International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 09 Issue: 04 | Apr 2022 www.irjet.net p-ISSN: 2395-0072

Heart Disease Prediction using Machine Learning

Prathamesh Keni [1], Pratik Poshe [2], Kaustubh Latake [3], Prof. Rovina Dbritto [4]

[1] [2] [3] Student, Department of Information Technology, Universal college of Engineering, Vasai
[4] Professor, Department of Information Technology, Universal college of Engineering, Vasai

---

Abstract – *Recently, the cases of heart disease have been increasing rapidly, and it's essential to predict these diseases. The detection of the conditions is difficult to perform, and it should be performed precisely due to its sensitive matter. We have developed an application to predict a person's probability of getting heart disease. We have tested multiple algorithms such as KNN, Support Vector machines, Naive Bayes, Decision tree Classifier, and Random Forest to find the probability. Logistic Regression showed good accuracy in comparison to the other algorithms. The heart disease prediction system enhances the medical care systems and reduces pressure on them. The project is implemented on the android app, and the backend is implemented in python.*

Key Words: **Algorithm, KNN, Logistic Regression, Vector Machines, Naive Bayes, Decision tree Classifier, Random Forest**

1.INTRODUCTION

Machine learning enables software programmes to grow increasingly effective at predicting outcomes without explicitly programming them. Machine learning algorithms estimate new output values by using past data as input [1]. Machine Learning is a diverse field, and its implementation is increasing daily. Machine Learning includes multiple classifiers, categorised as Supervised, Unsupervised and Ensemble Learning which are used to predict and find the precision of a given dataset. We will use the knowledge in our project on Heart Disease Prediction (HDP) as it will benefit many people.

Cardiovascular diseases (CVDs) are the primary cause of mortality worldwide. In 2019, an estimated 17.9 million individuals died from CVDs, accounting for 32% of global fatalities. A heart attack or a stroke caused Eighty-five per cent of these fatalities. [2].

The project's objective is to check whether the patient is likely to have cardiovascular disease or not based on attributes such as age, gender, height, weight, blood pressure, glucose level, cholesterol level, etc. We selected a dataset from Kaggle with the patient's medical history and these attributes. Using this dataset, we can determine whether the patient can have cardiovascular disease or not. We trained the dataset using Logistic Regression, Random Forest and Naive Bayes algorithms. The most efficient of these algorithms was Logistic Regression, with an accuracy of 72.85%. Once the model is trained, the classification of a patient is very cost-efficient.

2. DATA SOURCE

We chose a data source that contains the medical history of 70000 patients. This dataset gave us much-needed information such as age, gender, blood pressure, glucose level, etc. and whether the patient has heart disease or not. This data helps us in creating a machine learning model to classify the patients. The dataset is first split into two parts, Training and Testing. The training part is used to train the model, and the testing part is used to test the trained model. The dataset is split into 80:20, meaning that 80% of data is used for training while 20% is used for testing. All attributes in the dataset are listed below in 'Table 1'.

Table -1: Dataset Attributes

Feature	Variable Type	Variable	Value Type
Age	Objective Feature	age	int(days)
Height	Objective Feature	height	int (cm)
Weight	Objective Feature	weight	float(kg)
Gender	Objective Feature	gender	Categorical Code
Systolic blood pressure	Examination Feature	ap_hi	int
Diastolic blood pressure	Examination Feature	ap_lo	int
Cholesterol	Examination Feature	cholesterol	Categorical Code
Glucose	Examination Feature	gluc	Categorical Code
Smoking	Subjective Feature	smoke	binary
Alcohol intake	Subjective Feature		
Physical	Subjective Feature	active	binary
Presence or absence of cardiovascular disease	Target Variable	cardio	binary

3. PROPOSED SYSTEM

Our system is an app-based machine learning application trained on a dataset from Kaggle. The admin inputs the required attributes to get the prediction for that patient. The model will determine the probability of heart disease. The android application displays the result of the forecast.

We have tested the following three algorithms:

1. Logistic Regression
2. Naïve Bayes
3. Support Vector Machines

The algorithms have been trained and tested using the dataset obtained from Kaggle. 80% of data is used for training, while the rest is used for testing. Additional processing is done on the dataset to make the training process efficient and make the model accurate. The algorithms were tested and judged based on their accuracy, and Logistic Regression was the most accurate. Hence, we selected it for the main application. Flask API is used for connecting the python backend with the android application. We have used SQL to store the patient's data and Admin user accounts.

Input: Data Set, Patient Data

Output: Prediction of Cardiovascular Disease

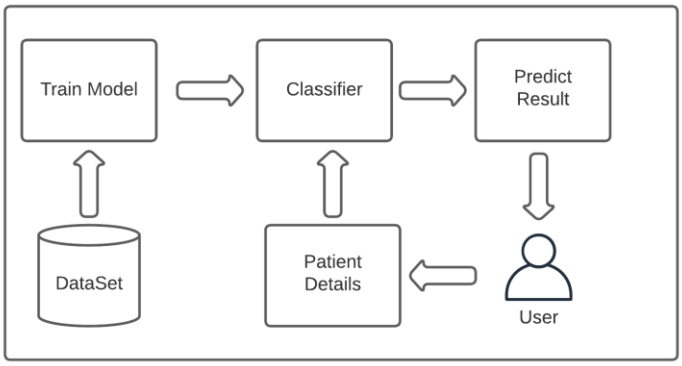


Figure -1: Architecture Diagram

4. PROCESS

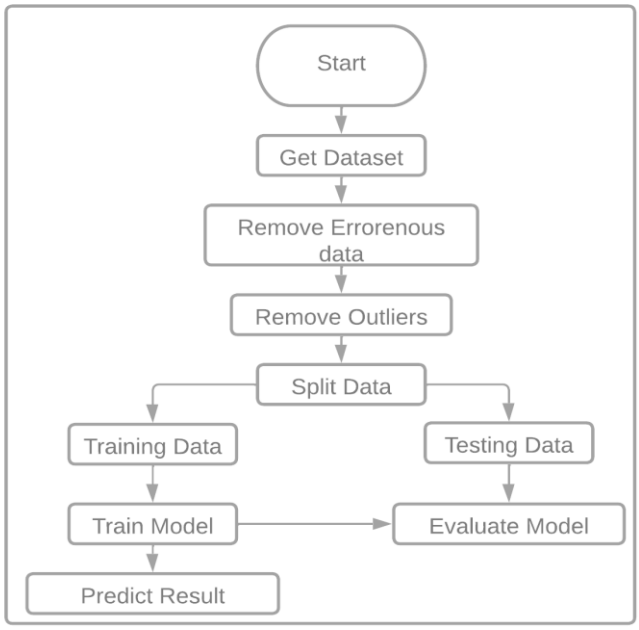


Figure -2: Training Process

4.1 Data Preprocessing

1. Data Cleaning: The dataset obtained from Kaggle is not perfect; first, we check if the dataset has any null values. If there are any empty columns, we will remove the records from the dataset. The dataset contained no null values, so we deleted no record. We also removed the blood pressure values, which were unrealistic from the dataset.

2. Removing Outliers: The next step we took was to detect and remove outliers. Outliers are the values that look different from the other values in the data. Below is a plot highlighting the outliers in 'red' and outliers can be seen in both the extremes of data [3]. We removed height and weight values from the end to normalize the data. The following figure(Figure 2) shows the boxplot for the data used to detect outliers.

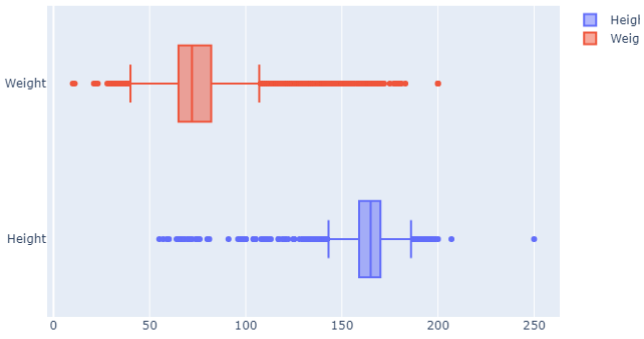


Figure -3: Detecting Outliers

3. Splitting Data: After preprocessing, the remaining data is divided into two parts, Training and Testing. Training data is used for the Training algorithm. Testing data is used for evaluating the algorithm. After testing, we can calculate the accuracy of the algorithm.

4.2 Training Model

We are using a Logistic Regression classifier for training a model for the system. It is a statistical model used for binary classification. We are using Logistic regression because we have to make a binary prediction of the Target Variable, which denotes whether a person is likely to have heart disease or not. Once the model is trained, we move on to evaluate it.

5. IMPLEMENTATION

Once the model is trained, we need a way to give it the data to process and get the predicted result. We have used an API to access the model from outside the system to do this. We have built a REST API using the FLASK framework.

An API, or application programming interface, is a collection of directions that define how programs or devices can link to and communicate with each other. A REST API is an API that fits the design principles of the REST or representational state transfer architectural style. For this reason, REST APIs are sometimes referred to as RESTful APIs [3].

Flask is a micro web framework written in Python. It is categorized as a microframework because it does not need particular tools or libraries [4].

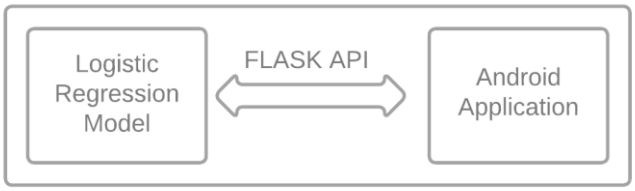


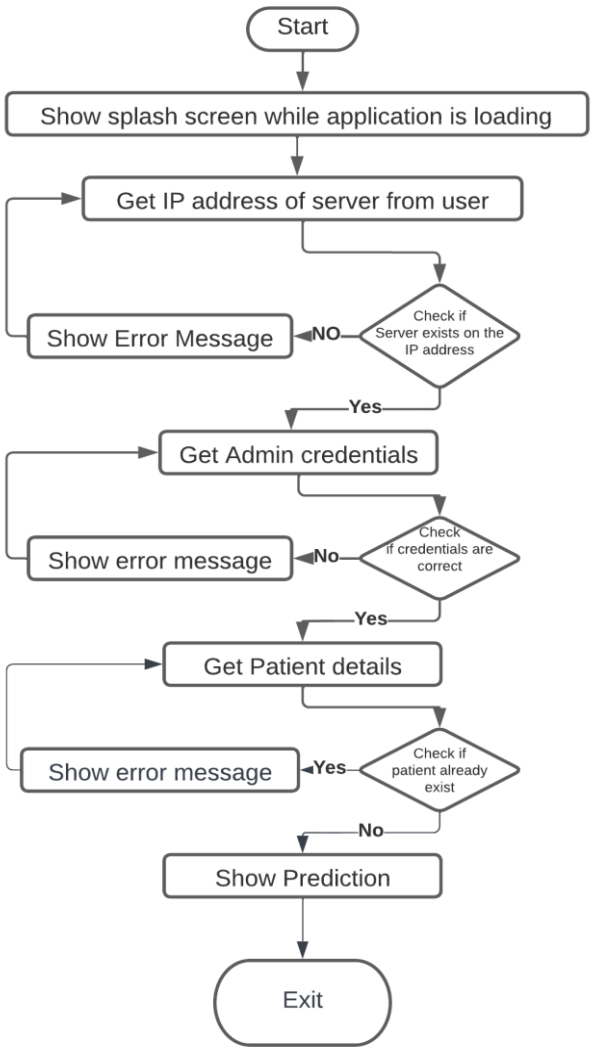
Figure -4: API Connection

By implementing the API, we can access the model and its methods and functions from other systems in the network. As this creates a security concern for the design, we have implemented Authentication for the API using OAuth 2.0.

OAuth 2.0 focuses on client developer transparency while providing detailed authorization flows for web applications, desktop applications, mobile phones, and living room devices. IETF OAuth Working Group does the development of specifications and extensions [5].

We have decided to make an android application using the API.

6. WORKING



6

Figure -5: Implementation Process

6.1. IP Address of the Server

The admin needs to enter the server's IP address, hosting the FLASK RESTful API. This step is required because the server is hosted on the local network whose IP address changes periodically.

6.2 Admin Login

Once the application successfully connects to the API server, the admin must authenticate with the API. When the user enters the valid credentials, the API server sends a Token that is valid for 90 mins and is required for accessing other endpoints of the API.

6.3 Getting Patient data

Once the admin is successfully logged in, we can send the patient's personal and medical details. The data is stored in

the database, and the medical information is then used to indicate whether the patient has CVD or not. The prediction is displayed on the next screen to the user and stored in the database.

7. RESULTS

The heart disease prediction application evaluates the risk of whether a person has heart disease or not successfully. The system has 73% accuracy in diagnosing the heart disease. The following diagram shows the ROC curve of the model.

An ROC curve (receiver operating characteristic curve) is a graph showing the performance of a classification model at all classification thresholds. This curve plots two parameters- True positive rate and False positive rate [6].

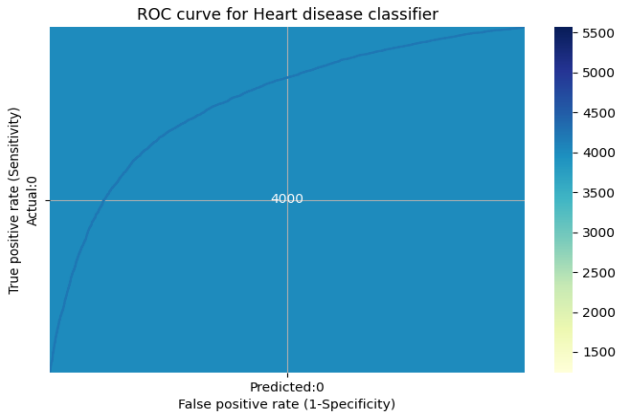


Figure -6: ROC curve

8. CONCLUSION

We tested 3 algorithms and selected Logistic Regression. We chose it because it had the highest accuracy of all three. The False negatives of this model is also low compared to the other models which can be seen in the following figure which is a confusion matrix for the logistic regression model.

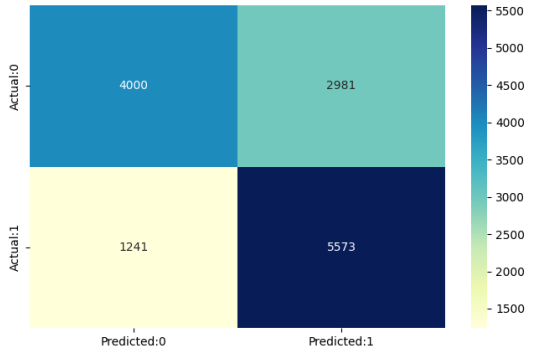


Figure -7: Confusion Matrix

For creating this application, we have used Python for training the model and implementing the Flask API, Java for creating the android application. By using these tools, we have created a user-friendly UI which can be used by novice users very easily and no technical knowledge is needed. This will help users with a preliminary prediction about their heart condition. Since Heart Diseases are increasing rapidly in the world this application can be used for fast diagnosis which can have a profound impact on the world.

ACKNOWLEDGEMENT

We have great satisfaction in presenting the report on "heart disease prediction using Machine Learning". We take this opportunity to express my sincere thanks to my guide, Prof. Rovina Dbritto, for providing the technical guidelines and suggestions regarding the line of this work. We want to convey our gratitude for her constant encouragement, support and guidance throughout the project's development. We are grateful to Dr J. B. Patil (Director, UCOE, Kaman) and Prof. Yogita Mane (HOD, Information Technology, UCOE); our project would not have shaped up without their support. We wish to express my deep gratitude toward all my colleagues at Universal College of Engineering, Kaman, for their encouragement.

REFERENCES

[1] https://www.techtarget.com/searchenterpriseai/definition/machine-learning-ML

[2] https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds)

[3] https://www.ibm.com/in-en/cloud/learn/rest-apis

[4] "Flask Foreword" https://flask.palletsprojects.com/en/2.1.x/foreword/

[5] https://oauth.net/2/

[6] https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc