MACHINE LEARNING LAB

MINI PROJECT
ON

HEART DISEASE PREDICTION

Submitted in partial fulfilment of the requirements of the degree
BACHELOR OF ENGINEERING IN COMPUTER ENGINEERING

By

Pooja Bhagat (BEA-110)

Priyanka Korde (BEA-113)

Raghuwardayal Maurya (BEA-154)

Asmit Patil (BEA-131)

Under the guidance of

Prof. Pranjali Gurnule

LOKMANYA TILAK COLLEGE OF ENGINEERING

Department of Computer

Engineering **YEAR 2022 – 2023**

CERTIFICATE

This is to certify that the project entitled **"Heart Disease Prediction"** is a bonafide work of **Pooja Bhagat (BEA-110), Priyanka Korde (BEA-113), Raghuwardayal Maurya (BEA-154), Asmit Patil (BEA-131)** Submitted to **Prof. Pranjali Gurnule** in partial fulfilment of the requirement for the course of Machine Learning Lab.

(Prof. Pranjali Gurnule)

ACKNOWLEDGEMENT

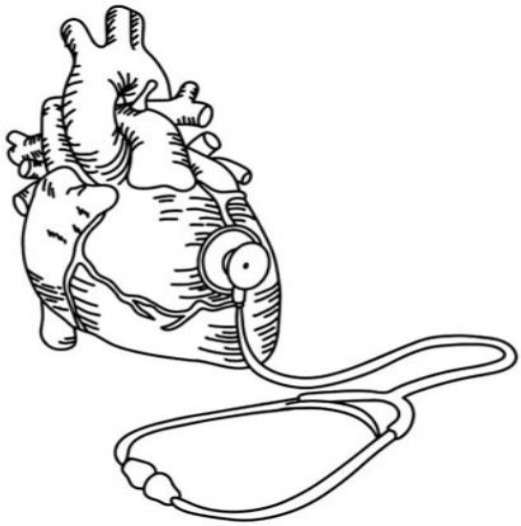
It gives us immense pleasure to express our deepest sense of gratitude and sincere thanks to my highly respected and esteemed guide **Prof. Pranjali Gurnule,** Computer Department for her valuable guidance, encouragement and help for completing this work. Her useful suggestions for this whole work and co-operative behaviour are sincerely acknowledged.

We also wish to express my gratitude to Prof. **R.D.Gawali** (Head – Computer Engineering) for his kind hearted support.

Students Signature

TABLE OF CONTENTS

INTRODUCTION



Human heart is the principal part of the human body. Basically, it regulates blood flow throughout our body. Any irregularity to heart can cause distress in other parts of body. Any sort of disturbance to normal functioning of the heart can be classified as a Heart disease. In today's contemporary world, heart disease is one of the primary reasons for occurrence of most deaths. Heart disease may occur due to unhealthy lifestyle, smoking, alcohol and high intake of fat which may cause hypertension. According to the World Health Organization more than 10 million die due to Heart diseases every single year around the world. A healthy lifestyle and earliest detection are only ways to prevent the heart related diseases.

The proposed work makes an attempt to detect these heart diseases at early stage to avoid disastrous consequences. Records of large set of medical data created by medical experts are available for analysing and extracting valuable knowledge from it. Machine Learning (ML) which is subfield of data mining handles large scale well-formatted dataset efficiently. In the medical field, machine learning can be used for diagnosis, detection and prediction of various diseases. The main goal is to provide a tool for doctors to detect heart disease as early stage. This in turn will help to provide effective treatment to patients and avoid severe consequences. ML plays a very important role to detect the hidden discrete patterns and thereby analyse the given data. After analysis of data ML techniques help in heart disease prediction and early diagnosis. This project presents performance analysis of various ML techniques such as k nearest neighbours Algorithm (KNN), Logistic Regression and Random Forest for predicting heart disease at an early stage.

PROBLEM STATEMENT

Heart disease prediction using machine learning is one of the most challenging tasks. The shortage of specialists and may be wrongly diagnosed cases have necessitated the need to develop a fast and efficient detection system.

MOTIVATION

The main motivation of doing this project is to present a heart disease prediction model for the prediction of occurrence of heart disease. Further, the aim is towards identifying the best classification algorithm for identifying the possibility of heart disease in a patient. This work is justified by performing a comparative study and analysis using three classification algorithms namely k nearest neighbours Algorithm (KNN), Logistic Regression and Random Forest are used at different levels of evaluations. Although these are commonly used machine learning algorithms, the heart disease prediction is a vital task involving highest possible accuracy. Hence, the three algorithms are evaluated at numerous levels and types of evaluation strategies. This will provide researchers and medical practitioners to establish a better.

DESIGN OF SYSTEM

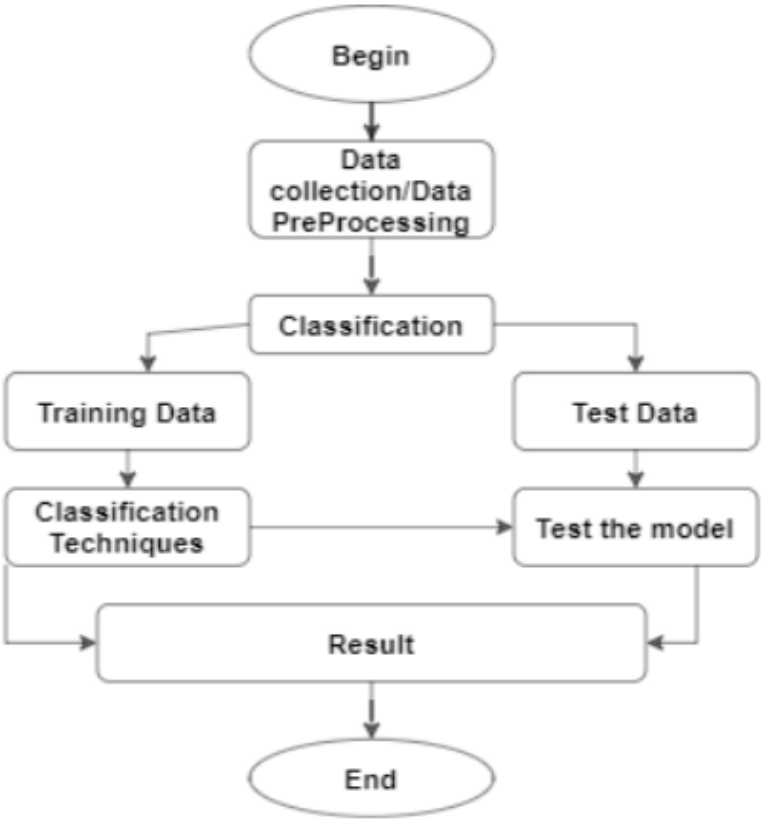


Fig 1: Generic Model Predicting Heart Disease

The proposed work predicts heart disease by exploring the three classification algorithms and does performance analysis. The objective is to effectively predict if the patient suffers from heart disease. The health professional enters the input values from the patient's health report. The data is fed into model which predicts the probability of having heart disease.

A. Data Collection and Preprocessing

The dataset used was the Heart Disease Dataset (Comprehensive) statlog + cleveland + hungary dataset which is taken from three other research datasets used in different research papers. The Nature article listing heart disease database and names of popular datasets used in various heart disease research. The dataset consists of 1190 records of patients from US, UK, Switzerland and Hungary. It has 11 features and 1 target variable. Therefore, we have used the already processed Cleveland dataset available in the Kaggle website for our analysis. The complete description of the attributes used in the proposed work is mentioned in below.

The detailed description of all the features are as follows:

1. Age: Age of the individual in year (Numeric)
2. Sex: displays the gender of the individual using the following format : 1 = male 0 = female (Nominal)
3. Chest-pain type: displays the type of chest-pain experienced by the individual using the following format : 1 = typical angina, 2 = atypical angina, 3 = non — anginal pain, 4 = asymptotic (Nominal)
4. Resting Blood Pressure: displays the resting blood pressure value of an individual in mmHg (unit) (Numeric)
5. Serum Cholestrol: displays the serum cholesterol in mg/dl (unit) (Numeric)
6. Fasting Blood Sugar: compares the fasting blood sugar value of an individual with 120mg/dl. If fasting blood sugar > 120mg/dl then : 1 (true), else : 0 (false) (Nominal)
7. Resting ECG : displays resting electrocardiographic results 0 = normal, 1 = having ST-T wave abnormality, 2 = left ventricular hyperthrophy (Nominal)
8. Max heart rate achieved : displays the max heart rate achieved by an individual. (Numeric)
9. Exercise induced angina (Nominal) : 1 = yes, 0 = no
10. Old peak: ST depression induced by exercise relative to rest (Nominal)
11. Peak exercise ST segment : ST segment measured in terms of slope during peak exercise (Nominal) 0 = Normal, 1 = upsloping, 2 = flat, 3 = downsloping

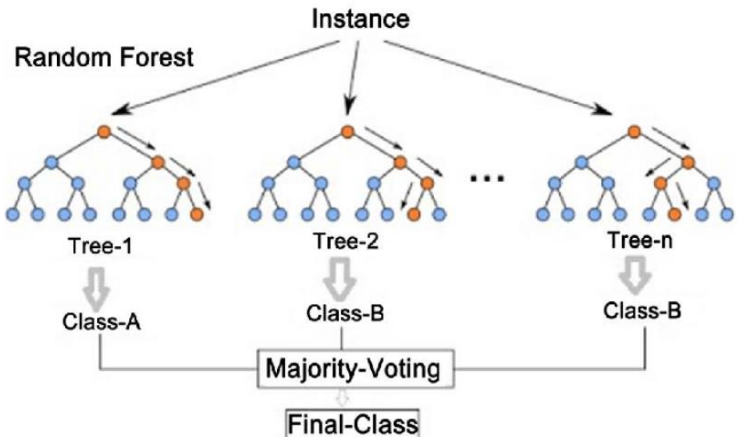
B. Classification

The attributes mentioned above are provided as input to the different ML algorithms such as Random Forest, Logistic Regression, k nearest neighbours Algorithm classification techniques.

i. Random Forest

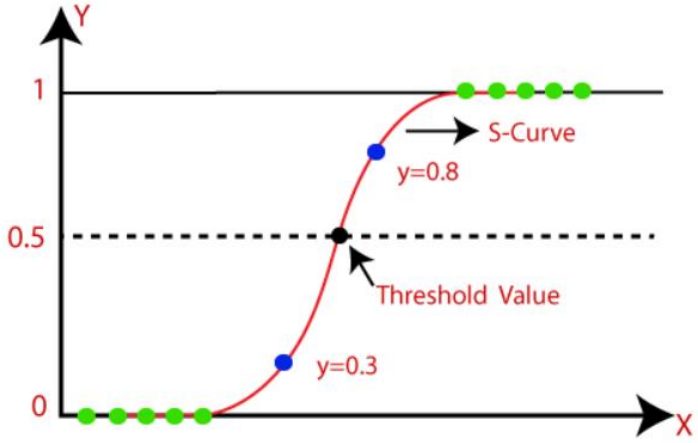
Random Forest algorithms are used for classification as well as regression. It creates a tree for the data and makes prediction based on that. Random Forest algorithm can be used on large datasets and can produce the same result even when large sets record values are missing. The generated samples from the decision tree can be saved so that it can be used on other data. In random forest

there are two stages, firstly create a random forest then make a prediction using a random forest classifier created in the first stage.



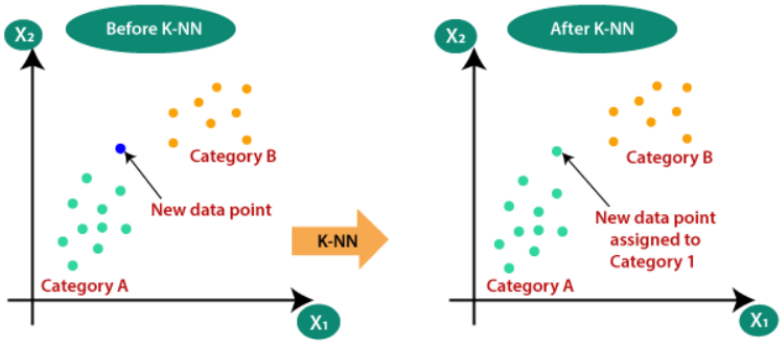
ii. Logistic Regression

Logistic Regression is a classification algorithm mostly used for binary classification problems. In logistic regression instead of fitting a straight line or hyper plane, the logistic regression algorithm uses the logistic function to squeeze the output of a linear equation between 0 and 1. There are 13 independent variables which makes logistic regression good for classification.



iii. K-nearest Neighbour

The k-nearest neighbors (KNN) algorithm is a simple, supervised machine learning algorithm that can be used to solve both classification and regression problems. It's easy to implement and understand, but has a major drawback of becoming significantly slows as the size of that data in use grows. KNN works by finding the distances between a query and all the examples in the data, selecting the specified number examples (K) closest to the query, then votes for the most frequent label (in the case of classification) or averages the labels (in the case of regression).



C. Model Evaluation

In this step we will first define which evaluation metrics we will use to evaluate our model. The most important evaluation metric for this problem domain is sensitivity, specificity, Precision, F1-measure, Geometric mean and mathew correlation coefficient and finally ROC AUC curve

Mathew Correlation coefficient (MCC)

The Matthews correlation coefficient (MCC), instead, is a more reliable statistical rate which produces a high score only if the prediction obtained good results in all of the four confusion matrix categories (true positives, false negatives, true negatives, and false positives), proportionally both to the size of positive elements and the size of negative elements in the dataset.

$$\mathrm{MCC} = \frac{\mathrm{TP} \cdot \mathrm{TN} - \mathrm{FP} \cdot \mathrm{FN}}{\sqrt{(\mathrm{TP} + \mathrm{FP}) \cdot (\mathrm{TP} + \mathrm{FN}) \cdot (\mathrm{TN} + \mathrm{FP}) \cdot (\mathrm{TN} + \mathrm{FN})}}$$

(worst value: −1; best value: +1)

F1 Score

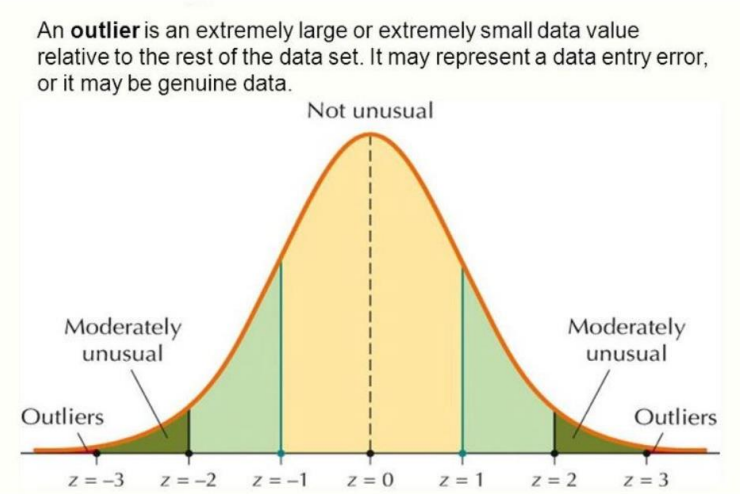
F1 Score is the weighted average of Precision and Recall. Therefore, this score takes both false positives and false negatives into account. Intuitively it is not as easy to understand as accuracy, but F1 is usually more useful than accuracy, especially if you have an uneven class distribution. Accuracy works best if false positives and false negatives have similar cost. If the cost of false positives and false negatives are very different, it's better to look at both Precision and Recall. In our case, F1 score is 0.701.

$$F_1 = 2 \cdot \frac{\mathrm{precision} \cdot \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}} = \frac{2 \cdot \mathrm{TP}}{2 \cdot \mathrm{TP} + \mathrm{FP} + \mathrm{FN}}$$

$$(\text{worst value} = 0; \text{best value} = 1).$$

Log Loss

Logarithmic loss measures the performance of a classification model where the prediction input is a probability value between 0 and 1. The goal of our machine learning models is to minimize this value. A perfect model would have a log loss of 0. Log loss increases as the predicted probability diverges from the actual label. So predicting a probability of .012 when the actual observation label is 1 would be bad and result in a high log loss.



PROJECT CODE

9. Model Building

Random Forest Classifier (criterion='entropy')

```
[46] rf_ent = RandomForestClassifier(criterion='entropy',n_estimators=100)
     rf_ent.fit(X_train, y_train)
     y_pred_rfe = rf_ent.predict(X_test)
```

Random Forest Classifier (criterion="Gini")

```
[47] rf_ent = RandomForestClassifier(criterion='gini',n_estimators=100)
     rf_ent.fit(X_train, y_train)
     y_pred_rfe = rf_ent.predict(X_test)
```

K Nearest negihbor (n = 5)

```
[48] knn = KNeighborsClassifier(5)
     knn.fit(X_train,y_train)
     y_pred_knn = knn.predict(X_test)
```

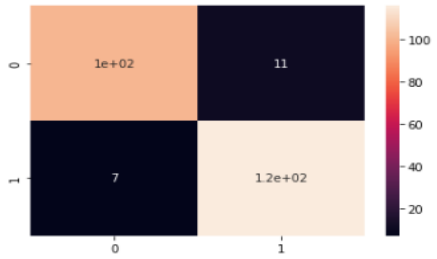
Logistic Regression

```
[49] log_res = LogisticRegression()
     log_res.fit(X_train,y_train)
     y_pred_Log_res = log_res.predict(X_test)
```

Confusion Matrix

```
[50] CM = confusion_matrix(y_test, y_pred_rfe)
     sns.heatmap(CM, annot = True)

     TN = CM[0][0]
     FN = CM[1][0]
     TP = CM[1][1]
     FP = CM[0][1]

     specificity = TN/(TN+FP)
     loss_log = log_loss(y_test, y_pred_rfe)
     acc= accuracy_score(y_test, y_pred_rfe)
     roc=roc_auc_score(y_test, y_pred_rfe)
     prec = precision_score(y_test, y_pred_rfe)
     rec = recall_score(y_test, y_pred_rfe)
     f1 = f1_score(y_test, y_pred_rfe)
     mathew = matthews_corrcoef(y_test, y_pred_rfe)

     model_results =pd.DataFrame([['Random Forest',acc, prec,rec,specificity, f1,roc, loss_log,mathew]],
                  columns = ['Model', 'Accuracy','Precision', 'Sensitivity','Specificity', 'F1 Score','ROC','Log_Loss','mathew_corrcoef'])

     model_results
```

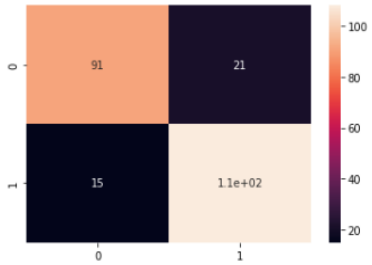
	Model	Accuracy	Precision	Sensitivity	Specificity	F1 Score	ROC	Log_Loss	mathew_corrcoef
0	Random Forest	0.923404	0.913386	0.943089	0.901786	0.928	0.922438	2.645561	0.846721



1.0 × 10-2 = 1E-02 = 0.01. 1.0 × 10-3 = 1E-03 = 0.001 [1e + 02]

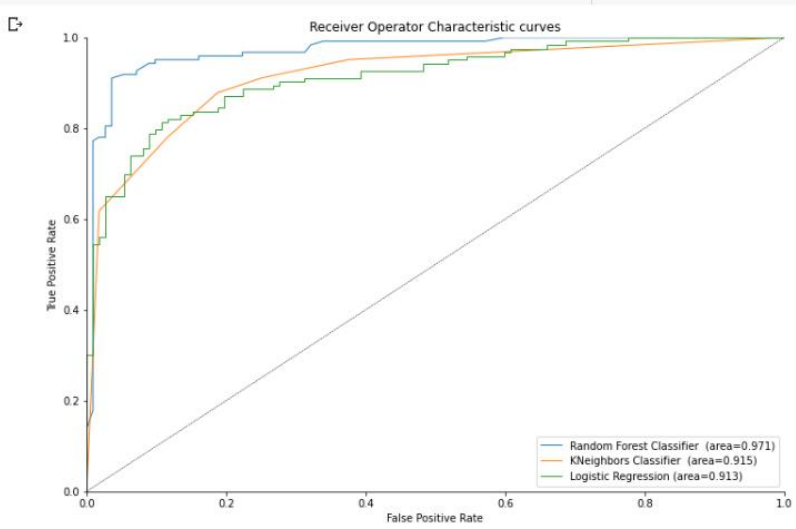
```
[51] CM = confusion_matrix(y_test, y_pred_knn)
     sns.heatmap(CM, annot = True)

     TN = CM[0][0]
     FN = CM[1][0]
     TP = CM[1][1]
     FP = CM[0][1]

     specificity = TN/(TN+FP)
     loss_log = log_loss(y_test, y_pred_knn)
     acc= accuracy_score(y_test, y_pred_knn)
     roc=roc_auc_score(y_test, y_pred_knn)
     prec = precision_score(y_test, y_pred_knn)
     rec = recall_score(y_test, y_pred_knn)
     f1 = f1_score(y_test, y_pred_knn)
     mathew = matthews_corrcoef(y_test, y_pred_knn)

     model_results =pd.DataFrame([['KNeighbors Classifier',acc, prec,rec,specificity, f1,roc, loss_log,mathew]],
                  columns = ['Model', 'Accuracy','Precision', 'Sensitivity','Specificity', 'F1 Score','ROC','Log_Loss','mathew_corrcoef'])

     model_results
```

	Model	Accuracy	Precision	Sensitivity	Specificity	F1 Score	ROC	Log_Loss	mathew_corrcoef
0	KNeighbors Classifier	0.846809	0.837209	0.878049	0.8125	0.857143	0.845274	5.291118	0.69312



ROC AUC Curve

```
def roc_auc_plot(y_true, y_proba, label=' ', l='-', lw=1.0):
    from sklearn.metrics import roc_curve, roc_auc_score
    fpr, tpr, _ = roc_curve(y_true, y_proba[:,1])
    ax.plot(fpr, tpr, linestyle=l, linewidth=lw,
            label="%s (area=%.3f)"%(label,roc_auc_score(y_true, y_proba[:,1])))

f, ax = plt.subplots(figsize=(12,8))


roc_auc_plot(y_test,rf_ent.predict_proba(X_test),label='Random Forest Classifier ',l='-')
roc_auc_plot(y_test,knn.predict_proba(X_test),label='KNeighbors Classifier ',l='-')
roc_auc_plot(y_test,log_res.predict_proba(X_test),label='Logistic Regression',l='-')

ax.plot([0,1], [0,1], color='k', linewidth=0.5, linestyle='--',
        )
ax.legend(loc="lower right")
ax.set_xlabel('False Positive Rate')
ax.set_ylabel('True Positive Rate')
ax.set_xlim([0, 1])
ax.set_ylim([0, 1])
ax.set_title('Receiver Operator Characteristic curves')
sns.despine()
```



As we can see highest average area under the curve (AUC) of 0.973 is attained by Random Forest Classifier

	Model	Accuracy	Precision	Sensitivity	Specificity	F1 Score	ROC	Log_Loss	mathew_corrcoef
0	KNeighbors Classifier	0.846809	0.837209	0.878049	0.812500	0.857143	0.845274	5.291118	0.693120
1	RFE	0.902128	0.884615	0.934959	0.866071	0.909091	0.900515	3.380442	0.804719
2	KNN	0.846809	0.837209	0.878049	0.812500	0.857143	0.845274	5.291118	0.693120
3	Log_res	0.825532	0.830645	0.837398	0.812500	0.834008	0.824949	6.025986	0.650182

RESULT

13. Saving the model

```
[63] from joblib import dump
     dump(rf_ent, 'Heart_model.joblib')
```

```
['Heart_model.joblib']
```

Using the model to perdict label for given features

```
from joblib import dump, load
import numpy as np
model = load('Heart_model.joblib')

# 15 features Model
# features = np.array([[54,150,195,0,122,0,0.0,0,1,0,1,0,0,1,0]])
# features = np.array([[0.591837,0.505495,0.457265,0,0.570370,1,0.522727,1,0,0,0,0,0,1,0]])

# 11 features Model
features = np.array([[0.469388,0.433809,0,0.429630,1,0.376623,1,1,0,0,1]])
features = np.array([[0.612245,0.527495,0,0.466667,1,0.584416,0,1,0,1,0]])

perdiction = model.predict(features)

if (perdiction):
    print("There are Chances of Heart Disease! Consult your Doctor Soon!")
else:
    print('NO fear of Heart Disease But for Better Understanding can Consult your Doctor!')
```

```
There are Chances of Heart Disease! Consult your Doctor Soon!
```

CONCLUSION

With the increasing number of deaths due to heart diseases, it has become mandatory to develop a system to predict heart diseases effectively and accurately. In this project, all the three Machine Learning methods accuracies are compared based on which one prediction model is generated. Accuracy of the algorithms in machine learning depends upon the dataset that used for training and testing purpose. When we perform the analysis of algorithms on the basis of dataset whose attributes are shown above. Hence, the aim is to use various evaluation metrics like confusion matrix, accuracy, precision, recall, and f1-score which predicts the disease efficiently. We have find that the best performing Classifier is **Random Forest Classifier Algorithm** with 90% accuracy. The top 5 most contribution features are: Max heart Rate achieved, Cholestrol, st_depression, exercise_induced_angina, Age. Random Forest algorithm as well as using a larger dataset as compared to the one used in this analysis which will help to provide better results and help health professionals in predicting the heart disease effectively and efficiently.

REFERENCES

1. https://www.ijert.org/research/heart-disease-prediction-using-machine-learning-IJERTV9IS040614.pdf
2. https://www.kaggle.com/datasets/sid321axn/heart-statlog-cleveland-hungary-final
3.