.3, random_state=40)
print('x_train-', x_train.size)
print('x_test-', x_test.size)
print('y_train-', y_train.size)
print('y_test-', y_test.size)

#appyingLogisticRegression

from sklearn.linear_model import LogisticRegression
lr=LogisticRegression()
modell=lr.fit(x_train, y_train)
```

```
prediction1=modell.predict(x_test)

from sklearn.metrics import confusion_matrix
cm=confusion_matrix(y_test, prediction1)
cm

TP=cm[0][0]
TN=cm[1][1]
FN=cm[1][0]
FP=cm[0][1]
print('Testing Accuracy:', (TP+TN)/(TP+TN+FN+FP))
sns.heatmap(cm, annot=True, cmap="BuPu")
TP=cm[0][0]
TN=cm[1][1]
FN=cm[1][0]
FP=cm[0][1]
print('Testing Accuracy: '. (TP+TN)/(TP+TN+FN+FP))
from sklearn.metrics import accuracy_score
accuracy_score(y_test, prediction1)
from sklearn.metrics import classification_report
print(classification_report(y_test, prediction1))
```

13. Code R

```
data <- read.csv("heart.csv")
head(data)
str(data)
summary(data)
colnames(data)
library(tidyverse)
library(readr)
library(ROCR)
library(PerformanceAnalytics)
library(e1071)
library(caret)
library(gbm)
library(corrplot)
library(ggcorrplot)
library(MASS)
library(rpart)
library(caTools)
library(naivebayes)
library(class)
library(ISLR)
library(glmnet)
library(Hmisc)
library(funModeling)
library(pROC)
library(randomForest)
library(klaR)
library(scales)
library(cluster)
library(factoextra)
library(DataExplorer)
library(ClustOfVar)
library(GGally)
library(ggplot2)
library(plotly)

heart <- data %>%
  mutate(sex = if_else(sex == 1, "MALE", "FEMALE"),
         fbs = if_else(fbs == 1, ">120", "<=120"),
         exang = if_else(exang == 1, "YES" ,"NO"),
         cp = if_else(cp == 1, "ATYPICAL ANGINA",
                      if_else(cp == 2, "NON-ANGINAL PAIN", "ASYMPTOMATIC")),
         restecg = if_else(restecg == 0, "NORMAL",
                           if_else(restecg == 1, "ABNORMALITY", "PROBABLE OR DEFINITE")),
         slope = as.factor(slope),
         ca = as.factor(ca),
         thal = as.factor(thal),
         target = if_else(target == 1, "YES", "NO")
         ) %>%
  mutate_if(is.character, as.factor) %>%
  dplyr::select(target, sex, fbs, exang, cp, restecg, slope, ca, thal, everything())
colnames(data)
colnames(heart)

head(data)
head(heart)
```

```
summary(heart)


boxplot(heart)
boxplot(heart[,10:13])
library(gridExtra)
box_plot <- grid.arrange(ggplot(heart, aes(age, age))+geom_boxplot(),
          ggplot(heart, aes(trestbps, trestbps))+geom_boxplot(),
          ggplot(heart, aes(chol, chol))+geom_boxplot(),
          ggplot(heart, aes(thalach, thalach))+geom_boxplot()
)
box_plot


bar_graph <- grid.arrange(ggplot(heart, aes(x = sex, fill = target))+geom_bar(position = "fill"),
               ggplot(heart, aes(x = fbs, fill = target))+geom_bar(position = "fill"),
               ggplot(heart, aes(x = exang, fill = target))+geom_bar(position = "fill")
               )
bar_graph


grid_bar <- grid.arrange(ggplot(heart, aes(x = cp, fill = target))+geom_bar(position = "fill")+ theme(axis.text.x = element_text(angle = 90, hjust = 1)),
          ggplot(heart, aes(x = restecg, fill = target))+geom_bar(position = "fill")+ theme(axis.text.x = element_text(angle = 90, hjust = 1))
          )
grid_bar


bar_target <- ggplot(heart, aes(target, fill = target))+geom_bar()+theme_classic()+scale_color_brewer(palette = "Accent")+scale_fill_brewer(palette = "Accent")+theme(plot.background = element_rect(fill = "grey97"))+labs(title = "Bar graph of Target variable", x = "Heart Disease", y = "Count")
bar_target


ggplotly(bar_target)
bar_cp <- ggplot(heart, aes(cp, fill = cp))+geom_bar()+theme_classic()+scale_color_brewer(palette = "Accent")+scale_fill_brewer(palette = "Accent")+theme(plot.background = element_rect(fill = "grey97"))+labs(title = "Bar graph of Chest Pain variable", x = "Chest Pain", y = "Count")
bar_cp
ggplotly(bar_cp)
hist_age <- ggplot(heart, aes(age, fill = sex))+geom_histogram(bins = 30)+theme_classic()+scale_color_brewer(palette = "Accent")+scale_fill_brewer(palette = "Accent")+theme(plot.background = element_rect(fill = "grey97"))+labs(title = "Histogram of age variable with sex", x = "age", y = "count")
hist_age
ggplotly(hist_age)
age_point <- ggplot(heart, aes(age, chol, color = sex, size = chol))+geom_point()+geom_smooth()+theme_classic()+theme(plot.background = element_rect(fill = "grey97"))+ggtitle("Age by Chol")
age_point
ggplotly(age_point)
bp_box <- ggplot(heart, aes(x=sex,y=trestbps))+geom_boxplot(fill = "pink")+facet_grid(~cp)+geom_smooth()+theme_classic()+theme(plot.background = element_rect(fill = "grey97"))+labs(title = "Comparison of Blood pressure across pain type", x = "Sex", y = "Blood Pressure")
bp_box


ggplotly(bp_box)
chol_box <- ggplot(heart, aes(x=sex, y=chol))+geom_boxplot(fill = "turquoise")+facet_grid(~cp)+geom_smooth()+theme_classic()+theme(plot.background = element_rect(fill = "grey97"))+labs(title = "Comparison of Cholestoral across pain type ", x = "Sex", y = "Chol")
chol_box
ggplotly(chol_box)
cor_heart <- cor(heart[,10:14])
cor_heart


ggcorrplot(cor_heart,lab = TRUE)
split <- createDataPartition(heart$target, time = 1, list = FALSE, p = 0.70)


heart_train <- heart[split,]
heart_test <- heart[-split,]
dim(heart)
dim(heart_train)
dim(heart_test)
```

```
heart_test_x <- heart_test %>% dplyr::select(-target)
heart_test_y <- heart_test$target
head(heart_train)
head(heart_test_x)
plot_num(heart)
freq(heart)
heart_mod <- glm(target~., data = heart_train, family = "binomial")
summary(heart_mod)
heart_mod
options(scipen = 999)
summary(heart_mod)
test_pred <- predict(heart_mod, type = "response", newdata = heart_test_x)
head(test_pred)
head(heart_test_y)
library(pROC)
roc <- roc(heart_test_y ~ test_pred, plot = TRUE, print.auc = TRUE,  col="red")
roc
pred <- ifelse(test_pred >= 0.8218, 'YES', 'NO')
head(pred)
head(heart_test_y)
str(heart_test_y)
str(pred)
heart_test_y <- as.factor(heart_test_y)
pred <- as.factor(pred)
library(irr)
kappa2(data.frame(heart_test_y, pred))
confusionMatrix(heart_test_y, pred)
```

VI. Acknowledgment

I have completed this task under the mentorship of HM Mostafizur Rahman (Lecturer and department head) school of IT, Bac International Study Centre. I am doing a Bachelor's in computer science. In this fall semester, I have completed the course Data science for business, where I have learned the fundamental of data science, machine learning algorithms from my mentors as course instructors. This work has been assigned as a project assignment. I would like to express my special thanks to my mentor for inspiring us to complete the working paper. Without his active guidance, help, cooperation & encouragement, I would not have been able to write this paper. I am very thankful for his guidance and help in the completion of this paper.

Thanking You.

VII. References

[1] . Mozaffarian, D., Benjamin, E., Go, A., Arnett, D., Blaha, M.Cushman, M. et al. (2015). Heart Disease and Stroke Statistics—2015, Update. Circulation, 131(4). DOI: 10.1161/cir.0000000000000152

[2] Thomas, J.& Princy, R. (2016). “Human heart disease prediction system using Machine Learning techniques”. 1-5. 10.1109/ICCPCT.2016.7530265.

[3] Larose, Chantal & Larose, Daniel. (2019). THE BASICS OF PYTHON AND R. 9-27. 10.1002/9781119526865.ch2.

[4] Larose, Chantal & Larose, Daniel. (2019). Data Science Using Python and R.

[5] Kumar, Anil & Upadhyay, Priyadarshi & Kumar, A.. (2020). Machine Learning. 10.1201/9780429340369-1.

[6] Abdi, Asad. (2016). Three types of Machine Learning Algorithms. 10.13140/RG.2.2.26209.10088.

[7] Möller, Dietmar. (2020). Machine Learning and Deep Learning. 10.1007/978-3-030-60570-4_5.

[8] Ketchakmadze, Ivane & Durglishvili, Nino & Ketchakmadze, Dimitri. (2020). Logistic regression model. 10.13140/RG.2.2.24871.37284.

[9] Bisong, Ekaba. (2019). Logistic Regression. 10.1007/978-1-4842-4470-8_20.

[10] Maalouf, Maher. (2011). Logistic regression in data analysis: An overview. International Journal of Data Analysis Techniques and Strategies. 3. 281-299. 10.1504/IJDATS.2011.041335.

[11] K. Grover, "Advantages and Disadvantages of Logistic Regression", *OpenGenus IQ: Learn Computer Science*, 2019. [Online]. Available:

https://iq.opengenus.org/advantages-and-disadvantages-of-logistic-regression/. [Accessed: 30- Dec- 2020].

[12] Zhou, Hong. (2020). Logistic Regression. 10.1007/978-1-4842-5982-5_6.

[13] R. Cotton, "Python vs. R for Data Science: What’s the Difference?", *Datacamp*, 2020. [Online]. Available: https://www.datacamp.com/community/blog/when-to-use-python-or-r.

[14] Ozgur, Ceyhun & Jha, Sanjeev & Shen, Yiming. (2018). Comparison of R vs. Python for teaching company problems.

[15] Colliau, Taylor & Rogers, Grace & Hughes, Z. & Ozgur, Ceyhun. (2016). ”MatLab vs. Python vs. R”.

[16] Odhiambo, Joab & Onsongo, Winnie & Osman, Shaibu. (2020). An Analytical Comparison Between Python Vs R Programming Languages Which one is the best for Machine Learning and Deep Learning?.

[17] Larose, Chantal & Larose, Daniel. (2019). THE BASICS OF PYTHON AND R. 9-27. 10.1002/9781119526865.ch2.

[18] Zhang, Nailong. (2020). More on R/Python Programming. 10.1201/9781003020646-2.

[19] Rajdhan, Apurb & Agarwal, Avi & Sai, Milan & Ghuli, Poonam. (2020). Heart Disease Prediction using Machine Learning. International Journal of Engineering Research and. V9. 10.17577/IJERTV9IS040614.