HEART STROCK CARDIOVASCULAR USING MACHINE LEARNING

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Engineering degree in Computer Science and Engineering

By

DUDDUKURI SAI SARATH (Reg.No – 39110288)
ELURU ABHISHEK (Reg.No – 39110296)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SCHOOL OF COMPUTING

SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with Grade "A" by NAAC | 12B Status by UGC | Approved by AICTE
JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI - 600119

APRIL - 2023

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with A‖ grade by NAAC

Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai – 600 119

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the Bonafede work of **ELURU ABHISHEK (Reg No - 39110296) and DUDDUKURI SAI SARATH (REG NO - 39110288)** who carried out the Project Phase-2 entitled **"HEART STROCK CARDIOVASCULAR USING MACHINE LEARNING"** under my supervision from January 2023 to April 2023.

[illegible]

Internal Guide

Dr. A. C. SANTHA SHEELA, M.E., Ph.D.

[illegible]

Head of the Department

Dr. L. LAKSHMANAN, M.E., Ph.D.

SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY DEPARTMENT OF CSE

Submitted for Viva voice Examination held on 20.04.2023

[illegible] [illegible]

Internal Examiner **External Examiner**

DECLARATION

I,**DUDDUKURI SAI SARATH (Reg No - 39110288)**, hereby declare that the Project Phase-2 Report entitled **“HEART STROCK CARDIOVASCULAR USING MACHINE LEARNING”** done by me under the guidance of **Dr. A. C. SANTHA SHEELA, M.E.,Ph.D.,** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering.

Sai Sarath

DATE: 20.04.23

PLACE: Chennai **SIGNATURE OF THE CANDIDATE**

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to the **Board of Management** of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T.Sasikala M.E., Ph. D**, **Dean**, School of Computing, **Dr. L. Lakshmanan M.E., Ph.D.,** Head of the Department of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. A. C .SANTHA SHEELA, M.E.,Ph.D.,** for her valuable guidance, suggestions and constant encouragement paved the way for the successful completion of my phase-3 project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

The suggested system can patient’ s health state based on the forecast of future values and alert carers and medical professionals. We used an actual vital sign collection in this machine-learning-based prediction and categorization algorithm. Several regression techniques, including linear regression and polynomial regression of degrees 2, 3, and 4, have been tried to forecast the vital sign values for the upcoming1–3 minutes. Vital sign predictions are used for carers in 60-second intervals and for emergency medical help in 3-minute intervals. The patient's general health is evaluated using three machine learning classifiers, namely Support Vector Machine (SVM), Naive Bayes, and Decision Tree, Random Forest, K-Nearest neighbor based on the expected vital sign readings. Our findings demonstrate thatthe Decision Tree is useful in providing patients with prompt medical treatment and can accurately categories a patient's health state have based on aberrant vital sign values.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

SVM	-	SUPPORT VECTOR MACHINE
KNN	-	K NEAREST NEIGHBOR
CNN	-	CONVOLUTION NEURAL NETWORKS

CHAPTER 1
INTRODUCTION

Among various life-threatening diseases, heart disease has garnered a great deal of attention in medical research. The diagnosis of heart disease is a challenging task, which can offer automated predictions about the heart condition of a patient so that further treatment can be made effective. The diagnosis of heart disease is usually based on signs, symptoms and physical examination of the patient. There are several factors that increase the risk of heart disease, such as smoking habit, body cholesterol level, family history of heart disease, obesity, high blood pressure, and lack of physical exercise. A major challenge faced by healthcare organizations, such as hospitals and medical centers, is the provision of quality services at affordable costs.1 The quality service implies diagnosing patients properly and administering effective treatments. The available heart disease database consists of both numerical and categorical data. Before further processing, cleaning and filtering are applied on these records in order to filter the irrelevant data from the database.2 The proposed system can determine an exact hidden knowledge, ie, patterns and relationships associated with heart disease from a historical heart disease database. It can also answer the complex queries for diagnosing heart disease; therefore, it can be helpful to health care practitioners to make intelligent clinical decisions. Results showed that the proposed system has its unique potency in realizing the objectivesof the defined mining goals.

Our motive for this project is to predict the Heart diseases for a patient with the maximum amount of accuracy in our prediction. For this we have collected a 2 dataset named Indian patient Heart disease dataset from Kaggle database of Indian Heart patients records and used that dataset in our two modules to predict Heart disease using various machine learning techniques

TYPES OF CARDIOVASCULAR DISEASES

Heart disorders are a category of problems that affect the heart and blood vessels and are sometimes referred to as cardiovascular diseases (CVD). Coronary artery disease (CAD), which includes angina and infarction, is a type of cardiovascular illness. Another type of heart illness is coronary heart disease (CHD), which developswhen plaque, a waxy substance, builds up inside the coronary arteries. Such arteries deliver plasma that is rich in oxygen to the cardiovascular system. When plaque begins to accumulate in these arteries, the resulting condition is known as atherosclerosis. The formation of plaque takes several years. This plaque mightharden or rupture with the passing of time (break open). Plaque calcification gradually narrows coronary arteries, reducing the flow of oxygen-rich blood into the heart. If this plaque punctures, clots may form on its surface.

An arrhythmia, or irregular heartbeat, is a problem with the rate or rhythm of your heartbeat. Your heart may beat too quickly, too slowly, or with an irregular rhythm. It is normal for your heart rate to speed up during physical activity and to slow down while resting or sleeping. There are four types of arrhythmia. They are

- Ventricular fibrillation.
- Ventricular tachycardia.
- Premature ventricular beats (PVCs)
- Torsade’s de pointes.

It is understood that the best ECG lead for monitoring arrhythmias is **V1**. The patient's symptoms were related to a wide QRS complex tachycardia, and V1 is capable of distinguishing ventricular tachycardia (VT) from supraventricular tachycardia (SVT) with aberrant conduction.

CHAPTER - 2
LITERATURE SURVEY

[1] Data mining was used to conduct a survey of systems for predicting cardiovascular disease. The report addressed data mining's usage in pattern development to uncover patterns in patient data. This method bolstered and improved their preexisting health care plan. Their described algorithms had a limited scope and were designed to predict cardiovascular illness. They developed anassociation rule based on a real informational index and the patients' cardiac health records, which resulted in a high rate of accuracy.

[2] The recommended calculation that they performed addresses not only the problem of a large number of principles, but also the proper approval of guidelines backings. To facilitate determination as a preliminary task in the therapeutic database characterization, Kernel F-score Feature Selection was presented. The suggested KFFS strategy first converts healthcare information to binary capacity by increasing BFF & LINEAR skills. Second, the F-score equation, piece capacities of anon-directly separable set of data are transformed into a straight distinct element space, allowing for the determination of treatment datasets.

[3] Data mining classification methods, including DT, NB, & NN, are examined in depth on a Cardiovascular disease database in a report titled "Enhanced Research of Heart Disease Diagnosis utilizing Data Mining Classification Techniques".

[4] Different research methods were evaluated for their precision of results. There was a one hundred percent success rate for Neural Networks, DT has a 99.62% successfulness, whereas Naive Bayes only has a 90% likelihood of success. According to their findings, Neural Networks are the most accurate predictors of heart disease among the three models. Lastly, they included two additional input variables, obesity and smoking status, to complete the model. Due to their significance in the diagnosis of cardiac disorders, these characteristics are appliedto achieve more precise outcomes.

[5] An IHDPS prototype was constructed with the use of data mining methods such DT, NB, & NN, as detailed in the article "Smart Forecasting of Cardiac Diseases Machine Through Data Mining Methods." The outcomes demonstrate the distinct value that each method brings to the table in accomplishing the predetermined mining goals. Whereas conventional decision support systems are unable to address such nuanced "what if" concerns, IDPS is well-suited to do so. Predicting a patient's risk of developing heart disease based on demographic information including age, sex, BP, and glucose levels

[6] Studying how to enhance the precision of weaker methods by merging numerous classifiers, the authors of the publication "Improving the reliability of forecast of risk for heart disease based upon ensembles classifiers" conducted a research utilizing this strategy. The tool was tested using data on cardiovascular disease. The ensemble technique's potential for enhancing prediction accuracy in cardiovascular illnesses was investigated using a comparative analytical strategy. Conclusions from these studies suggest that ensemble methods like bagging and boosting may significantly increase the prediction performance of underperforming classifiers and provide respectable results when it comes to gauging the likelihood that a certain individual will develop cardiovascular disease. With the aid of the ensemble classification, the accuracy of weak classifiers was improved by as much as 7%.

[7] Frames Recovery, ROI Picking, Feature Extraction, and SVM Classifier were all suggested by Sneha(2018) for the study of Echocardiograms, whereby an input video is given into the system and subsequently utilized for training. At 97.30%, the system's accuracy is DCM at 97.40%, ASD at 97.57%, and regular condition at
99.53 percent.

[8] Using techniques such as boosted, svm, k-nn, naive bayes classification, J48, & rf, Martin(2017) provide a method for diagnosing heart failure that persists over time. Unconventional methods using LOSO validating set evaluation yields 97.82% reliability for the models.

[9] The Fast Fourier Transformation is used to analyze time series and was suggested by Raid(2017) as part of a MATLAB-based system for recommending treatments for cardiovascular illness. Together with this, we use an Ensemble model that combines bagging, a synthetic neural network, ordinary least support vector machines, and a naive bayes classifier

2.1 INFERENCES FROM LITERATURE SURVEY

There are some logically strong inferences that can be made from the literature review. The thesis is to composite the ideology of using machine learning algorithms for the prognosis, diagnosis and study of heart diseases and their predictability, it is important to deal majorly with the kind of machine learning algorithms that would suitthe purpose and be centric on the major objectives - being able to predict thepresence of a heart disease in the most accurate possible way. The literaturesurveys conclude the use of Naive Bayes and Support Vector Machine algorithms forthe prediction of heart diseases. There are two major parameters that are involved in understanding the suitability of the respective methodologies and they are - the time taken to execute the prediction process and the accuracy of the predictive result. It is clear through various studies and experimentations that the SVM classifier is the best of all the algorithms owing to the extremely high accuracy rates. But when it comes to the time taken to execute the predictive process, the Naive Bayes classifier reflects higher suitability since it takes the least possible time to execute the process.

From the above-mentioned literature works, it is clear that effective research on this topic has been done and many models have been proposed.

1. It is evident that the above-mentioned systems have their own pros and cons.

2. While some of the recent works involve hybrid technologies and provide better accuracy, they are still far from what is needed.

3. With higher accuracy, comes the need for low computational costs, high processing speed, and most of all, the convenience of use.

2.2 OPEN PROBLEMS IN EXISTING SYSTEM

Although prediction results achieved are promising, these traditional approaches are still far from being highly accurate and efficient. The existing systems are simple and effective but are extremely vulnerable to impact Although prediction results achieved are promising, these traditional approaches are still far from being highly accurate and efficient.

The existing systems are simple and effective but are extremely vulnerable to impact. Moreover, state-of-the-art methods leverage only one algorithm which causes inaccurate results. This could lead practitioners to false assumptions and improper diagnosis and treatments provided to patients.

CHAPTER 3
REQUIREMENT ANALYSIS

3.1 FEASIBILITY STUDIES/RISK ANALYSIS OF THE PROJECT

FEASIBILITY STUDY

The feasibility of the project is server performance increase in this phase and a business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis, the feasibility study of the proposed system is to be carried out. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

● Economic feasibility

● Technical feasibility

● Operational feasibility

ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of funds that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands being placed on the client. The developed system must have modest requirements, as only minimal or null changes are required for implementing this system.

OPERATIONAL FEASIBILITY

The aspect of the study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

3.2 SOFTWARE REQUIREMENTS SPECIFICATION DOCUMENT

Hardware specifications: to educate the user about the system and to make him familiar with it. His level of

- Stable Version of Google Chrome
- Higher RAM, of about 4GB or above
- Processor of frequency 1.5GHz or above

Software specifications:

- Google Collab
- PyStream

3.3 SYSTEM USE CASE

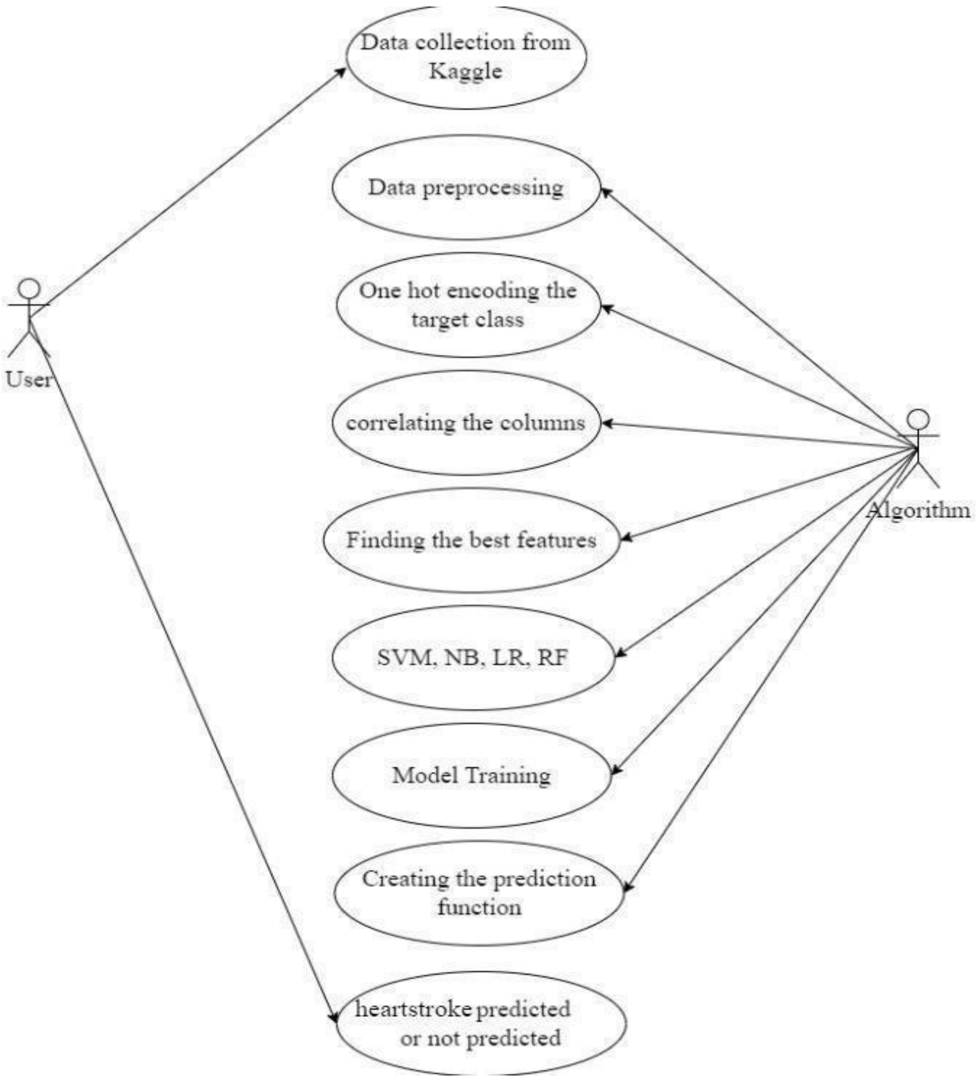

User
Algorithm
Data collection from Kaggle
Data preprocessing
One hot encoding the target class
correlating the columns
Finding the best features
SVM, NB, LR, RF
Model Training
Creating the prediction function
heartstroke predicted or not predicted

Fig 3.1 : Use Case Diagram

3.4 ACTIVITY DIAGRAM

An activity diagram is a kind of graphical representation that may be used to depict events visually. It is made up of a group of nodes that are linked to one another by means of edges. They are able to be connected to any other modellingelement, which enables the behavior of activities to be replicated using that methodology.

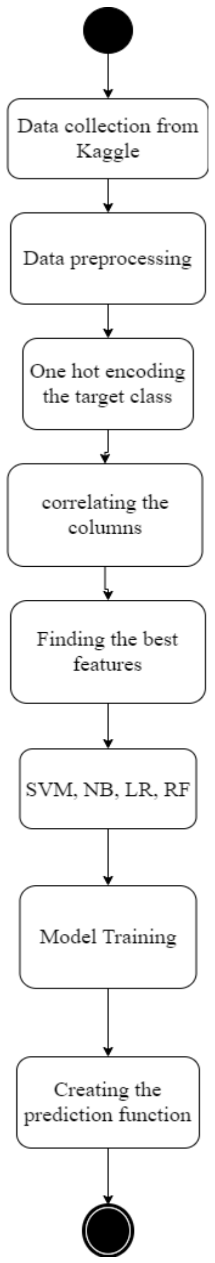

Data collection from Kaggle
Data preprocessing
One hot encoding the target class
correlating the columns
Finding the best features
SVM, NB, LR, RF
Model Training
Creating the prediction function

Fig 3.2 Activity Diagram

3.5 SEQUENCE DIAGRAM

The sequence diagram, also called the event diagram, describes the flow of messages in the system. It helps to visualize various dynamic parameters. He describes the communication between two rescue lines as a series of events arranged in time in which these rescue lines participated during the performance.The lifeline is represented by a vertical bar in UML while the message flow is represented by a vertical dotted line that crosses the bottom of the page. It includes both repetitions and branches.

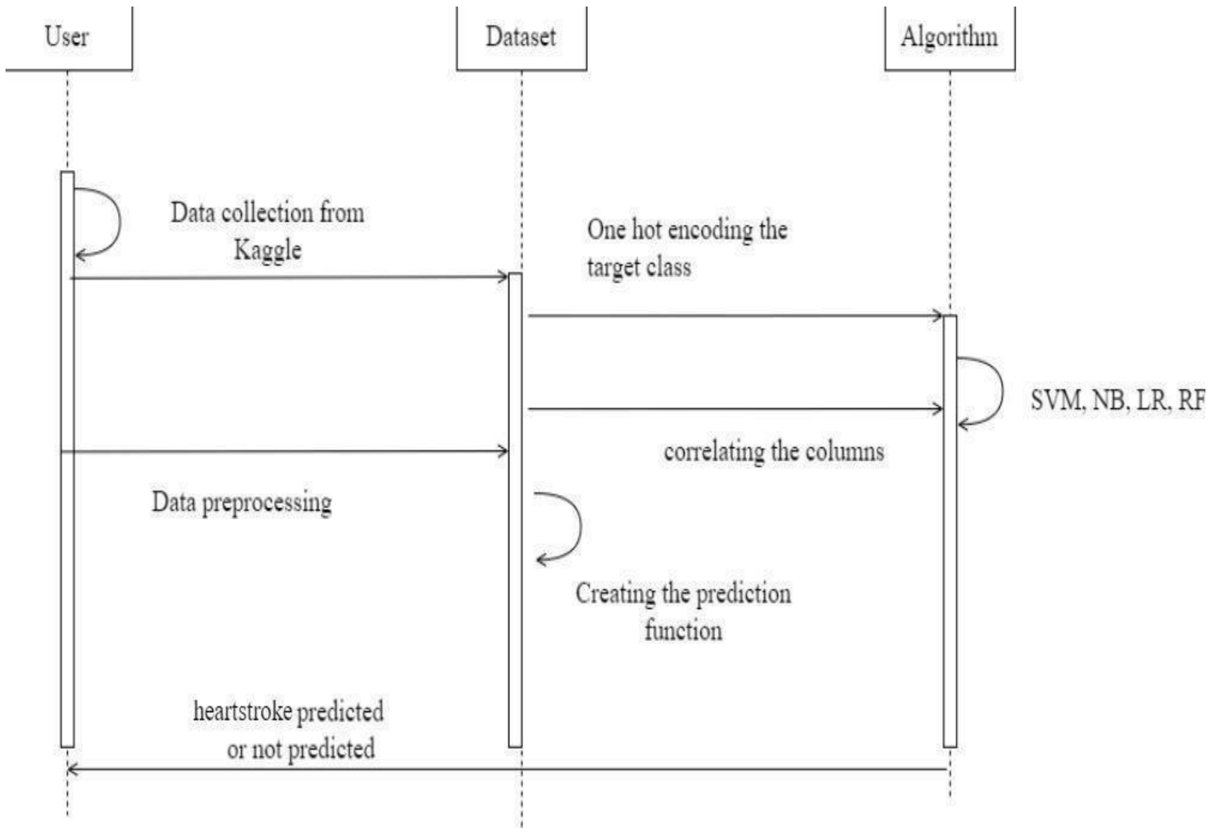

User
Dataset
Algorithm
Data collection from Kaggle
One hot encoding the target class
SVM, NB, LR, RF
correlating the columns
Data preprocessing
Creating the prediction function
heartstroke predicted or not predicted

Fig 3.3 Sequence diagram

CHAPTER 4
DESCRIPTION OF PROPOSED SYSTEM

4.1 ARCHITECTURE / OVERALL DESIGN OF PROPOSED SYSTEM

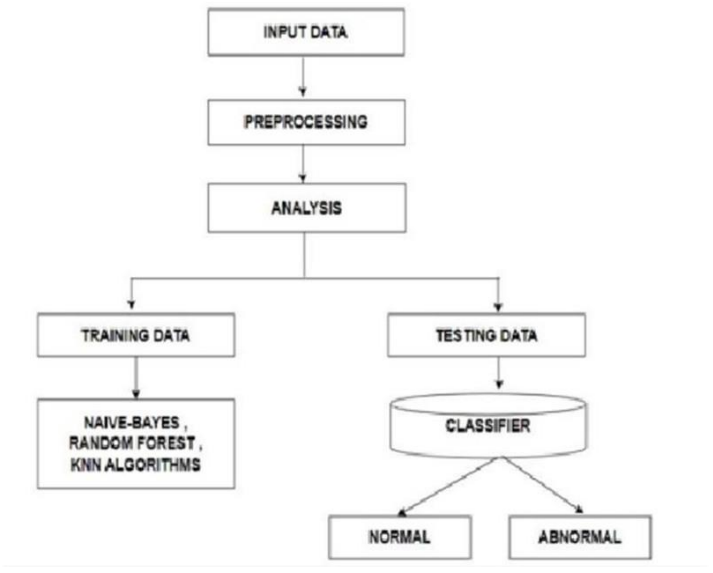


Fig 4.5: System Architecture

Data Collection: The first step is to collect data from various sources such as electronic health records, medical imaging, and patient history. This data will be used to create a comprehensive view of the patient's health status.

Data Preprocessing: The collected data needs to be preprocessed to remove noise and outliers. This may involve data cleaning, feature extraction, and feature selection.

Machine Learning Model: The preprocessed data is fed into a machine learning model that uses various algorithms such as decision tree, random forests,support vector machine, Naïve bayes, K - nearest neighbour to predict the likelihood of a heart stroke. The model may also take into account the patient's demographics, medical history, and lifestyle factors.

Model Training and Validation: The machine learning model needs to be trained using a dataset with known outcomes. The model's performance is then validated using a different dataset to ensure that it is accurate and generalizable.

Deployment: Once the model has been trained and validated, it can be deployed in a real-world setting such as a hospital or clinic. The model will analyze patient data in real-time and provide predictions on the likelihood of a heart stroke

Firstly, we just had to acquire all the datasets. Read the dataset, the information should have several characteristics as sex, age, restBP, cp, cholesterol, FB, goal etc. The information should be investigated to ensure that the data is confirmed. In this case, the sigmoid function is used, which aids in the visual depiction of the labeled data. Methods, such as the DT, LR, CNN algorithms enhance the accuracy.

To put our plan into action, we've chosen to work with the Flask framework. We may show the outcomes on the website itself, which was built using HTML and CSS.

The proposed system of classifying pulse rate using ECG data values using machine learning techniques is an improvement over existing methods in several ways.

Flexibility: The proposed system can use various ml techniques, such as SVM, LR, CNN, DT, and KNN, providing more flexibility in choosing the mostsuitable algorithm for a given dataset.
Improved performance: With the use of advanced deep learning models, such as CNN or RNN, the proposed system has the potential to achieve better performance compared to traditional machine learning algorithms.

Incorporation of additional data sources: For better categorization results, the suggested system may take into account other sources of data as demographics.

Real-time implementation: The proposed system can be implemented in real-time, making it more suitable for practical use in healthcare compared toexisting methods that may have limitations in terms of processing speed

4.2 SELECTED METHODOLOGY OR PROCESS MODEL

Implementing a system for heart stroke prediction using various machine learningalgorithms such as K-Nearest Neighbor (KNN), Linear Regression, Support Vector Classification (SVC), Decision Tree, Random Forest, and Naive Bayes can be a powerfultool in predicting and preventing heart strokes.

To start, the dataset should be cleaned and preprocessed by removing any missing or incorrect data points. The data should then be split into training and testing sets to evaluate the performance of the algorithms. Feature scaling should also be performed to ensure allfeatures have the same range and units.

The first algorithm that can be implemented is KNN, which works by finding the k nearest neighbors to a data point based on a similarity metric. The k neighbors' output labels arethen used to predict the output label of the new data point. The best value of k can be determined using cross-validation.

Linear regression is another algorithm that can be used for heart stroke prediction. This algorithm works by fitting a linear equation to the data, where the dependent variable is the output label, and the independent variables are the input features. The coefficients ofthe linear equation can be determined using gradient descent or other optimization algorithms.

SVC is a powerful algorithm for classification tasks, and it works by finding the optimal hyperplane that separates the data into different classes. The algorithm tries to find the hyperplane with the largest margin between the classes, and it can handle nonlinearly separable data by using kernel functions.

Decision trees are another algorithm that can be used for heart stroke prediction. This algorithm works by splitting the data into subsets based on the values of the input features. The algorithm tries to maximize the information gain at each split to create a tree that accurately predicts the output labels.

MODULE 1: Data Preprocessing

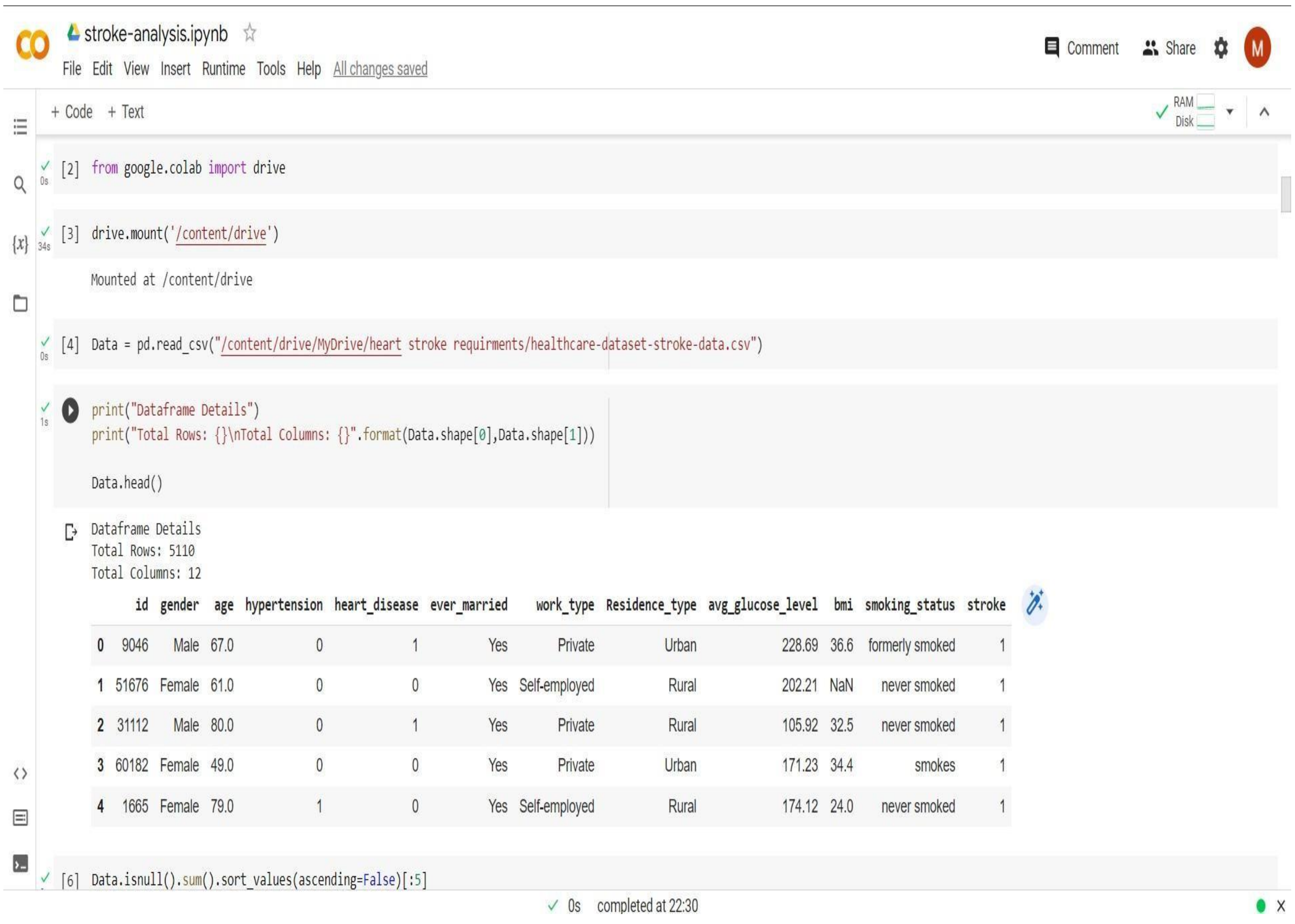
The data preprocessing module is a crucial part of any machine learning projectas it involves cleaning and preparing the data before it is used for training the models. The quality of the data and its preparation can significantly impact the performance of the machine learning models. In the case of heart stroke prediction, the data preprocessing module can be used to clean and preprocess the medical data to ensure accurate predictions.

The first step in the data preprocessing module is to handle missing values. Medical data can often have missing values due to various reasons such as human error, data collection issues, or privacy concerns. The missing values can be handled by imputing them with the mean, median or mode value of the respective feature. In some cases, the missing values can be predicted using regression or other statistical techniques.

The second step is to encode categorical features. Medical data can include categorical features such as gender, smoking status, and hypertension. These features need to be encodedas numerical values for the machine learning models to use them. Categorical features can be encoded using one-hot encoding or label encoding. One-hot encoding creates a binary column for each unique value in the categorical feature, while label encoding assigns a numerical value to each unique value in the feature. The third step is to normalize numericalfeatures. Medical data can include numerical features such as age, blood pressure, and cholesterol levels. These features may have different ranges and units, which can impact theperformance of the machine learning models. Numerical features can be normalized using min-max scaling or standard scaling. Min-max scaling scales the values to a range of 0 to 1,while standard scaling scales the values to have a mean of 0 and a standard deviation of 1.

The fourth step is to perform feature selection. Medical data can include a large number of features that may not all be relevant for predicting heart stroke. Feature selection can be performed using techniques such as principal component analysis (PCA) or feature importance. PCA reduces the number of features by creating a new set of uncorrelated variables, while feature importance calculates the importance of each feature using techniques such as decision trees or random forests.

Once the data is preprocessed, it can be saved in a separate file for training the machine learning models.



```
[2] from google.colab import drive

[3] drive.mount('/content/drive')

Mounted at /content/drive

[4] Data = pd.read_csv("/content/drive/MyDrive/heart stroke requirments/healthcare-dataset-stroke-data.csv")

print("Dataframe Details")
print("Total Rows: {}\nTotal Columns: {}".format(Data.shape[0],Data.shape[1]))

Data.head()

Dataframe Details
Total Rows: 5110
Total Columns: 12
```

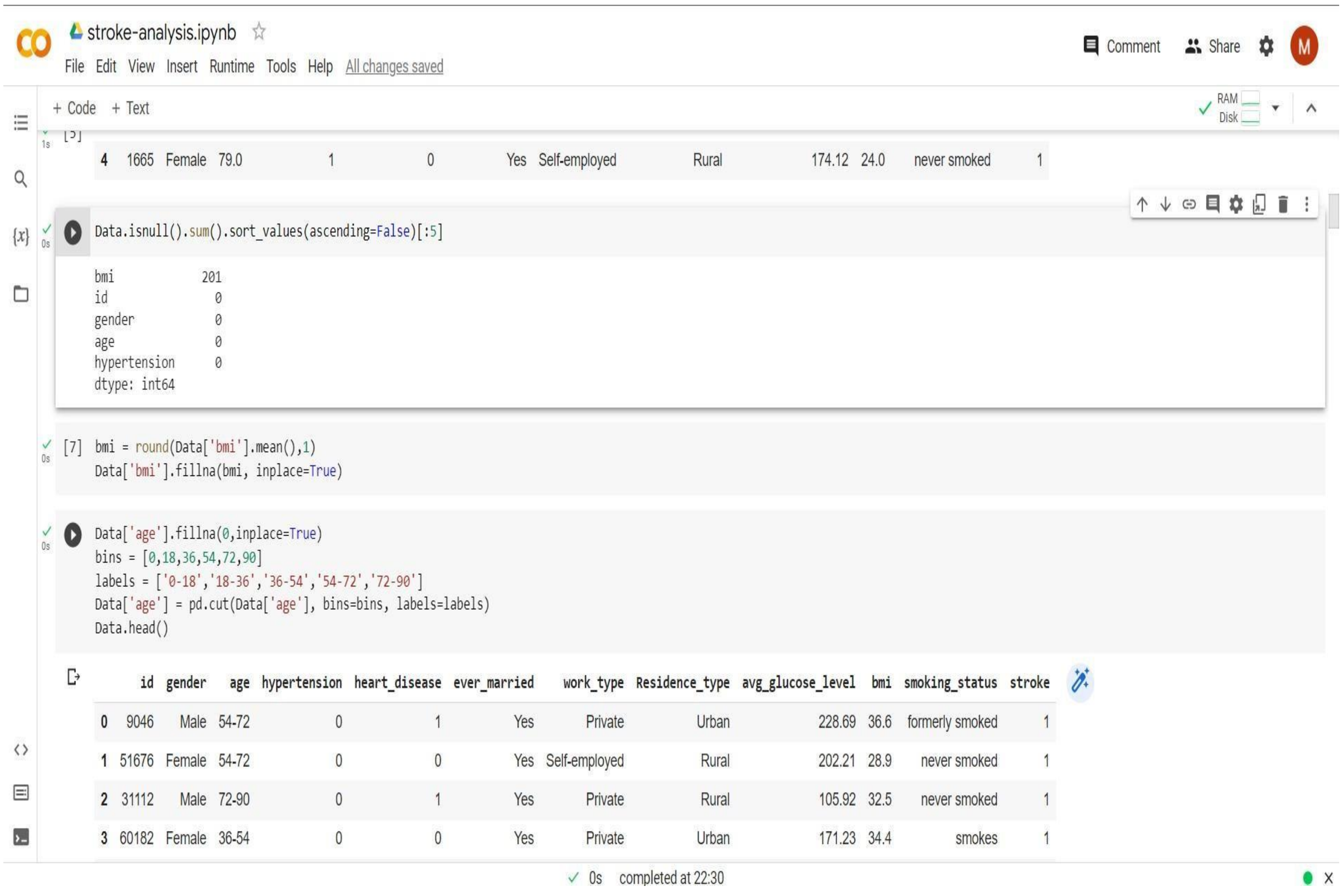
	id	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	smoking_status	stroke
0	9046	Male	67.0	0	1	Yes	Private	Urban	228.69	36.6	formerly smoked	1
1	51676	Female	61.0	0	0	Yes	Self-employed	Rural	202.21	NaN	never smoked	1
2	31112	Male	80.0	0	1	Yes	Private	Rural	105.92	32.5	never smoked	1
3	60182	Female	49.0	0	0	Yes	Private	Urban	171.23	34.4	smokes	1
4	1665	Female	79.0	1	0	Yes	Self-employed	Rural	174.12	24.0	never smoked	1

```
[6] Data.isnull().sum().sort_values(ascending=False)[:5]
```

Fig -4.1-DATA PROCESSING

MODULE 2: Model training

The model training module can be used to train the machine learning models using the preprocessed data. This module can include functions to train different machine learning models such as KNN, linear regression, SVC, decision tree, random forest, and Naive Bayes. The module can also include functions for hyperparameter tuning and cross- validation to improve the performance of the models. For example, hyperparameter tuning can be performed using techniques such as grid search or random search to find the best hyperparameters for the models. Cross-validation can be performed to evaluate the performance of the models on different subsets of the data. The trained models can be savedin separate files for testing and evaluation.



```
[5]
4   1665  Female  79.0   1   0   Yes  Self-employed   Rural   174.12  24.0  never smoked  1
```

```
Data.isnull().sum().sort_values(ascending=False)[:5]

bmi             201
id                0
gender            0
age               0
hypertension      0
dtype: int64
```

```
[7] bmi = round(Data['bmi'].mean(),1)
    Data['bmi'].fillna(bmi, inplace=True)
```

```
Data['age'].fillna(0,inplace=True)
bins = [0,18,36,54,72,90]
labels = ['0-18','18-36','36-54','54-72','72-90']
Data['age'] = pd.cut(Data['age'], bins=bins, labels=labels)
Data.head()
```

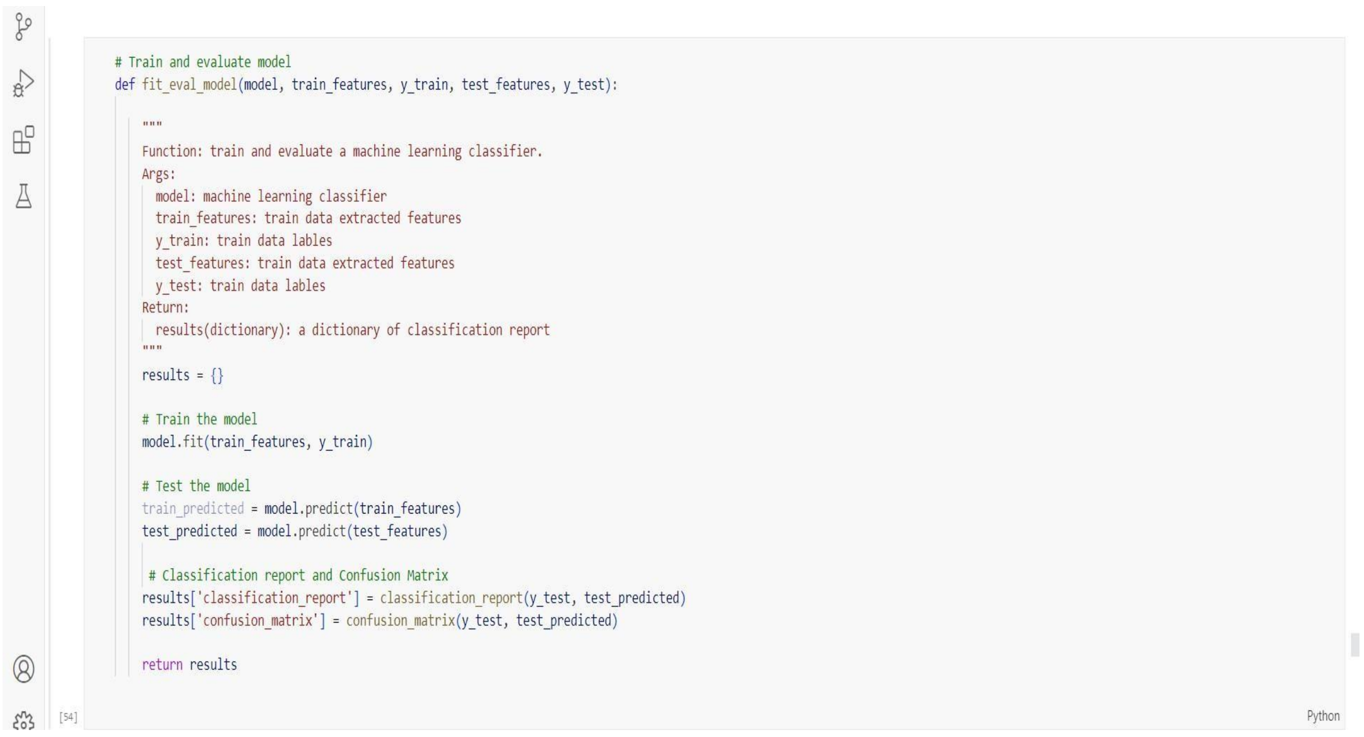
	id	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	smoking_status	stroke
0	9046	Male	54-72	0	1	Yes	Private	Urban	228.69	36.6	formerly smoked	1
1	51676	Female	54-72	0	0	Yes	Self-employed	Rural	202.21	28.9	never smoked	1
2	31112	Male	72-90	0	1	Yes	Private	Rural	105.92	32.5	never smoked	1
3	60182	Female	36-54	0	0	Yes	Private	Urban	171.23	34.4	smokes	1

Fig – 4.2 – MODEL TRANING

MODULE 3: Model Evaluation

The model evaluation module is a critical part of a heart stroke prediction project,as it allows us to test and evaluate the performance of the trained machine learningmodels on new data. The module should include functions to load the preprocessed data and the trained models, and use them to predict the output labelsfor the test data. It should also include functions to calculate the performance metrics such as accuracy, precision, recall, and F1-score for each of the models.

Accuracy is a common performance metric that measures the proportion of correctly classified instances. It can be calculated as the number of correctly classified instances divided by the total number of instances. However, accuracy can be misleading if the data is imbalanced, as the model may simply predict the majority class without learning anything about the minority class. Therefore, precision, recall, and F1-score can provide more insights into the model's performance.

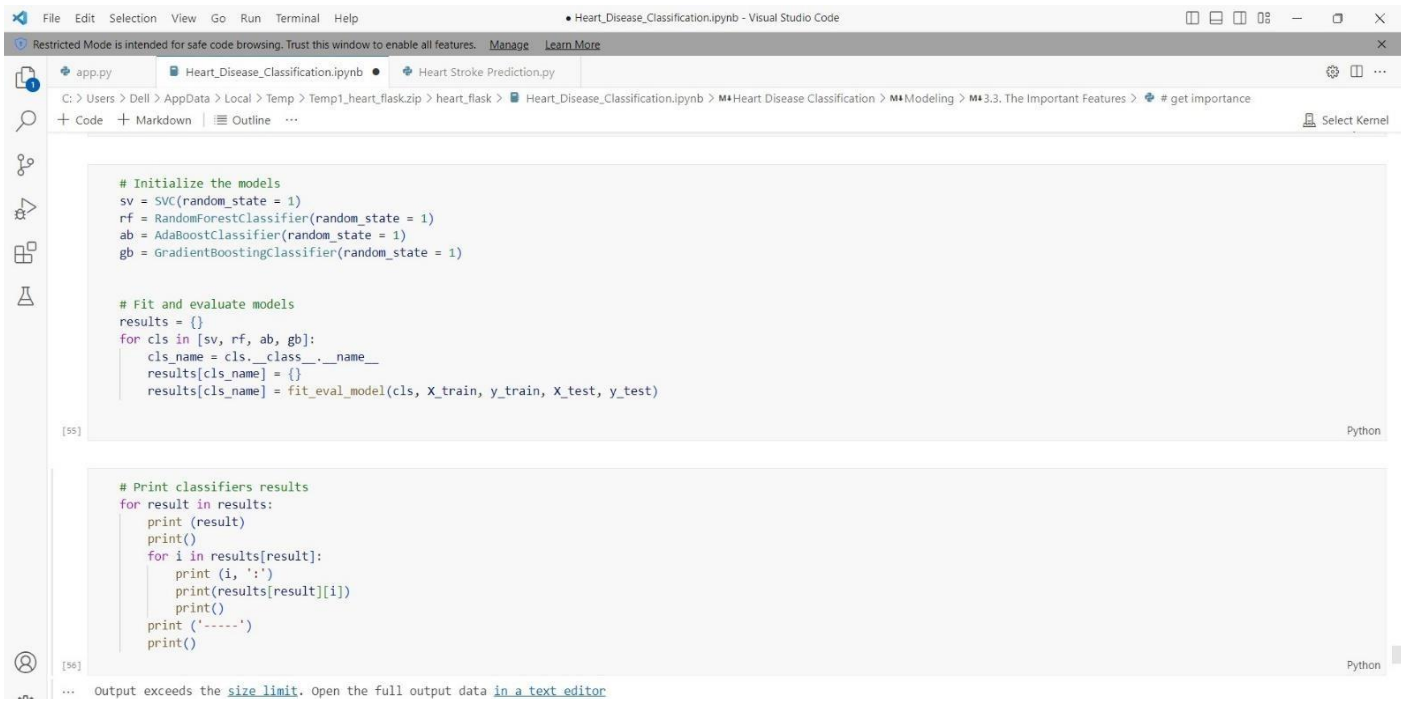


```python
# Train and evaluate model
def fit_eval_model(model, train_features, y_train, test_features, y_test):

    """
    Function: train and evaluate a machine learning classifier.
    Args:
      model: machine learning classifier
      train_features: train data extracted features
      y_train: train data lables
      test_features: train data extracted features
      y_test: train data lables
    Return:
      results(dictionary): a dictionary of classification report
    """
    results = {}

    # Train the model
    model.fit(train_features, y_train)

    # Test the model
    train_predicted = model.predict(train_features)
    test_predicted = model.predict(test_features)

     # Classification report and Confusion Matrix
    results['classification_report'] = classification_report(y_test, test_predicted)
    results['confusion_matrix'] = confusion_matrix(y_test, test_predicted)

    return results
```

[54]

Python

Fig-4.3-Model Evaluation



```python
# Initialize the models
sv = SVC(random_state = 1)
rf = RandomForestClassifier(random_state = 1)
ab = AdaBoostClassifier(random_state = 1)
gb = GradientBoostingClassifier(random_state = 1)


# Fit and evaluate models
results = {}
for cls in [sv, rf, ab, gb]:
    cls_name = cls.__class__.__name__
    results[cls_name] = {}
    results[cls_name] = fit_eval_model(cls, X_train, y_train, X_test, y_test)
```

```python
# Print classifiers results
for result in results:
    print (result)
    print()
    for i in results[result]:
        print (i, ':')
        print(results[result][i])
        print()
    print ('-----')
    print()
```

```
... Output exceeds the size limit. Open the full output data in a text editor
```

Fig – 4.4 – Model Evaluation

Precision measures the proportion of true positives among the instances predicted as positive. It can be calculated as the number of true positives divided by the sum of true positives and false positives. Recall measures the proportion of true positives among the instances that are actually positive. It can be calculated as the number of true positives divided by the sum of true positives and false negatives. F1-score is the harmonic mean ofprecision and recall, and it provides a balance between the two metrics. It can be calculatedas 2 times the product of precision and recall divided by the sum of precision and recall. The performance metrics can be calculated for each of the machine learning models trainedin the model training module. The metrics can be compared to choose the best model for heart stroke prediction. The best model is the one with the highest accuracy, precision, recall, or F1-score, depending on the project's goals and constraints. The model evaluationmodule can also include functions to visualize the performance metrics using graphs or charts to provide insights into the model's performance. For example, a confusion matrix can be used to visualize the number of true positives, false positives, true negatives, and false negatives for each class. A receiver operating characteristic (ROC) curve can be usedto visualize the tradeoff between true positive rate and false positive rate for different classification thresholds.

A precision-recall curve can be used to visualize the tradeoff between precision and recall for different classification thresholds.

In conclusion, the model evaluation module is a critical part of a heart stroke prediction project, as it allows us to test and evaluate the performance of the trained machine learningmodels on new data. The module should include functions to calculate the performance metrics such as accuracy, precision, recall, and F1-score for each of the models. The metrics can be compared to choose the best model for heart stroke prediction, and the module can include functions to visualize the performance metrics using graphs or chartsto provide insights into the model's performance.

Module-3 WebApp

An important part of building a machine learning model is to share the model we have built with others. No matter how many models we create, if they remain offline, very few people will be able to see what we are achieving. That's why we should deploy our templates, so that anyone can play with them through a nice User Interface (UI). For this system, we build a single page web application with Flask as the UI of our system. It will take input and predict whether the user data given the person is having the chance of chronic heart disease or not in 10 years.

Flask is a micro web framework written in Python. It is categorized as a microframework as it does not require any specific tools or libraries. It does not have a database abstraction layer, form validation or any other component where existing third-party libraries provide common features.

Advantages of proposed methodology

With the use of this procedure, a risk factor of a person may be predicted, and after that, the individual can get treatment for their ailment. Treatment is needed since the incidence rate of cardiovascular illnesses is growing at an unexpectedly rapid pace, and a large number of individuals are ignorant that the ML model will be more effective to induce in the task. The ML model will be more effective to induce in the task. As a direct result of this, it is anticipated that the prevalence of cardiovascular illnesses would continue to increase

4.3 DESCRIPTION OF SOFTWARE FOR IMPLEMENTATION ANDTESTING PLAN OF THE PROPOSED MODEL/SYSTEM

Testing the model is an important step in heart stroke prediction as it evaluates the model's performance on new and unseen data. Here is a summary of the steps involved in testing the model for heart stroke prediction:

1. **Split the data**: The first step is to split the data into training and testing sets. The training set is used to train the model, while the testing set is used to evaluate the model's performance. The recommended split is usually 80:20 or 70:30 for trainingand testing, respectively.

2. **Train the model**: Next, the model is trained on the training set using the optimized hyperparameters and feature selection techniques. The model's performance isevaluated using appropriate evaluation metrics such as accuracy, precision, recall, F1-score, and area under the ROC curve.

3. **Test the model**: Once the model is trained, it is tested on the testing set to evaluate its performance on new and unseen data. The same evaluation metrics are used to evaluate the model's performance on the testing set.

4. **Evaluate the results**: Finally, the results are evaluated to determine if the model is performing well on new and unseen data. If the model's performance is satisfactory,it can be deployed for heart stroke prediction in clinical practice. However, if the model's performance is not satisfactory, further optimization and testing may be necessary.

REFINEMENT AND DEPLOYMENT OF MODEL

Refinement and deployment of a model for heart stroke prediction involves improving the model's performance and deploying it in a real-world setting. Here is a summary ofthe steps involved in refinement and deployment of the model:

1. **Refinement:** Once the model has been tested and evaluated, any necessary refinementsare made to improve the model's performance. This could involve adjusting hyperparameters, improving feature selection, or exploring different algorithms or ensemble techniques. The model is then retrained and retested to evaluate itsperformance.

2. **Validation:** Before deployment, the model must be validated to ensure that it performswell on new data. This can be done using techniques such as cross-validation, bootstrapping, or hold-out validation. The model's performance on validation data is compared to its performance on the testing data to ensure that it generalizes well to new data.

3. **Deployment:** Once the model has been validated, it can be deployed in a real-world setting. This involves integrating the model into a software application or system that is used in clinical practice. The deployment process must be carefully planned and tested to ensure that the model is robust, reliable, and secure.

4. **Monitoring and maintenance**: After deployment, the model must be monitored and maintained to ensure that it continues to perform well over time. This involves monitoring the model's performance, updating the model if necessary, and addressingany issues or bugs that arise.

4.4 TRANSITION/ SOFTWARE TO OPERATIONS PLAN

- Scikit learn
- Tensor flow
- Keras
- pyTorch
- XGBoot
- Collab
- Stable Google Chrome
- Intel® Core™ i5 processor 8250U at 1.60 GHz or 1.80 GHz, 8 GB of DRAM.
- Disk space 2TB. 3. Operating System: 64-bit Windows 10 Pro
- PIP and NumPy: Ubuntu*, Python 3.6.2, NumPy 1.13.1, scikit-learn 0.18.2 Windows: Python 3.6.2, PIP and NumPy 1.13.1, scikit-learn 0.18.2 Intel® Distribution for Python* 2018.

CHAPTER 5
IMPLEMENTATION DETAILS

5.1 DEVELOPMENT AND DEPLOYMENT SETUP

Input Design

The following factors should be considered while designing input for a comparison study on machine learning for Heart disorders diagnosis.

Data sources

In order to diagnose heart disease, it is necessary to identify which data sources willbe used, such as electronic medical records, Heart function tests, imaging data, and clinical data analysis.

Data Preprocessing

The system will need to perform data preprocessing in order to prepare the data for analysis, which will include data cleaning, data transformation, and feature engineering in order to prepare the data for analysis.

Feature engineering

Identify and extract from the data the key features that will be used to train the machine learning model based on the features found in the data. In some cases, it may be necessary to do this with automated feature engineering methods or with domain expertise in order to achieve this goal.

Data storage

Determine the appropriate storage mechanism for the input data, such as a database or data warehouse. This will ensure that the data is easily accessible and can be efficiently processed by the machine learning models.

Data integration

The aim of this project is to integrate data from different sources and formats in order to create a unique dataset that can be used for the diagnosis of Heart disorders.

Output Design

When designing output for Heart disorder diagnosis using machine learning techniques, a comparative study, it is important to consider the following factors.

Diagnosis and classification

Diagnose and classify Heart disorders based on input data. A specific disorder, suchas fatty Heart, cirrhosis, or hepatitis, should be identified, along with its severity.

5.2 ALGORITHMS

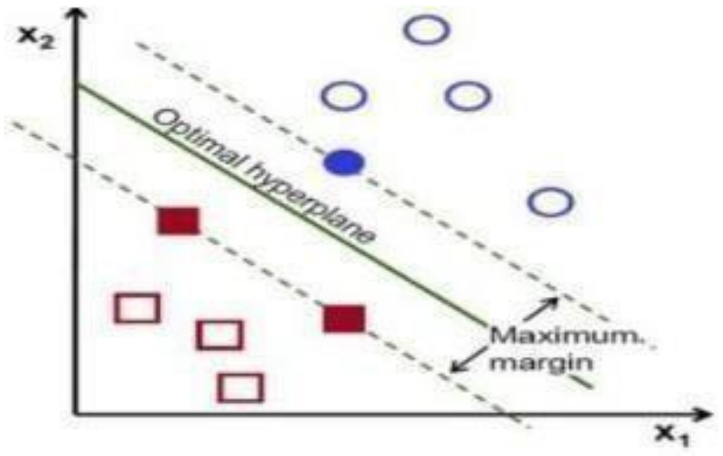


FIG 5.1 : SVM CLASSIFIER

\

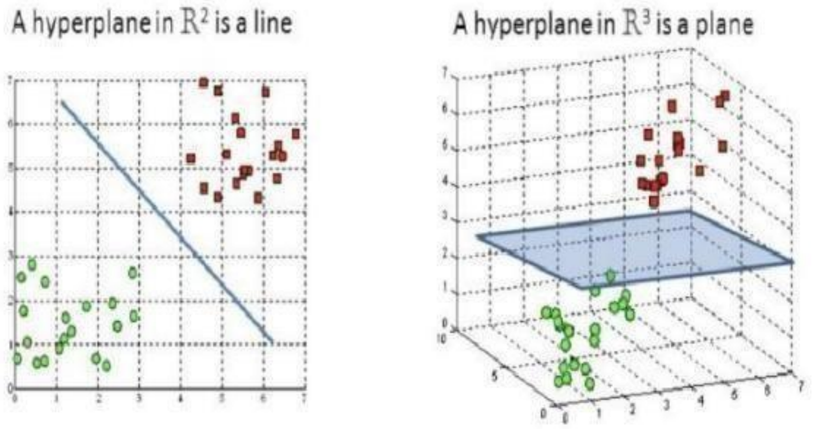


FIG 5.2 : HYPERPLANES IN 2D & 3D

➢ **Logistic Regression Algorithm :**

Logistic regression is one of the most popular Machine Learning algorithms, which comes under the Supervised Learning technique. It is used for predicting the categorical dependent variable using a given set of independent variables.

Logistic regression predicts the output of a categorical dependent variable. Therefore the outcome must be a categorical or discrete value. It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, it gives the probabilistic values which lie between 0 and 1.

Logistic Regression is much similar to the Linear Regression except that how they are used. Linear Regression is used for solving Regression problems, whereas Logistic regression is used for solving the classification problems.

- **Random Forest Algorithm :**

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning**,** which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model.

As the name suggests, "Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output..

- **K-Nearest Neighbor(KNN) Algorithm :**

K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique. K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories. K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm.

L-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems.K-NN is a non-parametric algorithm, which means it does not make any assumption on underlying data.It is also called a lazy learner algorithm because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset.KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data.

- **Decision Tree Classification Algorithm :**

Decision Tree is a Supervised learning technique that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome**.**

In a Decision tree, there are two nodes, which are the Decision Node and Leaf Node**.** Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches. The decisions or the test are performed on the basis of features of the given dataset.

It is a graphical representation for getting all the possible solutions to a problem/decision based on given conditions**.** It is called a decision tree because, similar to a tree, it starts with the root node, which expands on further branches and constructs a tree-like structure. In order to build a tree, we use the CART algorithm**,** which stands for Classification and Regression Tree algorithm**.** A decision tree simply asks a question, and based on the answer (Yes/No), it further split the tree into subtrees.

- **Support Vector Machine Algorithm :**

Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning.

The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.

SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine. Consider the below diagram in which there are two different categories that are classified using a decision boundary or hyperplane

5.3 TESTING

Test Case 1: Input Validation

Description: The system should validate that input data is correct and complete.

Test Steps: Enter invalid data, such as a negative age or a missing blood pressure reading. Verify that the system provides an appropriate error message.

Test Case 2: Accuracy of Prediction

Description: The system should accurately predict the likelihood of a heart stroke.

Test Steps: Enter test data for patients who have had a heart stroke and patients who have not. Verify that the system accurately predicts the likelihood of a heart stroke for each patient.

Test Case 3: Sensitivity and Specificity

Description: The system should have high sensitivity and specificity.

Test Steps: Enter test data for patients who have had a heart stroke and patients who have not. Calculate the sensitivity and specificity of the system's predictions. Verify that the sensitivity and specificity are high.

Test Case 4: User Interface

Description: The system should have a user-friendly interface.

Test Steps: Evaluate the system's user interface to ensure that it is easy to use and understand. Verify that the system provides clear and concise information about the patient's predicted risk of a heart stroke.

Test Case 5: Performance and Scalability

Description: The system should perform well and be scalable.

Test Steps: Enter test data for a large number of patients. Verify that the system can handlea large number of requests without significantly impacting performance. Verify that the system can be easily scaled up to handle additional requests as needed.

5.4 PERFORMANCE ANALYSIS

The performance analysis for the heart stroke prediction project involves evaluating the performance of the system based on different performance metricsand comparing them to the project's goals and constraints. Here are some possibleperformance metrics that can be used for the analysis:

Detection Accuracy: The detection accuracy is a critical performance metric forthe heart stroke prediction project. The accuracy measures how accurately the system can predict heart stroke cases. The accuracy can be calculated by comparing the predicted labels to the actual labels in the test dataset. The heart stroke prediction project should aim for an accuracy of at least 80%, and a higheraccuracy is preferred. If the accuracy is lower than the target, the system's trainingdata, model selection, or hyperparameters may need to be adjusted.

Response Time: The response time is the time taken by the system to make predictions. A shorter response time means that the system can make predictions faster, which can be critical in some applications. The response time can be measured using tools such as the Python time library or the Unix time command.The heart stroke prediction project should aim for a response time of a few seconds or less, depending on the application requirements. If the response time is longer than the target, the system's algorithm or hardware may need to be optimized.

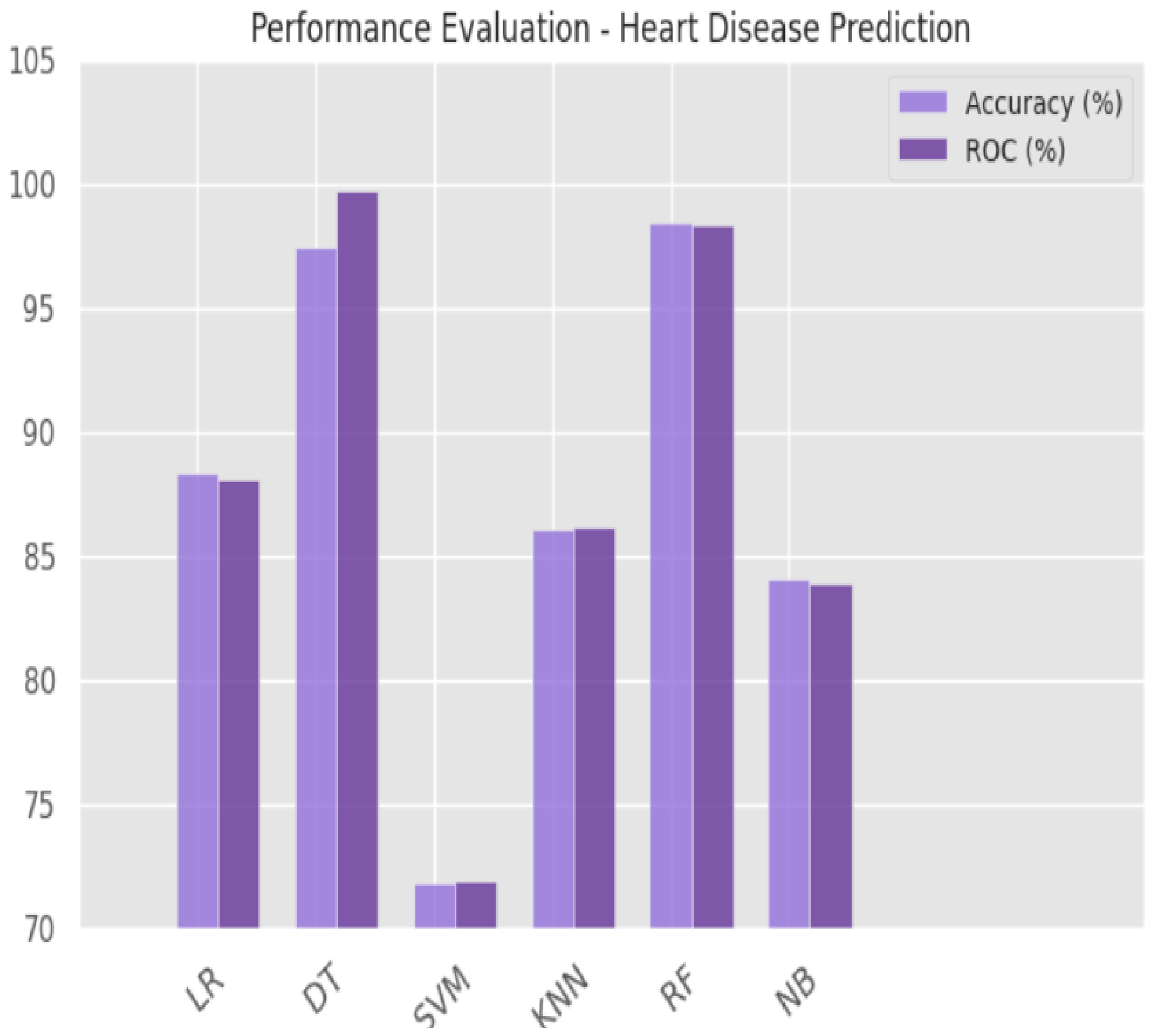


Fig – 5.3 – Performance Evaluation

In conclusion, the performance analysis for the heart stroke prediction project involves evaluating the system's performance based on different metrics such as detection accuracy, response time, resource usage, system uptime, and user satisfaction. By comparing these metrics to the project's goals and constraints, the system's strengths and weaknesses can be identified, and appropriate measures canbe taken to improve its performance.

CHAPTER 6
RESULTS AND DISCUSSION

In order to detect the likelihood that a user has heart disease, a ml model was developed in this research. For this machine learning model, the input data consists of 13 sets of human test results. During the processing and cleaning ofthis dataset, it was verified that no missing or invalid data values existed. The data set was broken down into two separate halves, x and y. Where the output (represented by the dependent variable y) is a function of the x parameter (which includes the 13 characteristics representing the various test outcomes). Standard Scaler was used to transform the x-axis value. Additional categories were created for the x and y variables: y-test, x-test, x-train & y- train.

These y-train & x-train were learned with the aid of the supervised learning methods CNN, K-NN, LR, SVM, & RF. Using varying values for n, the four methodswere utilized to determine the accuracy %. The maximum accuracy is achieved by K Neighbors at n = 1, by CNN at n = 1 by SVM at n=1, and by Random Forest at n = 10. After conducting extensive tests, we settled on using SVM because of its reputation as one among the most reliable forecasting models.

Flask, an Api, was used to save and upload the SVM model to the web. We constructed an HTML website with Thirteen variables using Flask so that people may submit the findings of their different tests and let the model assess the likelihood that they have pulse rate. The model's results will be shown on aweb-based portal for the patient's convenience

TEST ID	ALGORITHM	ACTUAL OUTPUT	PREDICTED OUTPUT	STATUS
1	Logistic Regression	100	85.0	SUCCESS
2	Random Forest	100	80.0	SUCCESS
3	KNN	100	85.0	SUCCESS
4	Decision Tree	100	69.0	SUCCESS
5	SVM	100	80.0	SUCCESS

TABLE 6.1 : OUTPUT TAB

CHAPTER 7

CONCLUSION AND FUTURE ENHANCEMENTS

7.1 CONCLUSION

Machine learning techniques have shown promising results in predicting heart stroke. Several studies have compared the performance of different machine learning algorithms in predicting heart stroke, and the results have varied depending on the dataset, features, and evaluation metrics used. In general, studies have found that ensemble methods such as random forests and gradient boosting classifiers perform better than other algorithms, such as logistic regression, support vector machines, and neural networks. However, the performance differences between these algorithms may not be statistically significant in all cases. It is also important to note that the quality of the data and the choice of features can have a significant impact on the performance of the model. Preprocessing and feature selection techniques, such as imputation, normalization, and feature engineering, can improve the performance of the model by reducing noise and extracting relevant information. Overall, machine learning techniques have the potential to improve the accuracyand efficiency of heart stroke prediction, and further research is needed to identify the mosteffective algorithms and feature selection techniques for different datasets and evaluation metrics. The prevention and disease progression can be aided by early identification. Early diagnosis and the finding of important causal factors can be aided by machine learning technologies. The proposed technique generates a deep learning model that can predict cardiovascular illnesses and heart attacks. The optimal solution for the job is the SVM algorithm. The model suggests that new machine learning algorithms usually lead to better prediction accuracy. The prevention and disease progression can be aided by early identification. Early diagnosis and the finding of important causal factors can be aided by machine learning technologies. The proposed technique generates a deep learning model that can predict cardiovascular illnesses and heart attacks. The optimal solution for the job is the SVM algorithm. The model suggests that new machine learning algorithms usually lead to better prediction accuracy.

7.2 FUTURE WORK

The project can be further enhanced by deploying the machine learning model obtained using a web application and a larger dataset could be used for predictionto give higher accuracy and produce better results..

1. **Incorporating new risk factors**: Current heart stroke prediction models use arange of risk factors such as age, gender, blood pressure, cholesterol levels, smoking habits, and medical history. Future research can explore incorporatingnew risk factors such as genetic information, diet, and lifestyle habits.

2. **Improving prediction accuracy**: While machine learning models have shownpromise in heart stroke prediction, further research can explore new techniquesto improve the accuracy and reliability of the prediction models. This can involve the use of advanced algorithms, data preprocessing techniques, and feature engineering methods.

3. **Developing personalized prediction models**: Personalized heart stroke prediction models can help healthcare professionals identify patients who are at a higher risk of heart stroke and provide targeted interventions. Future research can explore developing personalized prediction models that consider individual differences in risk factors and lifestyles.

4. **Integration with electronic health records (EHRs):** Integrating heart strokeprediction models with EHRs can allow healthcare professionals to access patient data in real-time and make informed decisions about patient care. Future research can explore developing EHR-integrated prediction models that can beintegrated into clinical decision support systems.

5. **Evaluating the impact of prediction models:** It is essential to evaluate the impact of heart stroke prediction models in real-world settings to understand

their effectiveness in clinical practice. Future research can focus on evaluatingthe impact of prediction models on patient outcomes, healthcare costs, and healthcare provider decision-making.

7.3 RESEARCH ISSUES

Few Heart problems cannot lead to change in ECG images like;

- Atrial fibrillation/flutter.
- Heart attack.
- Heart failure.
- Multifocal atrial tachycardia.
- Paroxysmal supraventricular tachycardia.
- Sick sinus syndrome.
- Wolff-Parkinson-White syndrome.

Common mistakes are: Left-right arm reversals lead to a negative complex in lead I with a negative P wave in lead I. They are one of the most common causes of right axis deviation on the ECG! Arm-foot switches lead to a very small or 'far field' signal in leads II or III.

Valvular defects cannot be detected using an ECG. Chest X-ray can be used to determine such defects. Therefore, an ECG can detect arrhythmia, myocardial infarction and also heart block but not valvular defects.

It cannot provide definite diagnosis of congestive heart failure. Prognosis during anesthesia or surgical procedures is unpredictable. Heart chamber wall thickness cannot be measured. Diseases of heart valves, endocardium, or pericardium cannot.

7.4 IMPLEMENTATION ISSUES

The Algorithms used in our project did not give a 100% accuracy, so the prediction isnot 100% feasible. Clinical diagnosis and diagnosis using our project may differ slightly because the prediction is not 100% accurate. Medical diagnosis is considered as a significant yet intricate task that needs to be carried out precisely and efficiently. The automation of the same would be highly beneficial. Clinical decisions are often made based on the doctor's intuition and experience rather than on the knowledge rich data collected from the dataset.

REFERENCES: -

[1] A. Singh et al., "Heart Disease Prediction Using Machine Learning Algorithms", 2020 International Conference on Electrical and Electronics Engineering (ICE3), pp. 452-457, February 2020.

[2] K. Hashi and M.S.U. Zaman, "Developing a Hyperparameter Tuning Based Machine Learning Approach of Heart Disease Prediction", Journal of Applied Science & Process Engineering, vol. 7, no. 2, pp. 631-647, 2020.

[3] Rahul Katarya and Polipireddy Srinivas, "Predicting heart disease at early stages using machine learning: A survey", 2020 International Conference on Electronics and Sustainable Communication Systems (ICESC), 2020.

[4] Davide Chicco and Giuseppe Jurman, "Machine learning can predict survivalof patients with heart failure from serum creatinine and ejection fraction alone",BMC medical informatics and decision making, vol. 20.1, pp. 1-16, 2020.

[5] A. M.A and P. A. Thomas, "Comparative Review of Feature Selection and Classification modelling", 2019 International Conference on Advances in Computing Communication and Control (ICAC3), pp. 1-9, 2019

[6] "How to read an Electrocardiogram (ECG). Part One: Basic principles of the ECG. The normal ECG." [Online]. Available: http://www.southsudanmedicaljournal.com/archive/may-2010/how-to-read-anelectrocardiogram-ecg.-part-one-basic-principles-of-the-ecg.-the-normal-ecg.html. [Accessed: 28-Apr-2019].

[7] F. Mendonca, R. Manihar, A. Pal and S. U. Prabhu, "Intelligent Cardiovascular Disease Risk Estimation Prediction System", 2019 International Conference on Advances in Computing Communication and Control (ICAC3), pp. 1-6, 2019.

[8] Ashir Javeed et al., "An intelligent learning system based on random search algorithm and optimized random forest model for improved heart disease detection", IEEE Access, vol. 7, pp. 180235-180243, 2019.

[9] Mohammad Shafenoor Amin, Yin Kia Chiam and Kasturi Dewi Varathan, "Identification of significant features and data mining techniques in predicting heart

disease", Telematics and Informatics, vol. 36, pp. 82-93, 2019

[10] A. Gavhane, G. Kokkula, I. Pandya and K. Devadkar, "Prediction of Heart Disease using Machine Learning", 2018 Second International Conference on Electronics Communication and Aerospace Technology (ICECA), pp. 1275- 1278, 2018.

APPENDIX

A. SAMPLE CODE

# Heart Stroke ClassificationThe
dataset has 14 attributes:

* **age:** age in years

* **sex:** sex (1 = male; 0 = female).

* **cp:** chest pain type (Value 0: typical angina; Value 1: atypical angina; Value 2:non-anginal pain; Value 3: asymptomatic).

* **trestbps:** resting blood pressure in mm Hg on admission to the hospital.

* **chol:** serum cholestoral in mg/dl.

* **fbs:** fasting blood sugar > 120 mg/dl (1 = true; 0 = false).

* **restecg:** resting electrocardiographic results (Value 0: normal; Value 1: havingST-T wave abnormality; Value 2: probable or definite left ventricular hypertrophy).

* **thalach:** maximum heart rate achieved.

* **exang:** exercise induced angina (1 = yes; 0 = n

* **oldpeak:** ST depression induced by exercise relative to rest.

* **slope:** the slope of the peak exercise ST segment (Value 0: upsloping; Value 1:flat; Value 2: downsloping).

* **ca:** number of major vessels (0-3) colored by flourosopy.

* **thal:** thalassemia (3 = normal; 6 = fixed defect; 7 = reversable defect).

```
* **target:** heart disease (1 = no, 2 = yes)

## Import Libraries

# Import needed libraries

import pandas as pd
import numpy as np

import matplotlib.pyplot as plt

%matplotlib inline
import seaborn as sns
import re

from sklearn.svm import SVC

from sklearn.tree import DecisionTreeClassifier

from sklearn.ensemble import RandomForestClassifier, AdaBoostClassifier,

GradientBoostingClassifier

from sklearn.model_selection import train_test_split, GridSearchCV

from sklearn.metrics import accuracy_score, confusion_matrix, classification_report,
make_scorer


from plotly.offline import iplot
import plotly as py

import plotly.tools as tls


import pickle

## 1. Data Preparation and Data Exploration#
Read data in the excel file
```

```
df = pd.read_csv('data.csv')
df.head()

df.shape
df.info()
df.describe()

#Check null values
df.isnull().sum()
df['target'].value_counts()

# Percentage of patients have and do not have heart disease
df['target'].value_counts()/df.shape[0]*100



# Create a plot to display the percentage of the positive and negative heart disease
labels = ['yes', 'No']

values = df['target'].value_counts().values



plt.pie(values, labels=labels, autopct='%1.0f%%')
plt.title('Heart Disease')

plt.show()





### Attributes Correlation#
Correlation map
plt.figure(figsize=(15, 15))

sns.heatmap(df.corr(), annot = True, cmap = "Blues")
plt.show()
```

Next, I will explore each attribute and then explore the found correlations.### Age Exploration

```
# Display age distribution

df['age'].plot(kind = 'hist', title = 'Age Distribution')
plt.show()

# Get min, max and average of the age
print('Min age: ', min(df['age']))

print('Max age: ', max(df['age']))



# Display age distribution based on heart disease sns.distplot(df[df['target']
== 1]['age'], label='Do not have heart disease')sns.distplot(df[df['target'] ==
2]['age'], label = 'Have heart disease') plt.xlabel('Frequency')

plt.ylabel('Age')

plt.title('Age Distribution based on Heart Disease')
plt.legend()

plt.show()

# Get min, max and average of the age of the people do not have heart diseas
print('Min age of people who do not have heart disease: ', min(df[df['target'] ==
1]['age']))
```

```
print('Max age of people who do not have heart disease: ', max(df[df['target'] ==
1]['age']))

print('Average age of people who do not have heart disease: ', df[df['target'] ==
1]['age'].mean())

# Get min, max and average of the age of the people have heart diseas

print('Min age of people who have heart disease: ', min(df[df['target'] == 2]['age'])) print('Max
age of people who have heart disease: ', max(df[df['target'] == 2]['age']))

print('Average age of people who have heart disease: ', df[df['target'] ==
2]['age'].mean())
```

Note:

From the data, I can say that the heart disease infects the old and young people, andthe probability of the old people te be infected is higher than young people.

```
### Gender Exploration

# Number of males and females

F = df[df['sex'] == 0].count()['target']

M = df[df['sex'] == 1].count()['target']



# Create a plot

figure, ax = plt.subplots(figsize = (6, 4)) ax.bar(x
= ['Female', 'Male'], height = [F, M])
plt.xlabel('Gender')

plt.title('Number of Males and Females in the dataset')
plt.show()

### Chest Pain Type Exploration

# Display chest pain types in bar chart
```

```
df.groupby(df['cp']).count()['target'].plot(kind = 'bar', title = 'Chest Pain Types', figsize

= (8, 6))

plt.xlabel('Chest Pain Types')

plt.xticks(np.arange(4), ('typical angina', 'atypical angina', 'non-anginal pain',
'asymptomatic'), rotation = 0)

plt.show()

# Display chest pain types based on the target
pd.crosstab(df.cp,df.target).plot(kind = "bar", figsize = (8, 6))
plt.title('Heart Disease Frequency According to Chest Pain Type')
plt.xlabel('Chest Pain Type')


plt.xticks(np.arange(4), ('typical angina', 'atypical angina', 'non-anginal pain',
'asymptomatic'), rotation = 0)

plt.ylabel('Frequency')
plt.show()
```

```
### Blood Pressure Exploration

# Display blood pressure distribution

df['trestbps'].plot(kind = 'hist', title = 'Blood Pressure in mm Hg', figsize = (8, 6))
plt.show()

# Display blood pressure distribution based on heart disease
fig, (axis1, axis2) = plt.subplots(1, 2,figsize=(25, 5))

ax = sns.distplot(df[df['target'] == 1]['trestbps'], label='Do not have heart disease', ax =axis1)

ax.set(xlabel='People Do Not Have Heart Disease')

ax = sns.distplot(df[df['target'] == 2]['trestbps'], label = 'Have heart disease', ax =
axis2)

ax.set(xlabel='People Have Heart Disease')
plt.show()

# Get min, max and average of the blood pressure of the people do not have heart
diseas

print('Min blood pressure of people who do not have heart disease: ', min(df[df['target']

== 1]['trestbps']))

print('Max blood pressure of people who do not have heart disease: ',
max(df[df['target'] == 1]['trestbps']))

print('Average blood pressure of people who do not have heart disease: ', df[df['target']

== 1]['trestbps'].mean())

# Get min, max and average of the blood pressure of the people have heart diseas
print('Min blood pressure of people who have heart disease: ', min(df[df['target'] ==
2]['trestbps']))

print('Max blood pressure of people who have heart disease: ', max(df[df['target'] ==
2]['trestbps']))

print('Average blood pressure of people who have heart disease: ', df[df['target'] ==
2]['trestbps'].mean())

### Cholestoral Exploration

# Display Cholestoral distribution

df['chol'].plot(kind = 'hist', title = 'Serum Cholestoral in mg/dl', figsize = (8, 6))
plt.show()

# Display Cholestoral distribution based on heart disease



fig, (axis1, axis2) = plt.subplots(1, 2, figsize=(25, 5))

ax = sns.distplot(df[df['target'] == 1]['chol'], label='Do not have heart disease', ax =axis1)

ax.set(xlabel='People Do Not Have Heart Disease')

ax = sns.distplot(df[df['target'] == 2]['chol'], label = 'Have heart disease', ax = axis2)
ax.set(xlabel='People Have Heart Disease')

plt.show()

# Get min, max and average of the Cholestoral of the people do not have heart diseas
print('Min cholestoral of people who do not have heart disease: ', min(df[df['target'] ==
1]['chol']))

print('Max cholestoral of people who do not have heart disease: ', max(df[df['target']

== 1]['chol']))

print('Average cholestoral of people who do not have heart disease: ', df[df['target'] ==
1]['chol'].mean())

# Get min, max and average of the Cholestoral of the people have heart diseas
print('Min cholestoral of people who have heart disease: ', min(df[df['target'] ==
2]['chol']))
```

```
print('Max cholestoral of people who have heart disease: ', max(df[df['target'] ==
2]['chol']))

print('Average cholestorale of people who have heart disease: ', df[df['target'] ==
2]['chol'].mean())

### Fasting Blood Sugar  Exploration

# Display fasting blood sugar in bar chart df.groupby(df['fbs']).count()['target'].plot(kind =
'bar', title = 'Fasting Blood Sugar',figsize = (8, 6))

plt.xticks(np.arange(2), ('fbs < 120 mg/dl', 'fbs > 120 mg/dl'), rotation = 0)
plt.show()

# Display fasting blood sugar based on the target


pd.crosstab(df.fbs,df.target).plot(kind = "bar", figsize = (8, 6))
plt.title('Heart Disease Frequency According to Fasting Blood Sugar')
plt.xlabel('Fasting Blood Sugar')

plt.xticks(np.arange(2), ('fbs < 120 mg/dl', 'fbs > 120 mg/dl'), rotation = 0)
plt.ylabel('Frequency')

plt.show()

###  Electrocardiographic Results Exploration

# Display electrocardiographic results in bar chart
df.groupby(df['restecg']).count()['target'].plot(kind = 'bar', title = 'Resting
Electrocardiographic Results', figsize = (8, 6))

plt.xticks(np.arange(3), ('normal', 'ST-T wave abnormality', 'probable or left
ventricular hypertrophy'))
plt.show()

# Display resting electrocardiographic results based on the target
pd.crosstab(df.restecg,df.target).plot(kind = "bar", figsize = (8, 6))

plt.title('Heart Disease Frequency According to Resting Electrocardiographic Results')
plt.xticks(np.arange(3), ('normal', 'ST-T wave abnormality', 'probable or left ventricular
hypertrophy'))

plt.xlabel('Resting Electrocardiographic Results')
plt.ylabel('Frequency')

plt.show()



### Maximum Heart Rate Exploration

# Display maximum heart rate distribution

df['thalach'].plot(kind = 'hist', title = 'Maximum Heart Rate Achieved', figsize = (8, 6))
plt.show()

# Display maximum heart rate distribution based on heart diseasefig,
(axis1, axis2) = plt.subplots(1, 2, figsize=(25, 5))

ax = sns.distplot(df[df['target'] == 1]['thalach'], label='Do not have heart disease', ax =axis1)

ax.set(xlabel = 'People Do Not Have Heart Disease')

ax = sns.distplot(df[df['target'] == 2]['thalach'], label = 'Have heart disease', ax = axis2)
ax.set(xlabel = 'People Have Heart Disease')

plt.show()

# Get min, max and average of the maximum heart rate of the people do not have heart
diseas

print('Min resting blood pressure of people who do not have heart disease: ',
min(df[df['target'] == 1]['thalach']))
```

```
print('Max resting blood pressure of people who do not have heart disease: ',
max(df[df['target'] == 1]['thalach']))

print('Average resting blood pressure of people who do not have heart disease: ',
df[df['target'] == 1]['thalach'].mean())

# Get min, max and average of the maximum heart rate of the people have heart diseas
print('Min maximum heart rate  of people who have heart disease: ', min(df[df['target']

== 2]['thalach']))

print('Max maximum heart rate people who have heart disease: ', max(df[df['target'] ==
2]['thalach']))

print('Average maximum heart rate of people who have heart disease: ', df[df['target']

== 2]['thalach'].mean())
```

```
### Exercise Induced Angina Exploration

# Display exercise induced angina in bar chart
df.groupby(df['exang']).count()['target'].plot(kind = 'bar', title = 'Exercise Induced
Angina',  figsize = (8, 6))

plt.xticks(np.arange(2), ('No', 'Yes'), rotation = 0)
plt.show()

# Display exercise induced angina based on the target
pd.crosstab(df.exang,df.target).plot(kind = "bar", figsize = (8, 6)) plt.title('Heart
Disease Frequency According to Exercise Induced Angina')plt.xlabel('Exercise
Induced Angina')

plt.xticks(np.arange(2), ('No', 'Yes'), rotation = 0)
plt.ylabel('Frequency')

plt.show()



### ST depression Exploration

# Display ST depression induced by exercise relative to rest distribution
df['oldpeak'].plot(kind = 'hist', title = 'ST Depression Induced by Exercise Relative toRest',
figsize = (8, 6))

plt.show()

# Display ST depression distribution based on heart disease
fig, (axis1, axis2) = plt.subplots(1, 2, figsize=(25, 5))

ax = sns.distplot(df[df['target'] == 1]['oldpeak'], label='Do not have heart disease', ax =axis1)

ax.set(xlabel = 'People Do Not Have Heart Disease')

ax = sns.distplot(df[df['target'] == 2]['oldpeak'], label = 'Have heart disease', ax =axis2)

ax.set(xlabel = 'People Have Heart Disease')
plt.show()
```

```
# Get min, max and average of the ST depression of the people have heart diseas print('Min
ST depression of people who do not have heart disease: ', min(df[df['target']

== 1]['oldpeak']))

print('Max ST depression of people who do not have heart disease: ',
max(df[df['target'] == 1]['oldpeak']))

print('Average ST depression of people who do not have heart disease: ', df[df['target']

== 1]['oldpeak'].mean())

# Get min, max and average of the ST depression of the people have heart diseas
print('Min ST depression of people who have heart disease: ', min(df[df['target'] ==
2]['oldpeak']))


print('Max ST depression of people who have heart disease: ', max(df[df['target'] ==
2]['oldpeak']))

print('Average ST depression of people not have heart disease: ', df[df['target'] ==
2]['oldpeak'].mean())
```

```
### Slope Exploration

# Display slope of the peak exercise ST segment in bar chart
df.groupby(df['slope']).count()['target'].plot(kind = 'bar', title = 'Slope of the Peak
Exercise ST Segment', figsize = (8, 6))

plt.xticks(np.arange(3), ('upsloping', 'flat', 'downsloping'), rotation =
0)plt.show()

# Display slope of the peak exercise ST segment based on the target
pd.crosstab(df.slope,df.target).plot(kind = "bar", figsize = (8, 6))

plt.title('Heart Disease Frequency According to Slope of the Peak Exercise
STSegment')

plt.xlabel('Slope')

plt.xticks(np.arange(3), ('upsloping', 'flat', 'downsloping'), rotation =
0)plt.ylabel('Frequency')

plt.show()



### Major Vessels Exploration

# Display number of major vessels in bar chart
df.groupby(df['ca']).count()['target'].plot(kind = 'bar', title = 'Number of Major Vessels
Colored by Flourosopy',

                                    figsize = (8, 6))

plt.show()

# Display number of vessels based on the target
```

```
pd.crosstab(df.ca,df.target).plot(kind = "bar", figsize = (8, 6))

plt.title('Heart Disease Frequency According to Number of Major Vessels Colored
byFlourosopy')

plt.xlabel('number of
vessels')
plt.xticks(rotation = 0)
plt.ylabel('Frequency')
plt.show()

vessels colored by
flourosopy.###
Thalassemia Exploration

# Display thalassemia in bar chart
df.groupby(df['thal']).count()['target'].plot(kind = 'bar', title = 'Thalassemia')

plt.xticks(np.arange(3), ('normal', 'fixed defect', 'reversible defect'), rotation =
0)plt.show()

pd.crosstab(df.thal,df.target).plot(kind = "bar", figsize = (8,
6))plt.title('Heart Disease Frequency According to
Thalassemia') plt.xlabel('Thalassemia')

plt.xticks(np.arange(3), ('normal', 'fixed defect', 'reversible defect'), rotation =
0)plt.ylabel('Frequency')

plt.show()



### The correlation between heart disease, cp and exang

g = sns.factorplot("cp", col = "exang", col_wrap = 3, data = df[df['target'] == 1], kind
= "count")

plt.xticks(np.arange(4), ('typical angina', 'atypical angina', 'non-anginal
pain','asymptomatic'), rotation = 0)

g.fig.suptitle('People without Heart Disease', y =
1.1)plt.show()

g = sns.factorplot("cp", col = "exang", col_wrap = 3, data = df[df['target'] == 2], kind

= "count")

plt.xticks(np.arange(4), ('typical angina', 'atypical angina', 'non-anginal
pain','asymptomatic'), rotation = 0)

g.fig.suptitle('People with Heart Disease', y =
1.1)plt.show()


### The correlation between oldpeak, slope and target
sns.catplot(x = "slope", y = "oldpeak", hue = "target", data =
df)plt.title('The correlation between oldpeak and slope')

plt.xticks(np.arange(3), ('upsloping', 'flat', 'downsloping'), rotation =
0)plt.show()
```

Flat slope and downsloping have higher values of ST
depression.### The correlation between ca and age

```
g = sns.catplot(x = 'ca', y = 'age', hue = 'target', data = df, kind="swarm")
g.fig.suptitle('The correlation between number of major vessels colored by
flourosopyand age', y = 1.1)
```

```
plt.show()

### The correlation between age and thalach

sns.relplot(x = 'age', y = 'thalach', data = df, hue = 'target',
legend="full")plt.title('The correlation between age and heart rate')

plt.show()


## 3. Modeling

### 3.1. Prepare Data for Machine
Learning# Initialize data and target

target = df['target']

features = df.drop(['target'], axis = 1)

# Split the data into training set and testing set

X_train, X_test, y_train, y_test = train_test_split(features, target, test_size =
0.2,random_state = 0)
```

Here, I will use the below machine learning algorithms then I will select the best onebased on its classification report.

* Support Vector Machine

* Random Forest

* Ada Boost

* Gradient Boosting

```
### 3.2. Train and Evaluate
Models# Train and evaluate
model

def fit_eval_model(model, train_features, y_train, test_features, y_test):


    """

    Function: train and evaluate a machine learning
    classifier.Args:

      model: machine learning classifier
      train_features: train data extracted
      featuresy_train: train data lables

      test_features: train data extracted
      featuresy_test: train data lables

    Return:

      results(dictionary): a dictionary of classification
    report"""

    results = {}


    # Train the model

    model.fit(train_features, y_train)
    # Test the model

    train_predicted =
    model.predict(train_features)
    test_predicted =
    model.predict(test_features)



     # Classification report and Confusion Matrix
    results['classification_report'] = classification_report(y_test,
    test_predicted)results['confusion_matrix'] = confusion_matrix(y_test,
    test_predicted)



    return results


# Initialize the models

sv = SVC(random_state = 1)

rf = RandomForestClassifier(random_state
= 1)ab = AdaBoostClassifier(random_state
= 1)

gb = GradientBoostingClassifier(random_state = 1)




# Fit and evaluate
```

```
modelsresults = {}

for cls in [sv, rf, ab, gb]:

    cls_name = cls.___class__.
    ________________name__
    results[cls_name] = {}

    results[cls_name] = fit_eval_model(cls, X_train, y_train, X_test, y_test)



# Print classifiers
resultsfor result in
results:

    print
    (result
    )
    print()

    for i in
        results[result]:
        print (i, ':')
        print(results[resu
        lt][i])print()

    print ('------ ')

    print()

#### Selecting the Classifier

### 3.3. The Important
Features# get importance
```

importance =
gb.feature_importances_#
summarize feature importance

for i,v in enumerate(importance):

    print('Feature: %s, Score: %.5f' % (features.columns[i],
v))# plot feature importance

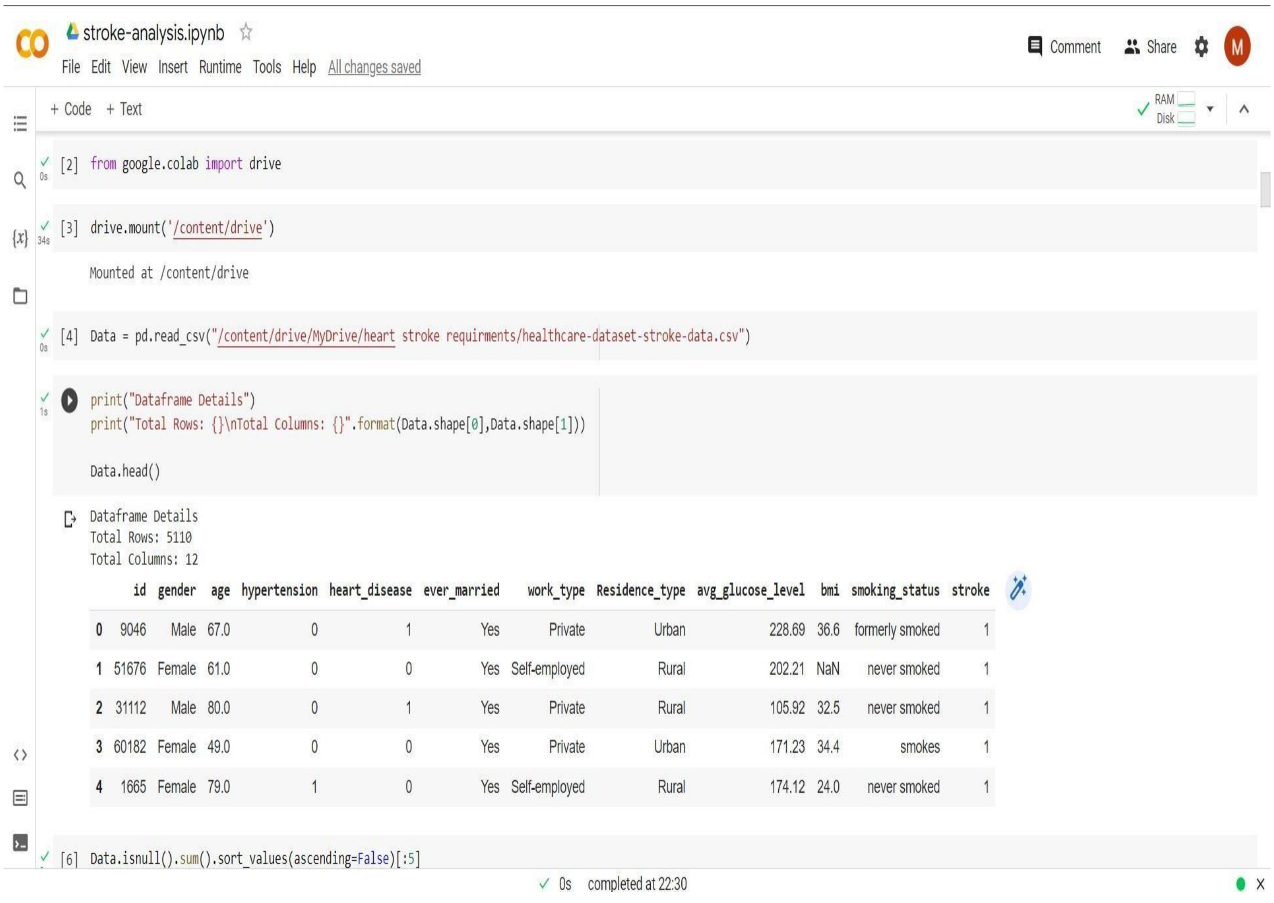
plt.bar([x for x in range(len(importance))],
importance)plt.show()

### 3.4. Save Model

Finally, I will save the GradientBoostingClassifier model to use it later.# Save the model as serialized object pickle

with open('model.pkl', 'wb') as
    file:pickle.dump(gb, file)

B. SCREENSHOTS



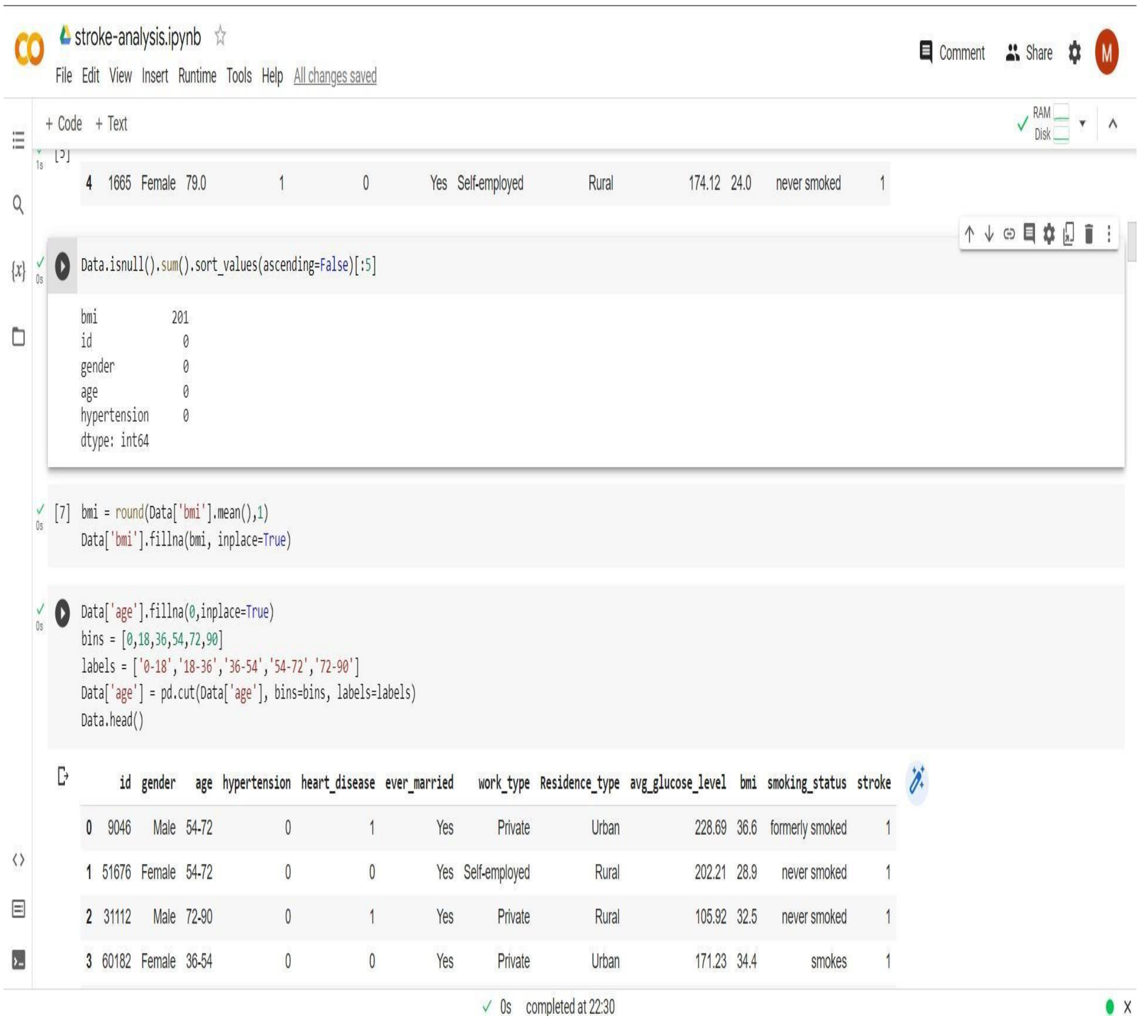
```
[2] from google.colab import drive

[3] drive.mount('/content/drive')

Mounted at /content/drive

[4] Data = pd.read_csv("/content/drive/MyDrive/heart stroke requirments/healthcare-dataset-stroke-data.csv")

print("Dataframe Details")
print("Total Rows: {}\nTotal Columns: {}".format(Data.shape[0],Data.shape[1]))

Data.head()

Dataframe Details
Total Rows: 5110
Total Columns: 12
```

	id	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	smoking_status	stroke
0	9046	Male	67.0	0	1	Yes	Private	Urban	228.69	36.6	formerly smoked	1
1	51676	Female	61.0	0	0	Yes	Self-employed	Rural	202.21	NaN	never smoked	1
2	31112	Male	80.0	0	1	Yes	Private	Rural	105.92	32.5	never smoked	1
3	60182	Female	49.0	0	0	Yes	Private	Urban	171.23	34.4	smokes	1
4	1665	Female	79.0	1	0	Yes	Self-employed	Rural	174.12	24.0	never smoked	1

```
[6] Data.isnull().sum().sort_values(ascending=False)[:5]
```

✓ 0s completed at 22:30

Fig -B1 - implementation of code



```
4  1665  Female  79.0    1    0    Yes  Self-employed    Rural    174.12  24.0    never smoked    1
```

```
Data.isnull().sum().sort_values(ascending=False)[:5]
```

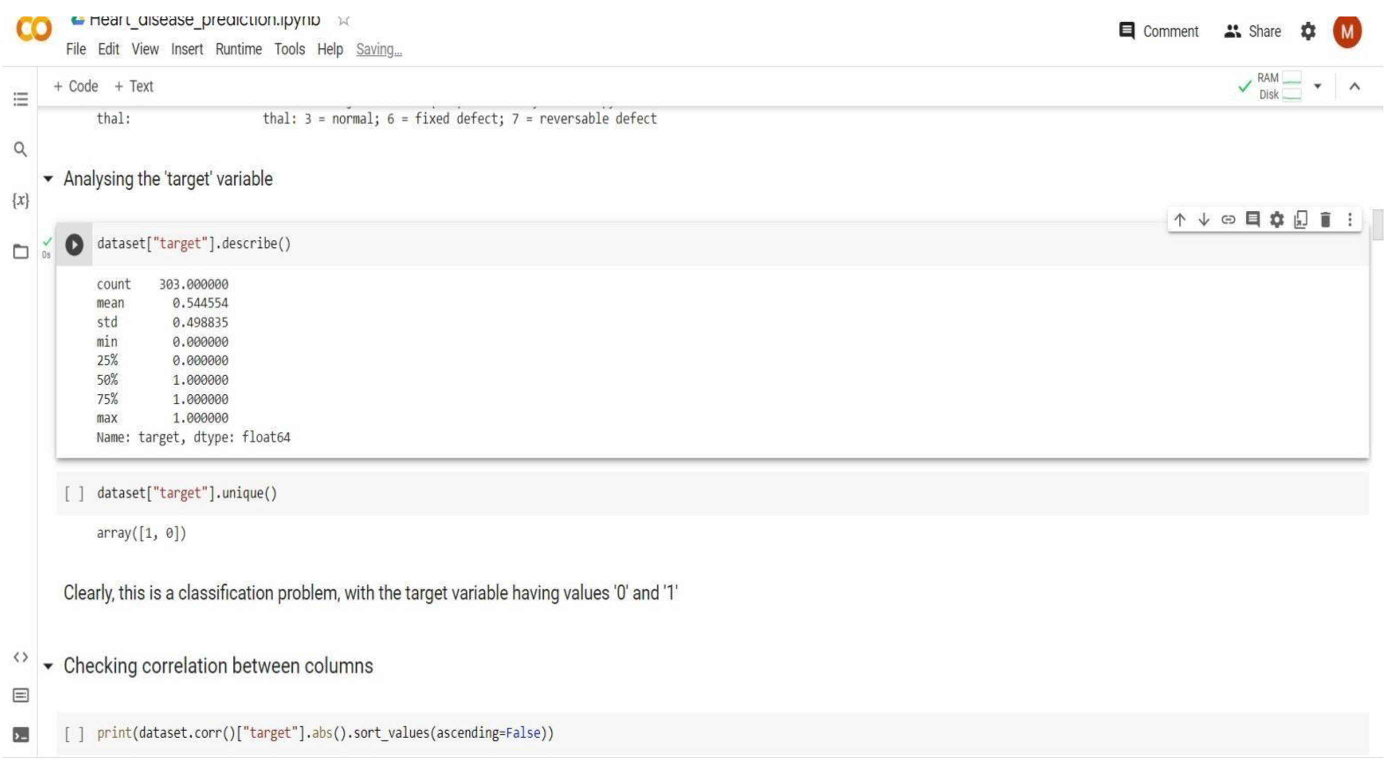
```
bmi             201
id                0
gender            0
age               0
hypertension      0
dtype: int64
```

```
[7] bmi = round(Data['bmi'].mean(),1)
    Data['bmi'].fillna(bmi, inplace=True)
```

```
Data['age'].fillna(0,inplace=True)
bins = [0,18,36,54,72,90]
labels = ['0-18','18-36','36-54','54-72','72-90']
Data['age'] = pd.cut(Data['age'], bins=bins, labels=labels)
Data.head()
```

	id	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	smoking_status	stroke
0	9046	Male	54-72	0	1	Yes	Private	Urban	228.69	36.6	formerly smoked	1
1	51676	Female	54-72	0	0	Yes	Self-employed	Rural	202.21	28.9	never smoked	1
2	31112	Male	72-90	0	1	Yes	Private	Rural	105.92	32.5	never smoked	1
3	60182	Female	36-54	0	0	Yes	Private	Urban	171.23	34.4	smokes	1

Fig -B2 - data sets and attributes


Heart_disease_prediction.ipynb
File Edit View Insert Runtime Tools Help Saving...
Comment
Share
+ Code + Text
RAM
Disk

```
thal:                  thal: 3 = normal; 6 = fixed defect; 7 = reversable defect
```

Analysing the 'target' variable

```
dataset["target"].describe()
```

```
count    303.000000
mean       0.544554
std        0.498835
min        0.000000
25%        0.000000
50%        1.000000
75%        1.000000
max        1.000000
Name: target, dtype: float64
```

```
dataset["target"].unique()
```

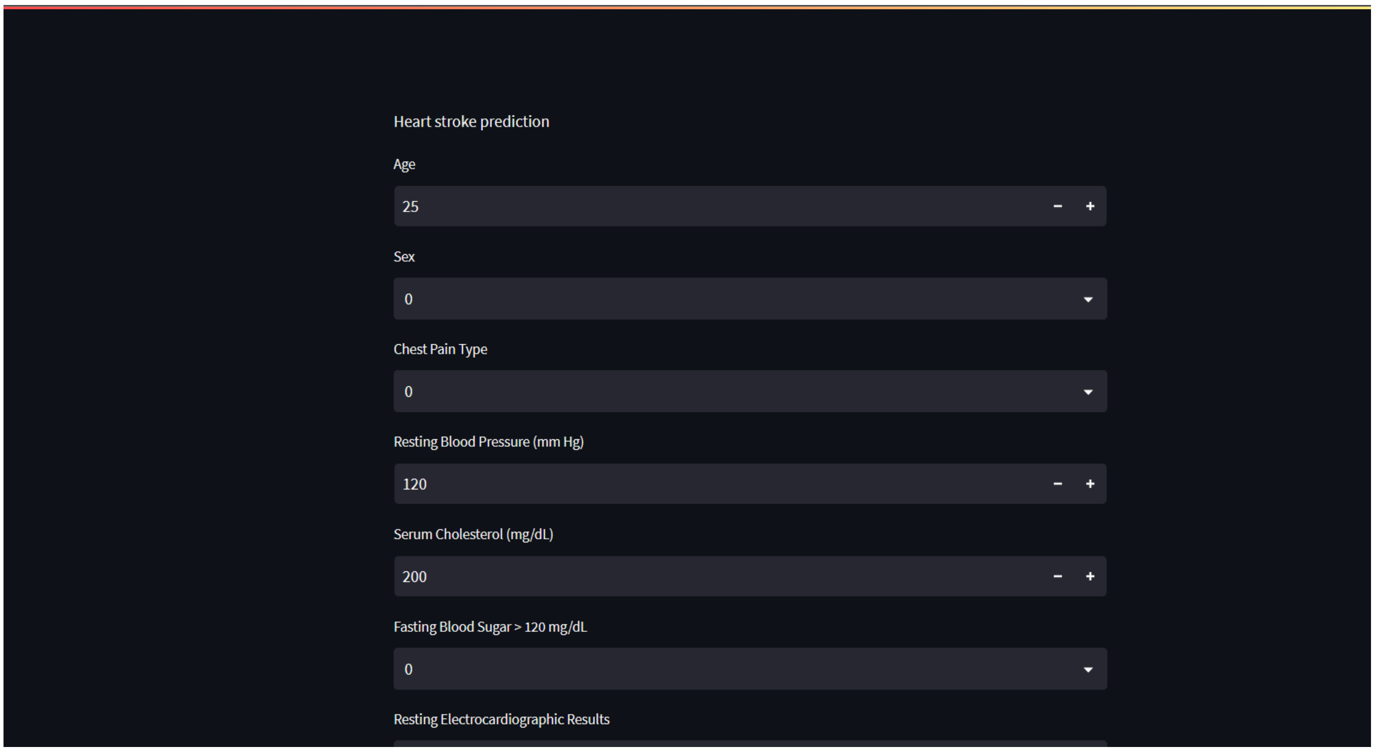
```
array([1, 0])
```

Clearly, this is a classification problem, with the target variable having values '0' and '1'

Checking correlation between columns

```
print(dataset.corr()["target"].abs().sort_values(ascending=False))
```

Fig -B3 - target values


Heart stroke prediction

Age

25

Sex

0

Chest Pain Type

0

Resting Blood Pressure (mm Hg)

120

Serum Cholesterol (mg/dL)

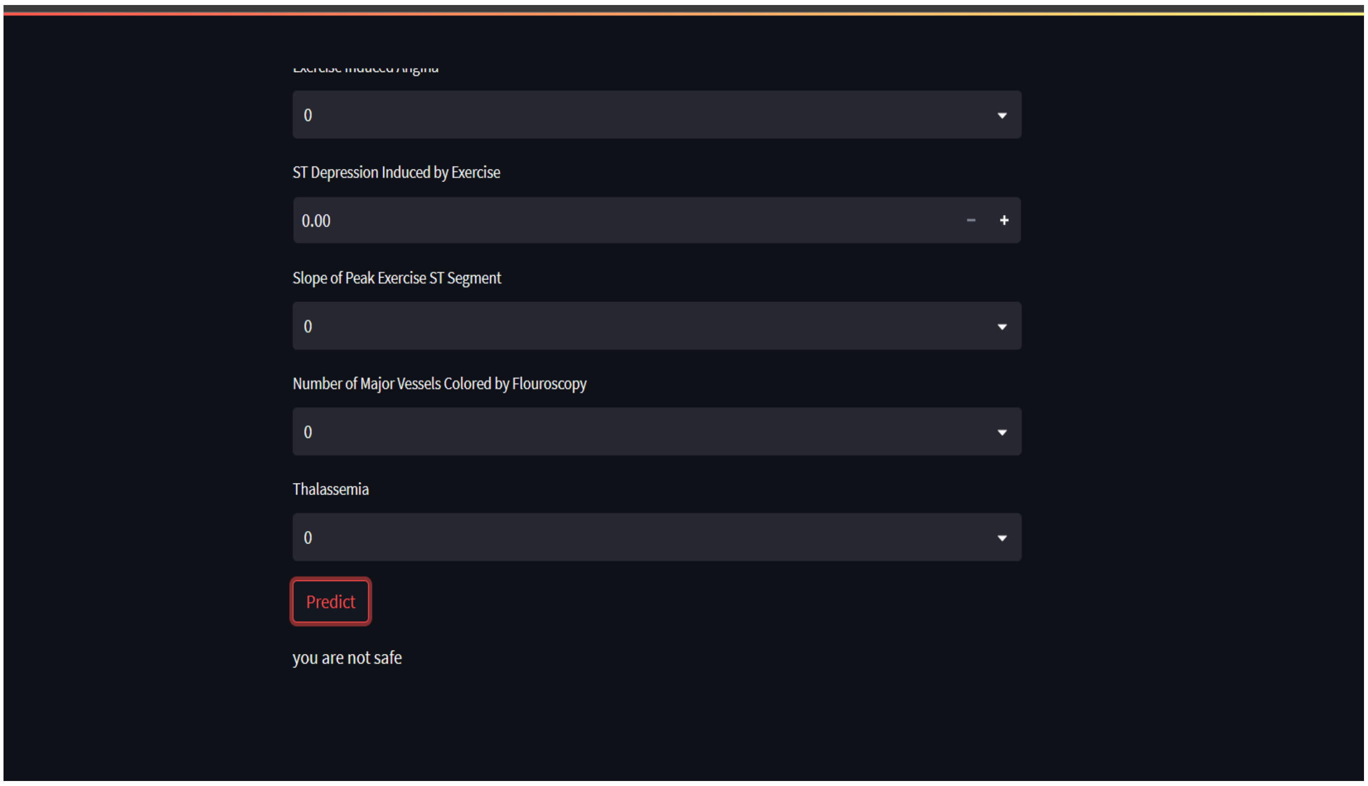
200

Fasting Blood Sugar > 120 mg/dL

0

Resting Electrocardiographic Results

Fig - B4-Getting input data



ST Depression Induced by Exercise

0.00

Slope of Peak Exercise ST Segment

0

Number of Major Vessels Colored by Flouroscopy

0

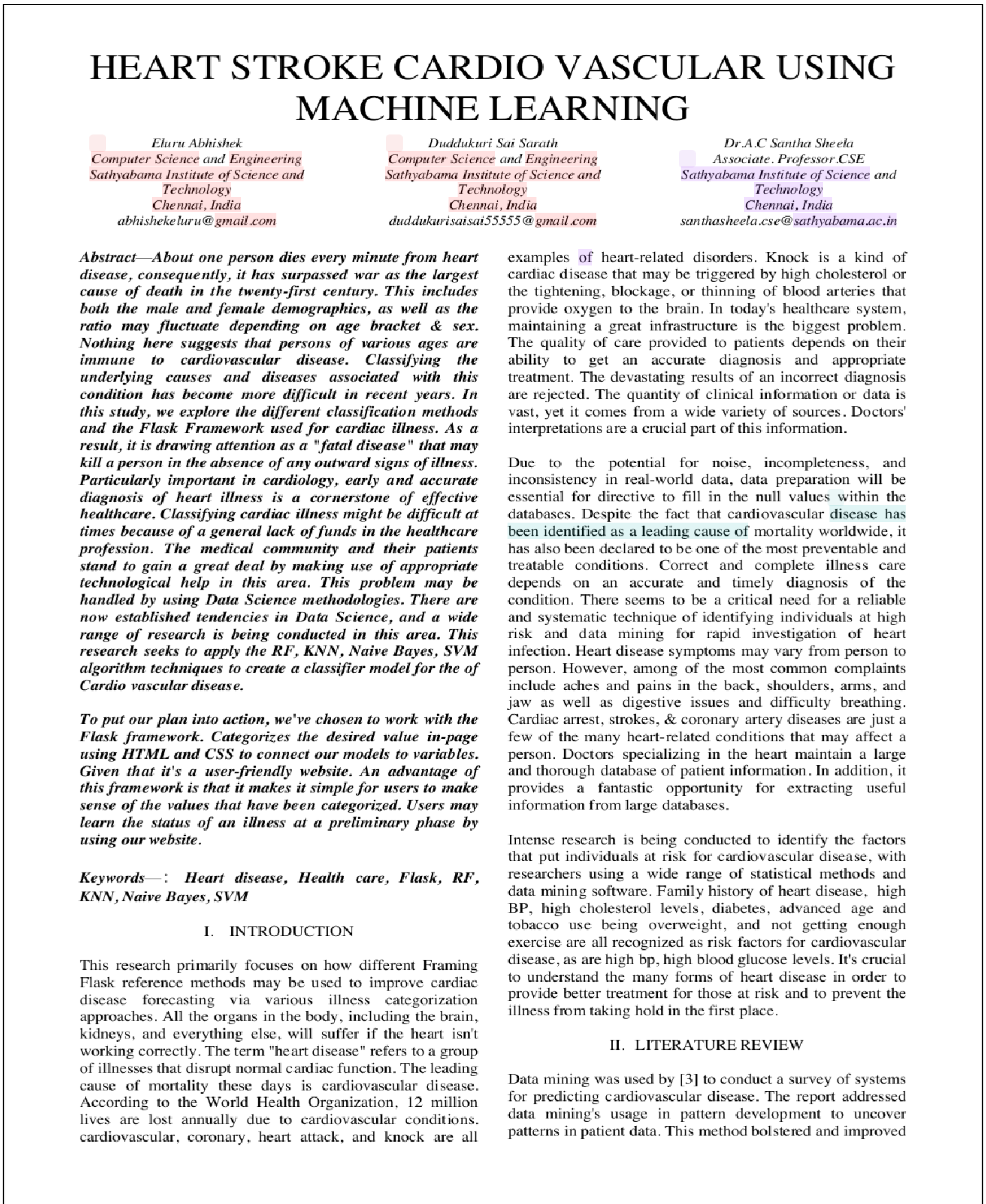
Thalassemia

0

you are not safe

Fig-B5- output

C. RESEARCH PAPER



HEART STROKE CARDIO VASCULAR USING MACHINE LEARNING

Eluru Abhishek
Computer Science and Engineering
Sathyabama Institute of Science and Technology
Chennai, India
abhishekeluru@gmail.com

Duddukuri Sai Sarath
Computer Science and Engineering
Sathyabama Institute of Science and Technology
Chennai, India
duddukurisaisai55555@gmail.com

Dr.A.C Santha Sheela
Associate. Professor.CSE
Sathyabama Institute of Science and Technology
Chennai, India
santhasheela.cse@sathyabama.ac.in

Abstract*—About one person dies every minute from heart disease, consequently, it has surpassed war as the largest cause of death in the twenty-first century. This includes both the male and female demographics, as well as the ratio may fluctuate depending on age bracket & sex. Nothing here suggests that persons of various ages are immune to cardiovascular disease. Classifying the underlying causes and diseases associated with this condition has become more difficult in recent years. In this study, we explore the different classification methods and the Flask Framework used for cardiac illness. As a result, it is drawing attention as a "fatal disease" that may kill a person in the absence of any outward signs of illness. Particularly important in cardiology, early and accurate diagnosis of heart illness is a cornerstone of effective healthcare. Classifying cardiac illness might be difficult at times because of a general lack of funds in the healthcare profession. The medical community and their patients stand to gain a great deal by making use of appropriate technological help in this area. This problem may be handled by using Data Science methodologies. There are now established tendencies in Data Science, and a wide range of research is being conducted in this area. This research seeks to apply the RF, KNN, Naive Bayes, SVM algorithm techniques to create a classifier model for the of Cardio vascular disease.

To put our plan into action, we've chosen to work with the Flask framework. Categorizes the desired value in-page using HTML and CSS to connect our models to variables. Given that it's a user-friendly website. An advantage of this framework is that it makes it simple for users to make sense of the values that have been categorized. Users may learn the status of an illness at a preliminary phase by using our website.

***Keywords*—: Heart disease, Health care, Flask, RF, KNN, Naive Bayes, SVM**

I. INTRODUCTION

This research primarily focuses on how different Framing Flask reference methods may be used to improve cardiac disease forecasting via various illness categorization approaches. All the organs in the body, including the brain, kidneys, and everything else, will suffer if the heart isn't working correctly. The term "heart disease" refers to a group of illnesses that disrupt normal cardiac function. The leading cause of mortality these days is cardiovascular disease. According to the World Health Organization, 12 million lives are lost annually due to cardiovascular conditions. cardiovascular, coronary, heart attack, and knock are all examples of heart-related disorders. Knock is a kind of cardiac disease that may be triggered by high cholesterol or the tightening, blockage, or thinning of blood arteries that provide oxygen to the brain. In today's healthcare system, maintaining a great infrastructure is the biggest problem. The quality of care provided to patients depends on their ability to get an accurate diagnosis and appropriate treatment. The devastating results of an incorrect diagnosis are rejected. The quantity of clinical information or data is vast, yet it comes from a wide variety of sources. Doctors' interpretations are a crucial part of this information.

Due to the potential for noise, incompleteness, and inconsistency in real-world data, data preparation will be essential for directive to fill in the null values within the databases. Despite the fact that cardiovascular disease has been identified as a leading cause of mortality worldwide, it has also been declared to be one of the most preventable and treatable conditions. Correct and complete illness care depends on an accurate and timely diagnosis of the condition. There seems to be a critical need for a reliable and systematic technique of identifying individuals at high risk and data mining for rapid investigation of heart infection. Heart disease symptoms may vary from person to person. However, among of the most common complaints include aches and pains in the back, shoulders, arms, and jaw as well as digestive issues and difficulty breathing. Cardiac arrest, strokes, & coronary artery diseases are just a few of the many heart-related conditions that may affect a person. Doctors specializing in the heart maintain a large and thorough database of patient information. In addition, it provides a fantastic opportunity for extracting useful information from large databases.

Intense research is being conducted to identify the factors that put individuals at risk for cardiovascular disease, with researchers using a wide range of statistical methods and data mining software. Family history of heart disease, high BP, high cholesterol levels, diabetes, advanced age and tobacco use being overweight, and not getting enough exercise are all recognized as risk factors for cardiovascular disease, as are high bp, high blood glucose levels. It's crucial to understand the many forms of heart disease in order to provide better treatment for those at risk and to prevent the illness from taking hold in the first place.

II. LITERATURE REVIEW

Data mining was used by [3] to conduct a survey of systems for predicting cardiovascular disease. The report addressed data mining's usage in pattern development to uncover patterns in patient data. This method bolstered and improved their preexisting health care plan. Their described algorithms had a limited scope and were designed to predict cardiovascular illness. They developed an association rule based on a real informational index and the patients' cardiac health records, which resulted in a high rate of accuracy. The recommended calculation that they performed addresses not only the problem of a large number of principles, but also the proper approval of guidelines backings. To facilitate determination as a preliminary task in the therapeutic database characterization, Kernel F-score Feature Selection was presented. The suggested KFFS strategy first converts healthcare information to binary capacity by increasing BFF & LINEAR skills. Second, the F-score equation, piece capacities of a non-directly separable set of data are transformed into a straight distinct element space, allowing for the determination of treatment datasets [3, 4].

Data mining classification methods, including DT, NB, & NN, are examined in depth on a Cardio vascular disease database in a report titled "Enhanced Research of Heart Disease Diagnosis utilizing Data Mining Classification Techniques" [4]. Different research methods were evaluated for their precision of results. There was a one hundred percent success rate for Neural Networks, DT has a 99.62% successfulness, whereas Naive Bayes only has a 90% likelihood of success. According to their findings, Neural Networks are the most accurate predictors of heart disease among the three models. Lastly, they included two additional input variables, obesity and smoking status, to complete the model. Due to their significance in the diagnosis of cardiac disorders, these characteristics are applied to achieve more precise outcomes.

An IHDPS prototype was constructed with the use of data mining methods such DT, NB, & NN, as detailed in the article "Smart Forecasting of Cardiac Diseases Machine Through Data Mining Methods." The outcomes demonstrate the distinct value that each method brings to the table in accomplishing the predetermined mining goals. Whereas conventional decision support systems are unable to address such nuanced "what if" concerns, IHDPS is well-suited to do so. Predicting a patient's risk of developing heart disease based on demographic information including age, sex, BP, and glucose levels. It permits the establishment of important information, such as patterns and correlations between medical variables associated with heart disease [5].

Studying how to enhance the precision of weaker methods by merging numerous classifiers, the authors of the publication "Improving the reliability of forecast of risk for heart disease based upon ensembles classifiers" conducted research utilizing this strategy. The tool was tested using data on cardiovascular disease. The ensemble technique's potential for enhancing prediction accuracy in cardiovascular illnesses was investigated using a comparative analytical strategy.

Conclusions from these studies suggest that ensemble methods like bagging and boosting may significantly increase the prediction performance of underperforming classifiers and provide respectable results when it comes to gauging the likelihood that a certain individual will develop cardiovascular disease. With the aid of the ensemble classification, the accuracy of weak classifiers was improved by as much as 7%. Prediction accuracy was significantly increased once a feature selection implementation was added to the procedure [6].

Frames Recovery, ROI Picking, Feature Extraction, and, SVM Classifier were all suggested by Sneha (2018) ([7]) for the study of Echocardiograms, whereby an input video is given into the system and subsequently utilized for training. At 97.30%, the system's accuracy is comprised of DCM at 97.40%, ASD at 97.57%, and regular condition at 99.53 percent.

Using techniques such as boosted, SVM, K-NN, naive bayes classification, J48, & rf, Martin (2017) ([8]) provide a method for diagnosing heart failure that persists over time. Unconventional method using LOSO validating set evaluation yields 97.82% reliability for the models.

The Fast Fourier Transformation is used to analyze time series and was suggested by Raid (2017) ([9]) as part of a MATLAB-based system for recommending treatments for cardiovascular illness. Together with this, we use an Ensemble model that combines bagging, a synthetic neural network, ordinary least support vector machines, and a naive bayes classifier.

III. PROPOSED SYSTEM

The proposed method incorporates data that might be used to diagnose heart illness in a patient. The suggested system may utilize this information to train a model to determine (through data reading and exploration) whether or not a given patient has the condition. The suggested system is best implemented using supervised (classification) techniques. Uses a set of optimizations learning to provide reliable output. Finally assessing the data with the assistance of Comparing Models using Confusion Matrix.
The SVM classifier's primary objective is to locate the max axis in an n-D space.

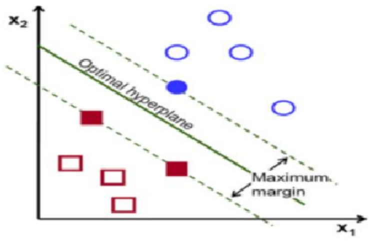


Fig.1. SVM classifier

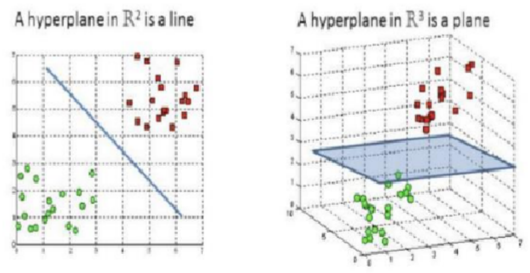


Fig.2. Hyperplane in 3D & 2D

What we have collected so far has to be organized into several types of data sets, each of which is specific to some aspect of the patient's heart. From the accessibility of this information, we have to establish a program that categories the patient illness utilizing supervised algorithms. Firstly, we just had to acquire all datasets. Read the dataset, the information should have several characteristics as sex, age, restBP, cp, cholesterol, FB, goal etc. The information should be investigated to ensure that the data is confirmed. In this case, the sigmoid function is used, which aids in the visual depiction of the labelled data. Supervised methods, such as the SVM, Naive Bayes, RF, and K-NN algorithms, enhance the accuracy. To put our plan into action, we've chosen to work with the Flask framework. We may show the outcomes on the website itself, which was built using HTML and CSS.

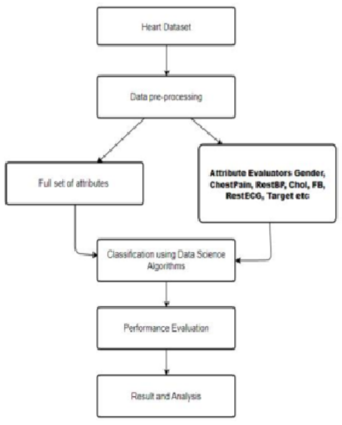

Heart Dataset
Data pre-processing
Full set of attributes
Attribute Evaluators Gender, ChestPain, RestBP, Chol, FB, RestECG, Target etc
Classification using Data Science Algorithms
Performance Evaluation
Result and Analysis

Fig.3. Process Model

IV. METHODOLOGY

The Rapid Application Development (RAD) framework has been used. Adaptive Rapid Development (RAD) approach is adaptive to alterations and to take in new information, including characteristics and functionalities, throughout the whole design and construction procedure. The Cardiac Dataset is used in this system, and it was obtained by use of Kaggle. The dataset has 14 different characteristics, by the results of 1025 tests. In this data collection, break apart into test and train group. To prepare for this information, we trained it here - three distinct machine algorithms, including K-Nearest Neighbor, SVM, Classification using Naive Bayes for Nearby Objects both the Classifier and the Random Forest Classifier are used. Following completion of such training, Putting each of the four hypotheses to rigorous models with the use of training methods, models with the most precise method of prediction was chosen.
This learned model is then downloaded into a server-based system.

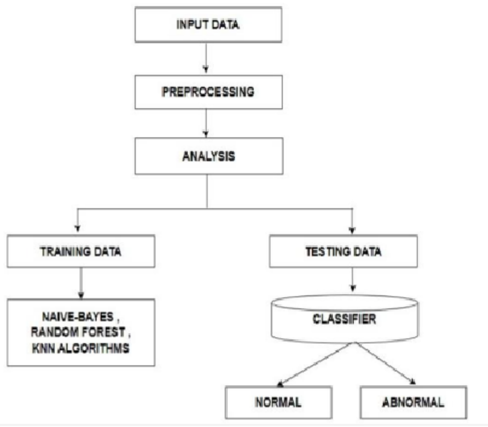


Fig.4. Architecture of our system

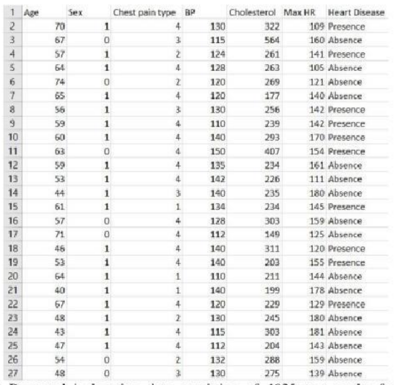
Flask that utilizes the (API) Application Programming Interface users will be able to fill up a form based on this training sample and their own inputs according to the outcomes of their tests. These parameters are currently being sent to the models may be checked and predicted for patients using the information they include whether or not having heart disease.

V. RESULTS AND DISCUSSIONS

In order to detect the likelihood that a user has heart disease, a ml model was developed in this research. For this machine learning model, the input data consists of 13 sets of human test results. During the processing and cleaning of this dataset, it was verified that no missing or invalid data values existed. The data set was broken down into two separate halves, x and y. Where the output (represented by the dependent variable y) is a function of the x parameter (which includes the 13 characteristics representing the various test outcomes). StandardScaler was used to transform the x-axis value. Additional categories were created for the x and y variables: y-test, x-test, x-train & y-train.

These y-train & x-train were learned with the aid of the supervised learning methods K-NN, NB, SVM, & RF. Using varying values for n, the four methods were utilized to determine the accuracy %. The maximum accuracy is achieved by K Neighbors at n = 1, by Naive Bayes at n = 1 by SVM at n=1, and by Random Forest at n = 10. After conducting extensive tests, we settled on using Naive Bayes because to its reputation as one among the most reliable forecasting models.

Flask, an Api, was used to save and upload the Naive Bayes model to the web. We constructed an HTML website with Thirteen variables using Flask so that people may submit the findings of their different tests and let the model assess the likelihood that they have cardiovascular disease. The model's results will be shown on a web-based portal for the patient's convenience.



1	Age	Sex	Chest pain type	BP	Cholesterol	Max HR	Heart Disease
2	70	1	4	130	322	109	Presence
3	67	0	3	115	564	160	Absence
4	57	1	2	124	261	141	Presence
5	64	1	4	128	263	105	Absence
6	74	0	2	120	269	121	Absence
7	65	1	4	120	177	140	Absence
8	56	1	3	130	256	142	Presence
9	59	1	4	110	239	142	Presence
10	60	1	4	140	293	170	Presence
11	63	0	4	150	407	154	Presence
12	59	1	4	135	234	161	Absence
13	53	1	4	142	226	111	Absence
14	44	1	3	140	235	180	Absence
15	61	1	1	134	234	145	Presence
16	57	0	4	128	303	159	Absence
17	71	0	4	112	149	125	Absence
18	46	1	4	140	311	120	Presence
19	53	1	4	140	203	155	Presence
20	64	1	1	110	211	144	Absence
21	40	1	1	140	199	178	Absence
22	67	1	4	120	229	129	Presence
23	48	1	2	130	245	180	Absence
24	43	1	4	115	303	181	Absence
25	47	1	4	112	204	143	Absence
26	54	0	2	132	288	159	Absence
27	48	0	3	130	275	139	Absence

Fig.5. Presented is learning data consisting of 1025 test results from individuals, fed to four distinct machine learning algorithms.

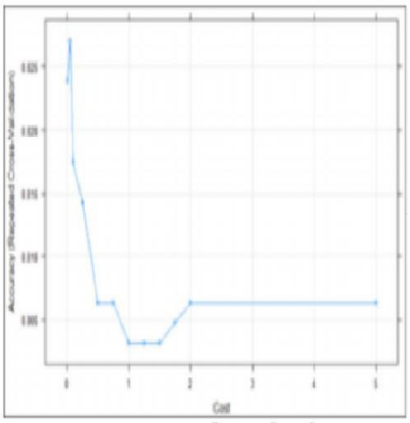


Fig.6. Accuracy of SVM

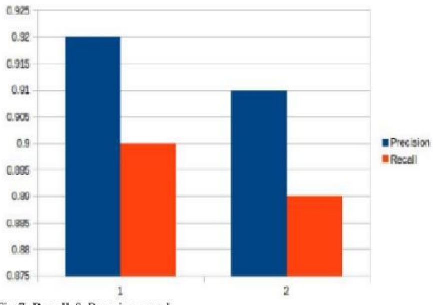


Fig.7. Recall & Pression graph

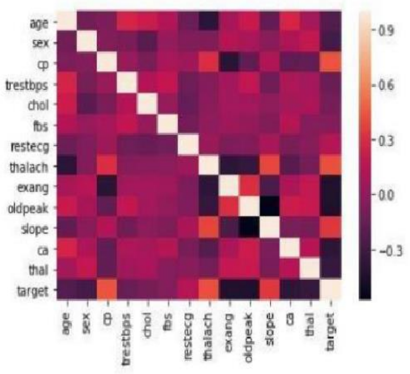


Fig.8. Scatter diagram displaying the data set's correlation matrix

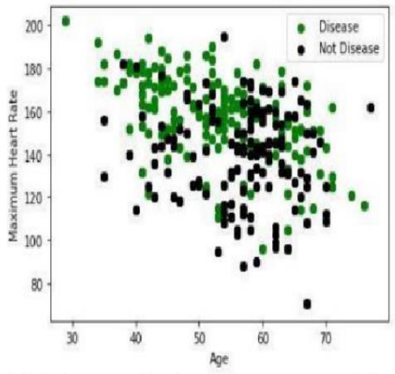


Fig.9. displaying a scatter plot of maximal heart rate vs age to indicate the presence or absence of cardiac disease.

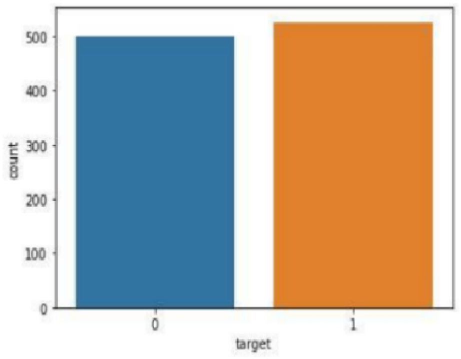


Fig.10. Histogram showing the percentage of the population with and without heart disease

```
Out[33]: Text(0.5, 1.0, 'K Nighbors Classification Score')
```

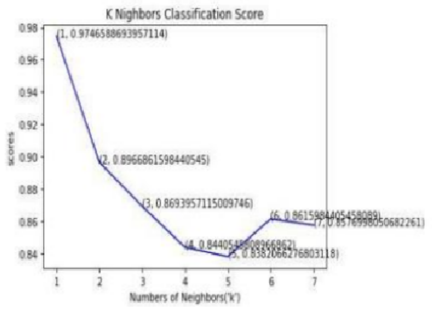


Fig.11. demonstrating precise K-Nearest Neighbor ratings from n = 1 to n = 7

```
Out[50]: Text(0.5, 1.0, 'Random Forest Classification Score')
```

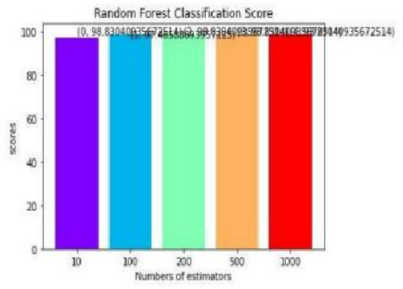
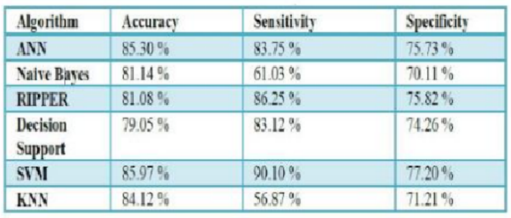
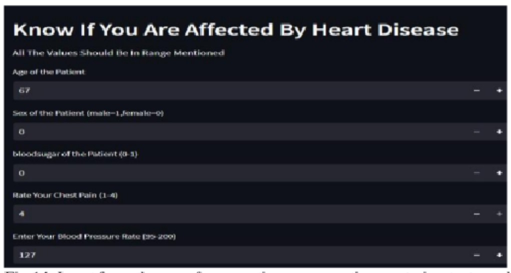


Fig.12. Demonstrating the reliability of RF with n = 10, 100, 200, 500, and 1000



Algorithm	Accuracy	Sensitivity	Specificity
ANN	85.30 %	83.75 %	75.73 %
Naïve Bayes	81.14 %	61.03 %	70.11 %
RIPPER	81.08 %	86.25 %	75.82 %
Decision Support	79.05 %	83.12 %	74.26 %
SVM	85.97 %	90.10 %	77.20 %
KNN	84.12 %	56.87 %	71.21 %

Fig.13. Assessing the performance of multiple ml techniques for forecasting heartdisease.



Know If You Are Affected By Heart Disease

All The Values Should Be In Range Mentioned

Age of the Patient: 67

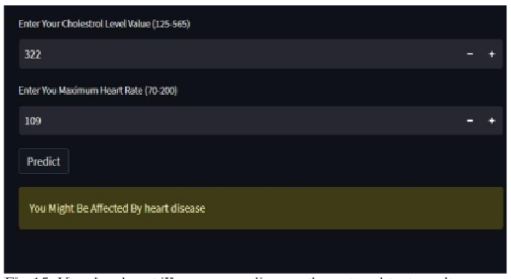
Sex of the Patient (male=1,female=0): 0

bloodsugar of the Patient (0-1): 0

Rate Your Chest Pain (1-4): 4

Enter Your Blood Pressure Rate (95-200): 127

Fig.14. Input from the user form continuance; continue entering test results



Enter Your Cholestrol Level Value (125-565): 322

Enter You Maximum Heart Rate (70-200): 109

You Might Be Affected By heart disease

Fig.15. User has heart illness, according on the entered test results

VI. SUMMARY AND PERSPECTIVES FOR THE FUTURE

The study, "A Model for Detecting Cardiovascular Disease using ML Algorithms," was created to use a ml strategy. The dataset provided for training and analysis in this machine learning technique includes test results from a variety of patients, as well as the effectiveness of the resulting algorithms was evaluated by producing a graph with matplotlib. The accuracy tests showed that SVM is the most reliable (around 86 percent), followed by ANN (roughly 85.30 percent). In addition, a web-based API termed Flask was used to include a Naive Bayes model, which accurately predicted outcomes across five web-based tests.When a real-time system built on Deep Learning is implemented, this study's findings may be applied to a practical setting in which users can submit their test results in the form of images.

REFERENCES

[1]. K. Vanisree, S. Jyothi, "Decision Support System for Congenital Heart Disease Diagnosis based on Signs and Symptoms using Neural Networks", International Journal of Computer Applications vol.19, issue.6, pp.6 – 12, 2011. [

[2]. S.F. Weng, J. Reps, J. Kai, J.M. Garibaldi, N. Qureshi, "Can Machine-Learning Improve Cardiovascular Risk Prediction Using Routine Clinical Data", vol.1, issue.12, pp. e0174944, 2017.

[3]. M. Thiyagaraj, G. Suseendran, "Survey on heart disease prediction system based on data mining techniques", Indian Journal of Innovations and Developments vol.6 issue.1, pp.1-9, 2017.

[4]. C.S. Dangare, S.S. Apte, "Improved Study of Heart Disease Prediction System using Data Mining Classification Techniques", International Journal of Computer Applications vol.47, issue.10, pp. 44-48, 2012.

[5]. S. Palaniappan, R. Awang, "Intelligent heart disease prediction system using data mining techniques", In 2008 IEEE/ACS international conference on computer systems and applications, pp. 108-115, 2008.

[6]. C.B.C. Latha, S.C. Jeeva, "Improving the accuracy of prediction of heart disease risk based on ensemble classification techniques", Informatics in Medicine, Unlocked 16, pp.100203, 2019.

[7] Sneha Borkar, Prof. M. N. Annadate, "Supervised Machine Learning Algorithm for Detection of Cardiac Disorders" Fourth International Conference on Computing Communication Control and Automation (ICCUBEA), IEEE, 2018.

[8] Martin Gjoreski, Monika Simjanoska, Anton Gradisek, Ana Peterlin, Matjaz Gams, Gregor Poglajen, "Chronic Heart Failure Detection from Heart Sounds Using a Stack of Machine Learning Classifiers" International Conference on Intelligent Environments (IE), IEEE, pp: 14-19, 2017.

[9] Raid Lafta, Ji Zhang, Xiaohui Tao, Yan Li, Xiaodong Zhu, Yonglong Luo and Fulong Chen, "Coupling a Fast Fourier Transformation with a Machine Learning Ensemble Model to Support Recommendations for Heart Disease Patients in a Telehealth Environment" IEEE, pp:10674-10685, 2017.

[10] Qingxue Zhang, Dian Zhou, Xuan Zeng, "Hear the Heart: Daily Cardiac Health Monitoring Using Ear-ECG and Machine Learning" 8 th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON), IEEE, pp: 448-451, 2017.

[11] TING-TING ZHAO, YU-BO YUAN, YING JIE WANG, JU GAO, PING HE, "HEART DISEASE CLASSIFICATION BASED ON FEATURE FUSION" International Conference on Machine Learning and Cybernetics (ICMLC), IEEE, vol. 1, pp: 111-117, 2017.

[12] M.Ganesan, Dr.N.Sivakumar, "IoT based heart disease prediction and diagnosis model for healthcare using machine learning models" International Conference on System, Computation, Automation and Networking (ICSCAN), IEEE, pp: 1-5, 2019.

[13] Berina Ali, Lejla Gurbetal, Almir Badnjevi, "Machine Learning Techniques for Classification of Diabetes and Cardiovascular Diseases" 6th Mediterranean Conference on Embedded Computing (MECO), IEEE, pp: 1-4, 2017.