A REPORT

ON

Prevent of Heart and Blood Vessel Dysfunctional

SECOND SEMESTER 2023-2024
MTECH (Data Science)
Dissertation Final Report

BY

NAME: HARSH AHUJA ID: 2021FC04297

AT
GE Healthcare
JFWTC, Bangalore

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI

(September 22, 2023)

Dissertation

Prevent of Heart and Blood Vessel Dysfunctional

Submitted in partial fulfilment of the requirements of
MTech Data Science Program

By
Harsh Ahuja

Under the supervision of
Mohammed Shoeb
Staff Software Engineer

Dissertation work carried out at
GE HEALTHCARE, Bangalore

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI

(September 22, 2023)

Acknowledgements

I would like to thank my supervisor **Mohammed Shoeb** for his guidance and support. His comment helped me enhance the work. I would like to express my sincere gratitude towards the project for the support they provide so far.

Special Thanks to my examiner **Vinaya Sathyanarayana** for their support and guidance.

I am also thankful to GE Healthcare for giving me this opportunity and also making me available for all the resources which is required. I am also thankful to Bits Pilani for organizing this course.

My Sincere regards and Thanks to my examiner for the guidance throughout the dissertation.

BIRLA INSTITUTE OF TECHNOLOGY &
SCIENCE, PILANISECOND SEMESTER 2022-23

DSECLZG628T **DISSERTATION**

Dissertation Title : _Prevention of Heart and Blood Dysfunctional using ML Algorithm_

Name of Supervisor : _Mohammed Shoeb_

Name of Student : Harsh Ahuja

ID No. of Student : _2021FC04297_

Abstract

Health is a crucial part of everyone's life. However, owing to multiple reasons like unhealthy lifestyles, work stress, psychological strain, and external factors such as pollution, hazardous work environment, and lack of proper health services, millions of people worldwide fall prey to chronic ailments which affect both the heart and blood vessels, resulting in death or disability. The goal is to enhance heart and blood vessels risk prediction accuracy in population primary care at large.

The system serves as a tool for the health professionals allow them to diagnose at the early stage leading to patient improved outcomes which initially collect the data set and preprocessed the data with cleansing and internally apply the algorithm for the data set.

The solution being developed will be useful to enhance the accuracy of the heart disease and creating the importance in the treatment planning and the medical professionals and also help the patient for the applicable of the pacemaker spikes.

Key Words: Heart and Blood Vessels prediction, Device Drivers data, Models and machine learning algorithm

BIRLA INSTITUTE OF TECHNOLOGY AND
SCIENCE, PILANI

II SEMESTER 22-23

CERTIFACTE

This is to certify that dissertation entitled Prevention of Heart and Blood Vessel Dysfunctional using ML Algo is carried and submitted by Harsh Ahuja, BITS ID NO 2021FC04297 is completed based on the requirement and the scope of the work under my supervision.

(Signature of Supervisor)
Date: 24-09-2023

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI

II SEMESTER 22-23

DSE CL ZG628T **DISSERTATION**

Dissertation FINAL REPORT

BITS ID No. 2021FC04297 Name of Student: Harsh Ahuja

Name of Supervisor: Mohammed Shoeb

Designation of Supervisor: Staff Software Engineer

Qualification and Experience: Mtech in Data Science with 14 years of Experience

E- mail ID of Supervisor: mohammed.shoeb@ge.com

Topic of Dissertation: Prevention of Heart and Blood Vessel Dysfunctional using ML Algo

Name of First Examiner: Vinaya Sathyanarayana

Designation of First Examiner:____

Qualification and Experience: ____

E- mail ID of First Examiner:____

Name of Second Examiner: ____

Designation of Second Examiner:____

Qualification and Experience:____

E- mail ID of Second Examiner:____

(Signature of Student)
Date:22-09-2023

(Signature of Supervisor)
Date: 24-09-2023

Table of Content

1. Introduction
2. Data
3. Data Collection
4. Preprocessing of Data
5. Libraries
6. Machine Learning Technique
7. Conclusion
8. Reference

9. Introduction

Problem Statement

Heart disease is a leading cause of death worldwide due which there is a lack of system which can be tested for the occurring of the heart and blood vessel Dysfunctional. Therefore, there is a need of the system which evaluate and process the medical parameter using the machine learning algorithm.

There are several guidelines for the evaluation and management of Blood Vessel using multiple models as a means of identifying patient which has high risk and under clinical decision making.

Since Machine learning is a very extreme predictors and adopt the complex interaction and non linear linkage that exist between result and the variable.

Since Dysfunctional is a major part for the increase of death and in medical information that can be used to understand the symptoms of various diseases

The main understanding of this research is

1. To extract the 14 attributes or characteristic from the dataset
2. Preprocessing of the data and train the datasets
3. Helps to construct a model
4. Finally, with the achieve result over several metrics analyses and models

10. Data

In this study, we have used a continuous monitoring data providing 24 hour of continuous beat data over 8 months old patient data.
Patient characteristic was recorded at the start of monitoring like age, weight and height.

There are other parameter attributes which was captured at the time of monitoring system. During the test, the equivalent devices are connected to monitor the patient and each heart rate time series have been observed and recorded the data.

11. Data Collection

Collection of the dataset, we split the dataset into training data and testing data. The training dataset is used for prediction model learning and testing data is used for evaluating the prediction model. For this project, 70% of training data is used and 30% of data is used for testing.

The dataset consists of 76 attributes; out of which, 14 attributes are used for the system.

1. Age: in years

2. Sex: (1 = male, 0= female)

3. CP: pain type in chest

4. Trestbps: blood pressure at the time of resting

5. Cholesterol: serum cholesterol

6. Fbs: fasting blood sugar

7. Restecg: Resting ECG

8. Thalach: Max Heart Rate

9. Exang: Exercise angina

10. Oldpeak: ST Depression

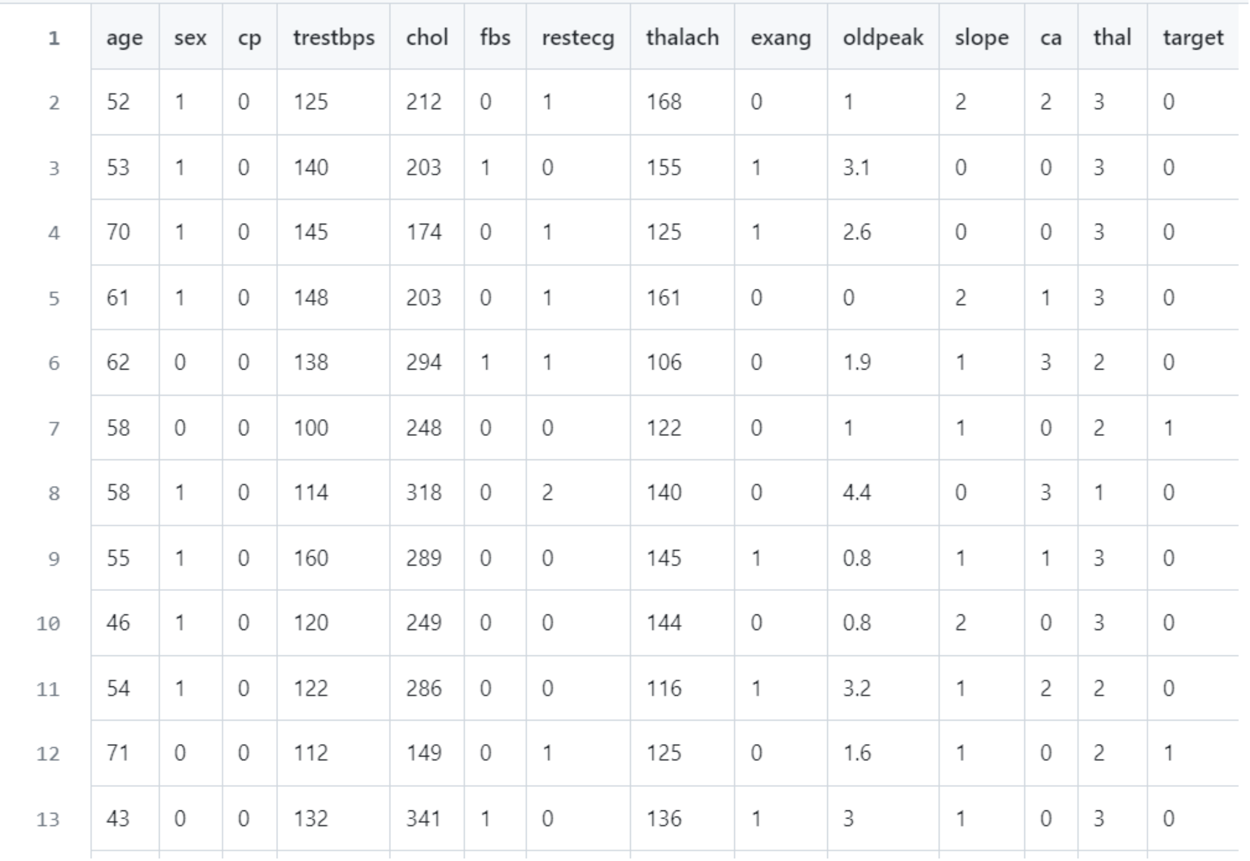
11. Slope: Peak at ST

12. Ca: Major Vessel

13. Thal: Reverse Defect

14. Target: Disease or not having disease

Snippet of CSV



1	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
2	52	1	0	125	212	0	1	168	0	1	2	2	3	0
3	53	1	0	140	203	1	0	155	1	3.1	0	0	3	0
4	70	1	0	145	174	0	1	125	1	2.6	0	0	3	0
5	61	1	0	148	203	0	1	161	0	0	2	1	3	0
6	62	0	0	138	294	1	1	106	0	1.9	1	3	2	0
7	58	0	0	100	248	0	0	122	0	1	1	0	2	1
8	58	1	0	114	318	0	2	140	0	4.4	0	3	1	0
9	55	1	0	160	289	0	0	145	1	0.8	1	1	3	0
10	46	1	0	120	249	0	0	144	0	0.8	2	0	3	0
11	54	1	0	122	286	0	0	116	1	3.2	1	2	2	0
12	71	0	0	112	149	0	1	125	0	1.6	1	0	2	1
13	43	0	0	132	341	1	0	136	1	3	1	0	3	0

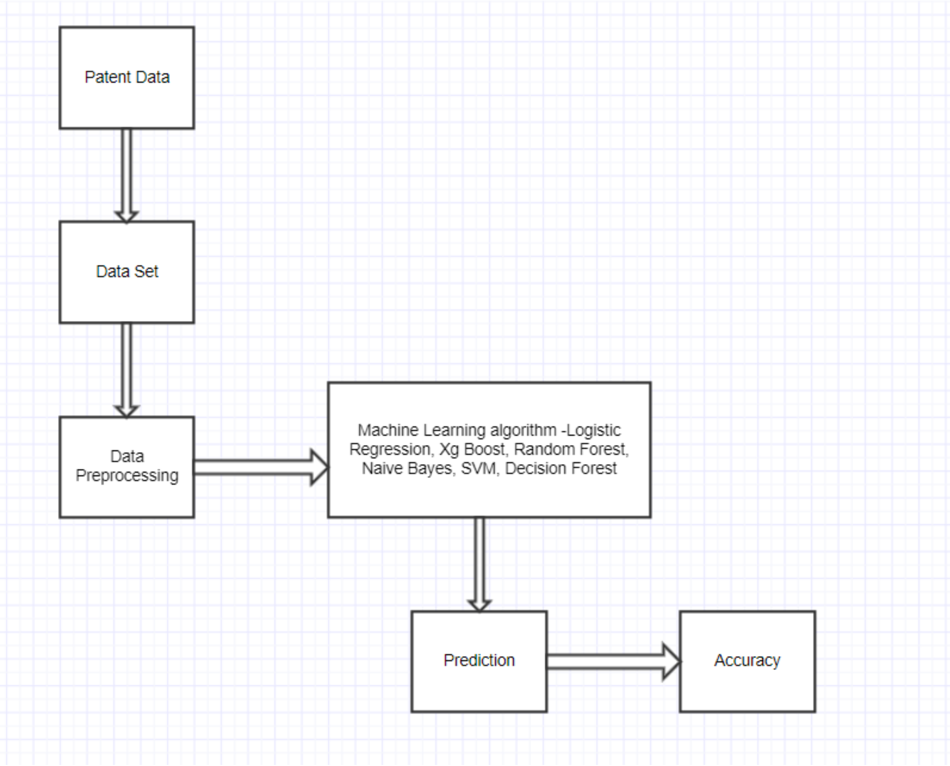
12. Pre-processing of Data

In pre-processing of data, we transform data into our required format. It is used to deal with noises, duplicates, and missing values of the dataset.

In Over Sampling, dataset balance is done by increasing the size of the scarce samples. This process is considered when the amount of data is inadequate.

Difference (t) = observation (t) – observation (t - 1)

Comparative analysis is performed among algorithms and the algorithm that gives the highest accuracy is used for heart disease prediction.


Patent Data
Data Set
Data Preprocessing
Machine Learning algorithm -Logistic Regression, Xg Boost, Random Forest, Naive Bayes, SVM, Decision Forest
Prediction
Accuracy

Tools/ Tech – Python, Sklearn, Numpy, Librosa, Matplotlib, Seaborn, SciPy

Ploting the heapmap and accuracy projection via algorithm

- Uses a subset of training points in the decision function
- Document classification tasks
- Building a tree like structure in the alogorithm
- Defining the accuracy model

13.Import Libraries

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split, StratifiedKFold, cross_val_score
from sklearn.pipeline import make_pipeline, Pipeline from sklearn.model_selection
import GridSearchCV
from sklearn.svm import SVC
from sklearn.naive_bayes import MultinomialNB
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.externals import joblib
from sklearn.metrics import make_scorer, f1_score, recall_score, precision_score
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score
from sklearn.metrics import log_loss
import warnings
```

14.Split the Datasets

```
from sklearn.model_selection import train_test_split
```

15.Machine Learning Technique

Model learns the association function between pairs of input and output sequence denoted by X and Y respectively to make the prediction. Therefore to make the prediction from the time series data sets, we reframe the time series into supervised learning problem by converting the sequenced HR time series observation window and target window

Observation from last step (t-1) as Input

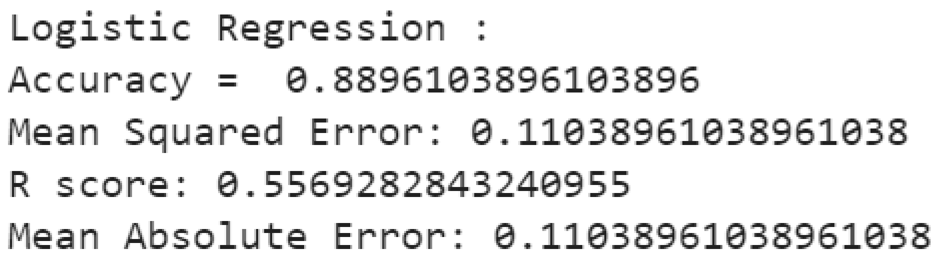
Observation at the current time step (t) as Output

Algorithm logic – SVM, Naive Bayes, Decision Tree, Random Forest, Logistic Regression, Adaboost Algorithm

a. Logistic Regression

It describe data and explain relationship between dependent binary variable and 1 or more nominal or ratio level independent variable.

```
from sklearn.linear_model import LogisticRegression
classifier = LogisticRegression()
classifier.fit(xTrain,yTrain)
yPred = classifier.predict(xTest)
mse = mean_squared_error(yTest,yPred)
r = r2_score(yTest,yPred)
mae = mean_absolute_error(yTest,yPred)
accuracy = accuracy_score(yTest,yPred)
print("Logistic Regression :")
print("Accuracy = ", accuracy)
print("Mean Squared Error:",mse)
print("R score:",r)
print("Mean Absolute Error:",mae)
```

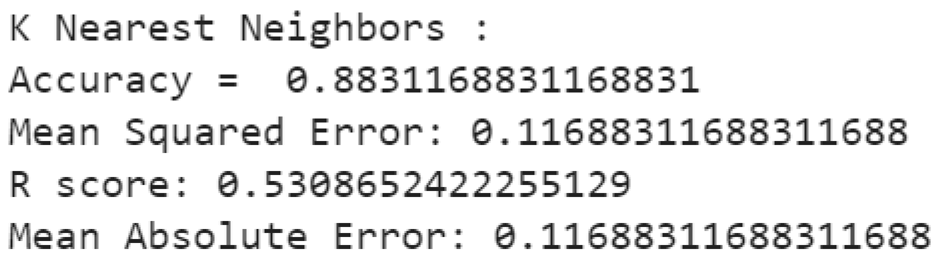


```
Logistic Regression :
Accuracy =  0.8896103896103896
Mean Squared Error: 0.11038961038961038
R score: 0.5569282843240955
Mean Absolute Error: 0.11038961038961038
```

b. K Nearest Neighbors:

It’s a non parametric method used for classification and regression. In Both cases, the input consist of closest training examples

```
from sklearn.neighbors import KNeighborsClassifier
classifier = KNeighborsClassifier(n_neighbors=5, p=2, metric='minkowski')
classifier.fit(XTrain,yTrain)
yPred = classifier.predict(XTest)
mse = mean_squared_error(yTest,yPred)
r = r2_score(yTest,yPred)
mae = mean_absolute_error(yTest,yPred)
accuracy = accuracy_score(yTest,yPred)
print("K Nearest Neighbors :")
print("Accuracy = ", accuracy)
print("Mean Squared Error:",mse)
print("R score:",r)
print("Mean Absolute Error:",mae)
```

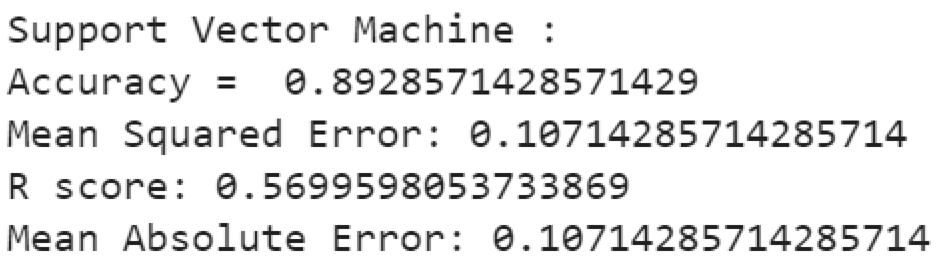


```
K Nearest Neighbors :
Accuracy =  0.8831168831168831
Mean Squared Error: 0.11688311688311688
R score: 0.5308652422255129
Mean Absolute Error: 0.11688311688311688
```

c. Support Vector Machine

It can be used for classification or regression problem. Uses a technique called Kernel Trick to transform the data and then based on these transformation find optimal boundary for the outputs.

```
from sklearn.svm import SVC
classifier = SVC(kernel='linear',random_state=0)
classifier.fit(XTrain,yTrain)
yPred = classifier.predict(XTest)
mse = mean_squared_error(yTest,yPred)
r = r2_score(yTest,yPred)
mae = mean_absolute_error(yTest,yPred)
accuracy = accuracy_score(yTest,yPred)
print("Support Vector Machine :")
print("Accuracy = ", accuracy)
print("Mean Squared Error:",mse)
print("R score:",r)
print("Mean Absolute Error:",mae)
```



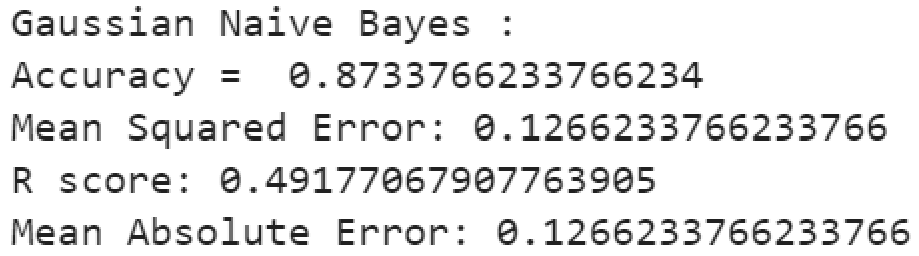
```
Support Vector Machine :
Accuracy =  0.8928571428571429
Mean Squared Error: 0.10714285714285714
R score: 0.5699598053733869
Mean Absolute Error: 0.10714285714285714
```

d. GNB (Gaussian Naïve Bayes)

It specifically used to have continous values and assume all feature are following the model ie normal distribution.

```
from sklearn.naive_bayes import GaussianNB
```

```
classifier = GaussianNB()
classifier.fit(XTrain,yTrain)
yPred = classifier.predict(XTest)
mse = mean_squared_error(yTest,yPred)
r = r2_score(yTest,yPred)
mae = mean_absolute_error(yTest,yPred)
accuracy = accuracy_score(yTest,yPred)
print("Gaussian Naive Bayes :")
print("Accuracy = ", accuracy)
print("Mean Squared Error:",mse)
print("R score:",r)
print("Mean Absolute Error:",mae)
```

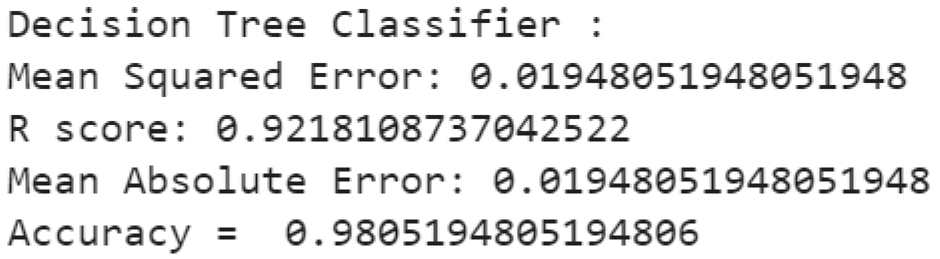


```
Gaussian Naive Bayes :
Accuracy =  0.8733766233766234
Mean Squared Error: 0.1266233766233766
R score: 0.49177067907763905
Mean Absolute Error: 0.1266233766233766
```

e. Decision Tree Classifier

This method used for classification and regression. The goal is to create a model that predicts value of target variable by simple decision rule from data features.

```
from sklearn.tree import DecisionTreeClassifier as DT
classifier = DT(criterion='entropy', random_state=0)
classifier.fit(XTrain,yTrain)
yPred = classifier.predict(XTest)
mse = mean_squared_error(yTest,yPred)
r = r2_score(yTest,yPred)
mae = mean_absolute_error(yTest,yPred)
accuracy = accuracy_score(yTest,yPred)
print("Decision Tree Classifier :")
print("Mean Squared Error:",mse)
print("R score:",r)
print("Mean Absolute Error:",mae)
print("Accuracy = ", accuracy)
```

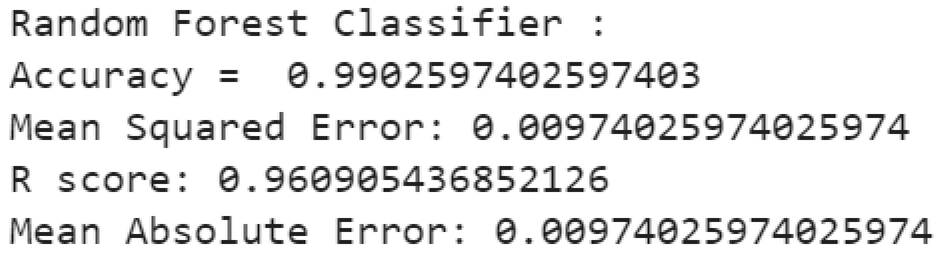


```
Decision Tree Classifier :
Mean Squared Error: 0.01948051948051948
R score: 0.9218108737042522
Mean Absolute Error: 0.01948051948051948
Accuracy =  0.9805194805194806
```

f. Random Forest Classifier

It consist large number of decision tree operate as ensemble. Each individual tree in rainforest spits out a class prediction and the class with the most votes become our model prediction.

```
from sklearn.ensemble import RandomForestClassifier as RF
classifier = RF(n_estimators=10, criterion='entropy', random_state=0)
classifier.fit(XTrain,yTrain)
yPred = classifier.predict(XTest)
mse = mean_squared_error(yTest,yPred)
r = r2_score(yTest,yPred)
mae = mean_absolute_error(yTest,yPred)
accuracy = accuracy_score(yTest,yPred)
print("Random Forest Classifier :")
print("Accuracy = ", accuracy)
print("Mean Squared Error:",mse)
print("R score:",r)
print("Mean Absolute Error:",mae)
```

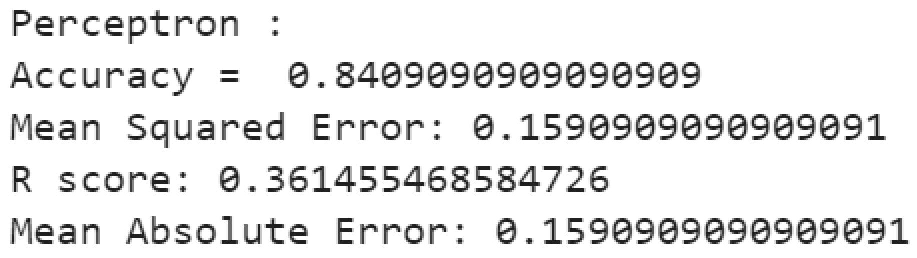


```
Random Forest Classifier :
Accuracy =  0.9902597402597403
Mean Squared Error: 0.00974025974025974
R score: 0.960905436852126
Mean Absolute Error: 0.00974025974025974
```

g. Perceptron

It’s an algorithm for supervised learning of binary classifiers. It functions which decide whether or not an input represented by vector of numbers and specific class

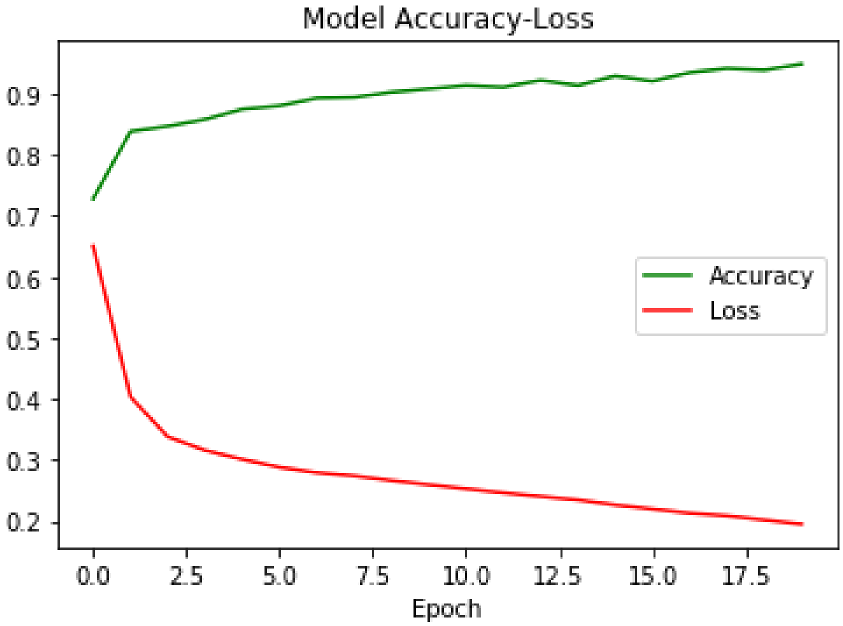
```
from sklearn.linear_model import Perceptron
classifier = Perceptron(tol=1e-3, random_state=0)
classifier.fit(XTrain,yTrain)
yPred = classifier.predict(XTest)
mse = mean_squared_error(yTest,yPred)
r = r2_score(yTest,yPred)
mae = mean_absolute_error(yTest,yPred)
accuracy = accuracy_score(yTest,yPred)
print("Perceptron :")
print("Accuracy = ", accuracy)
print("Mean Squared Error:",mse)
print("R score:",r)
print("Mean Absolute Error:",mae)
```



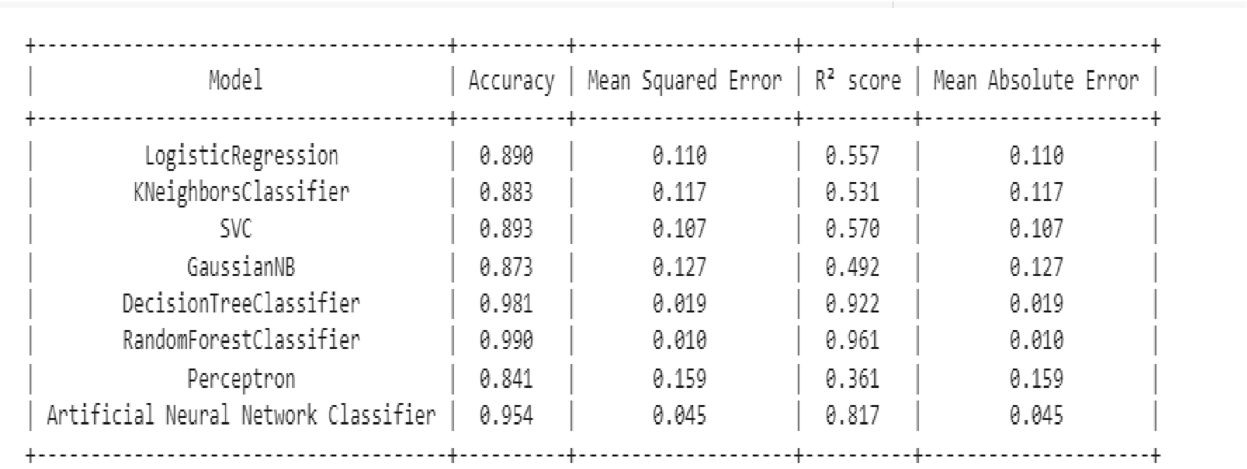
```
Perceptron :
Accuracy =  0.8409090909090909
Mean Squared Error: 0.1590909090909091
R score: 0.361455468584726
Mean Absolute Error: 0.1590909090909091
```

h. Artificial Neural Network

Its an interconnected group of node and represent an neuron and also represent connection from the output of one neuron to another neuron.



Artificial Neural Network Classifier :
Accuracy = 0.9642857142857143
Mean Squared Error: 0.03571428571428571
R score: 0.8566532684577957
Mean Absolute Error: 0.03571428571428571



```
+-----------------------------------+----------+------------------+----------+--------------------+
|               Model               | Accuracy | Mean Squared Error | R² score | Mean Absolute Error |
+-----------------------------------+----------+------------------+----------+--------------------+
|         LogisticRegression        |  0.890   |      0.110       |  0.557   |       0.110        |
|        KNeighborsClassifier       |  0.883   |      0.117       |  0.531   |       0.117        |
|                SVC                |  0.893   |      0.107       |  0.570   |       0.107        |
|             GaussianNB            |  0.873   |      0.127       |  0.492   |       0.127        |
|       DecisionTreeClassifier      |  0.981   |      0.019       |  0.922   |       0.019        |
|       RandomForestClassifier      |  0.990   |      0.010       |  0.961   |       0.010        |
|             Perceptron            |  0.841   |      0.159       |  0.361   |       0.159        |
| Artificial Neural Network Classifier |  0.954   |      0.045       |  0.817   |       0.045        |
+-----------------------------------+----------+------------------+----------+--------------------+
```

16. Conclusion

Result of this study show that data provided are sufficient and can be explored using data analytics to predict future HR

Our result show that Random Forest Classifier works best for our datasets

Literature Survey

Multiple data source from the medical history of different age group is taken across 8 months duration that helps us to diagnosed any blood vessel dysfunctional in the heart. The attribute thalasammel is used for the dysfunctional of the heart and used different techniques for the machine learning model.

The accuracy among the model was achieved. We had a discussion with Dr Ishani for the event diagnosis and coronary failure to congestive and perform the experiment on this one.

Random Forest selection model uses for diagnosing the disease. This learning system can help to improve the quality of blood vessel dysfunctional detection.

17. References

WHO. Disease https://www.who.int/en/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds)

Chest and Heart Stroke Understanding from the series.

Calculate Precision, Recall, and F-Measure for Imbalanced Classification. 2020