CARDIOVASCULAR DISEASE DETECTION

A project report submitted to

MALLA REDDY UNIVERSITY

in partial fulfillment of the requirements for the award of degree of

BACHELOR OF TECHNOLGY

in

COMPUTER SCIENCE & ENGINEERING (AI&ML)

Submitted by

P. PRAMOD REDDY : 2011CS020287

Under the Guidance of

Dr. K. LITTLE FLOWER

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING (AI&ML)

MALLA REDDY UNIVERSITY

(Telangana State Private Universities Act No.13 of 2020 and G.O.Ms.No.14, Higher Education (UE) Department)

2023

MALLA REDDY UNIVERSITY
(Telangana State Private Universities Act No.13 of 2020 and G.O.Ms.No.14, Higher Education (UE) Department)

COLLEGE CERTIFICATE

This is to certify that this is the bonafide record of the application development entitled,**"CARDIOVASCULAR DISEASE DETECTION BY MACHINE LEARNING TECHNIQUES"** Submitted by P. PRAMOD REDDY (2011CS020287) B. Tech III year I semester, Department of CSE (AI&ML) during the year 2022-23. The results embodied in the report have not been submitted to any other university or institute for the award of any degree or diploma

INTERNAL GUIDE
Dr. K. LITTLE FLOWER

HEAD OF THE DEPARTMENT
Dr. Thayyaba Khatoon
CSE(AI&ML)

ABSTRACT

Cardiovascular Disease or coronary illness is one of the significant dangerous infections in the entire world. It is also the major cause for significant number of deaths. So, proper and timely diagnosis treatment of such diseases is necessary. It Requires a system that can predict with precise accuracy and reliability. Intensive research is carried out by our team members using diverse machine learning algorithms to forecast the heart disease taking different datasets which consists of different attributes that result in heart attack. We analyzed the dataset collected from Kaggle which consists of attributes related to heart disease such as age, gender, blood pressure, cholesterol and so on.

We employ six classification algorithms to predict cardiovascular disease. Models are trained using a publicly available dataset of cardiovascular disease cases. we proposed an efficient and accurate system to diagnosis heart disease and the system is based on machine learning techniques. The system is developed based on classification algorithms includes Support vector machine, Logistic regression, K-nearest neighbor, Naïve bays, Random Forest, and Decision tree while standard features selection algorithms have been used such as Relief, Minimal redundancy maximal relevance, least absolute shrinkage selection operator and Local learning for removing irrelevant and redundant features.

CONTENTS

INTRODUCTION

1.1 PROJECT DEFINITION

The major challenge in heart disease is its detection. There are instruments available which can predict heart disease but either it is expensive or are not efficient to calculate chance of heart disease in human. Early detection of cardiac diseases can decrease the mortality rate and overall complications. However, it is not possible to monitor patients every day in all cases accurately and consultation of a patient for 24 hours by a doctor is not available since it requires more sapience, time and expertise. Since we have a good amount of data in today's world, we can use various machine learning algorithms to analyze the data for hidden patterns. The hidden patterns can be used for health diagnosis in medical data.

1.2 OBJECTIVE OF PROJECT

The main objective is to develop a system which can detect the type of heart disease that patient is affected from. To detect heart disease with usage of machine learning models. To detect whether Person has a Heart disease or not.

1.3 LIMITATIONS OF PROJECT

- It is not very effective in case of small data.
- Require high knowledge of machine learning development.
- It is difficult to maintain the system.
- It is not widely used at present.

ANALYSIS

2.1 INTRODUCTION

Cardiovascular disease is a category of ailments that are caused by heart problems. Cardiovascular disease may be diagnosed by shortness of breath, physical weakness, swollen feet, and exhaustion. High cholesterol, smoking, sedentary lifestyles, and high blood pressure are a few risk factors for cardiovascular disease. The leading cause of mortality, according to the World Health Organization**.** The most common kind of heart disease is coronary artery disease. As a result, heart disease and stroke are major public health concerns. Prediction aids in the early detection of disease and the most effective treatment. The use of ML in medical diagnostics is proliferating. This work's goal is a classification model that can detect cardiovascular disease. Making predictions based on experience has become a central goal of ML research. The machine learning predictive models need proper data for training and testing. The performance of machine learning model can be increased if balanced dataset is use for training and testing of the model. Furthermore, the model predictive capabilities can be improved by using proper and related features from the data. Therefore, data balancing and feature selection is significantly important for model performance improvement. In literature various diagnosis techniques have been proposed by various researchers, however these techniques are not effectively diagnosis HD. In order to improve the predictive capability of machine learning model data preprocessing is important for data standardization. Various Preprocessing techniques such removal of missing feature value instances from the dataset.

2.1 SOFTWARE REQUIREMENT SPECIFICATION

2.1.1. SOFTWARE REQUIREMENTS

- JUPYTER NOTEBOOK.
- NUMPY
- SCIPY
- SCIKIT-LEARN
- PANDAS

2.2.2 HARDWARE REQUIREMENTS

- RAM: 16 GB
- CPU: 2GHz OR Above
- ARCHITECTURE: 32-bits or 64-bits

2.3 EXISTING SYSTEM

There are many diagnostic centers to diagnose the patients by predicting their heart diseases using blood samples, X- Rays, MRI, CT- images etc. For this, the patient need to visit the hospitals and consult a Cardiologist or Radiologist to know and verify about their health condition. And also there are some applications which only monitor and track their heart rate, pulse, blood pressure, etc.

2.4 PROPOSED SYSTEM

First, we acquired the heart disease dataset from Kaggle. The dataset consists of several heart-related disease records and many target classes. So, the proposed approach begins with the selection of a dataset for the experiment. The Cardiovascular Disease dataset is used for experiments in this research. The preprocessing stage includes many steps that must be completed before model training. The features extraction approach is then utilized to determine the relevance of the features, and finally, several machine learning and deep learning classifiers are employed for experiments. Deep learning methods are also being evaluated in this study to detect cardiovascular disease. To identify the existence of cardiovascular disease.

2.5 MODULES

Collection of dataset:

Initially, we collect a dataset for our heart disease prediction system. After the collection of the dataset, we split the dataset into training data and testing data. The training dataset is used for prediction model learning and testing data is used for evaluating the prediction model. For this project, 70% of training data is used and 30% of data is used for testing. The dataset used for this project is Heart Disease UCI. The dataset consists of 76 attributes; out of which, 14 attributes are used for the system.

Selection of attributes:

Attribute or Feature selection includes the selection of appropriate attributes for the prediction system. This is used to increase the efficiency of the system. Various attributes of the patient like id, age, gender, height, weight, ap_hi, ap_lo, cholesterol, gluc, smoke, alco, active, cardio etc. are selected for the prediction. The Correlation matrix is used for attribute selection for this model.

Pre-processing of Data:

Data pre-processing is an important step for the creation of a machine learning model. Initially, data may not be clean or in the required format for the model which can cause misleading outcomes. In pre-processing of data, we transform data into our required format. It is used to deal with noises, duplicates, and missing values of the dataset. Data pre-processing has the activities like importing datasets, splitting datasets, attribute scaling, etc. Preprocessing of data is required for improving the accuracy of the model.

Balancing of Data:

Imbalanced datasets can be balanced in two ways. They are Under Sampling and Over Sampling

(a) Under Sampling:

In Under Sampling, dataset balance is done by the reduction of the size of the ample class. This process is considered when the amount of data is adequate.

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

Prediction of Disease:

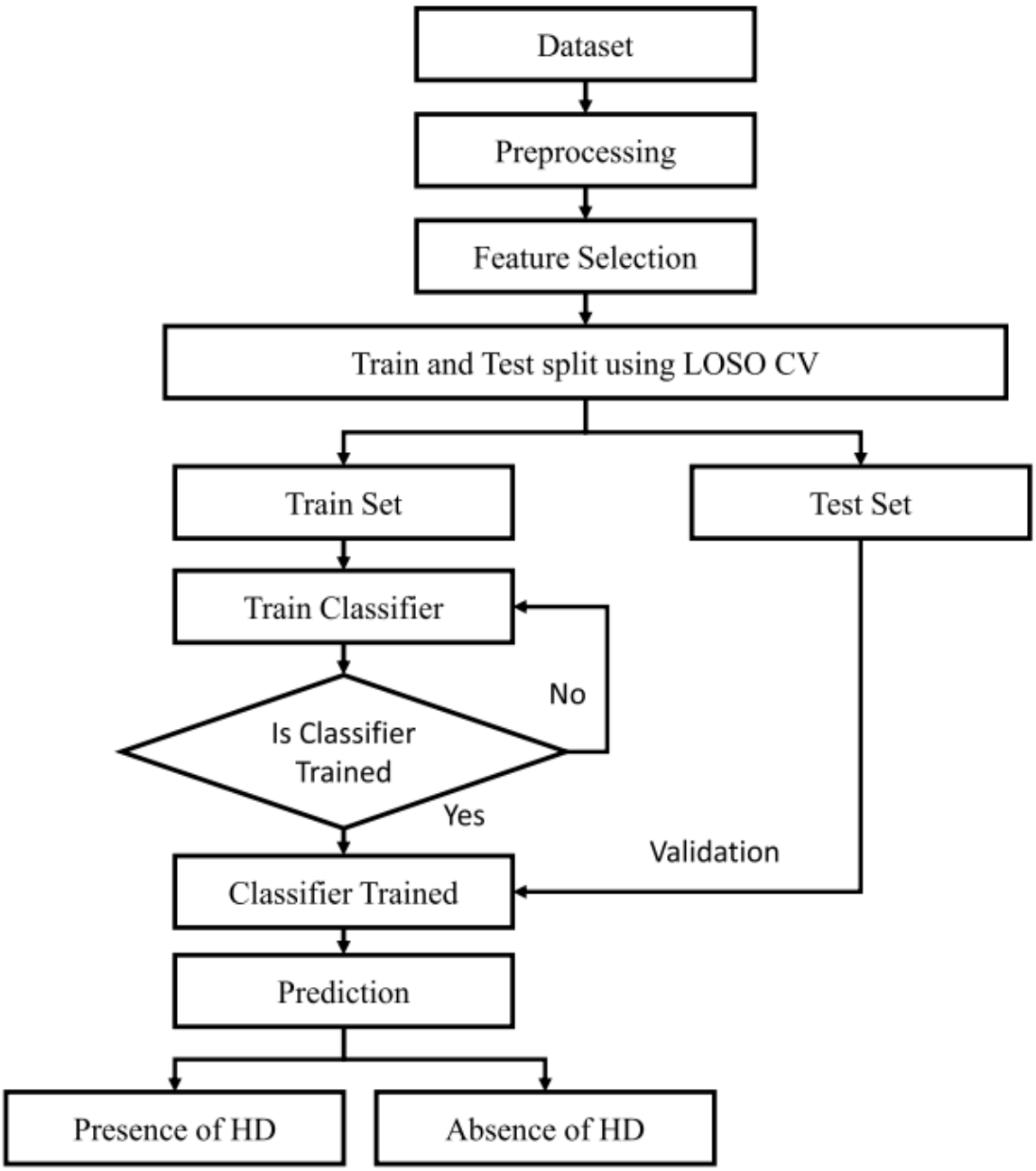
Various machine learning algorithms like SVM, Naive Bayes, Decision Tree, Random Tree, Logistic Regression, Random Forest Classification are used for classification. Comparative analysis is performed among algorithms and the algorithm that gives the highest accuracy is used for heart disease prediction.

Validation:

To define the degree of accuracy or effectiveness of any machine learning model, it is necessary to perform a thorough assessment of the performance obtained when the approach is used to produce forecasts on real data. Therefore, to obtain a more reliable estimate of performance metrics we use cross-validation a statistical method that is particularly suitable for machine learning models assessment.

The metrics we used to validate the model are accuracy, precision, recall, and F-score. In detail, accuracy indicates the accuracy of the model, i.e. the fraction of the test dataset on which the model provides a correct prediction. By defining true positives (TP) and true negatives (TN) as the instances that are correctly classified, and the false positives and false negatives (FN) as the instances that are misclassified.

2.6 ARCHITECTURE



3.DESIGN

3.1 INTRODUCTION

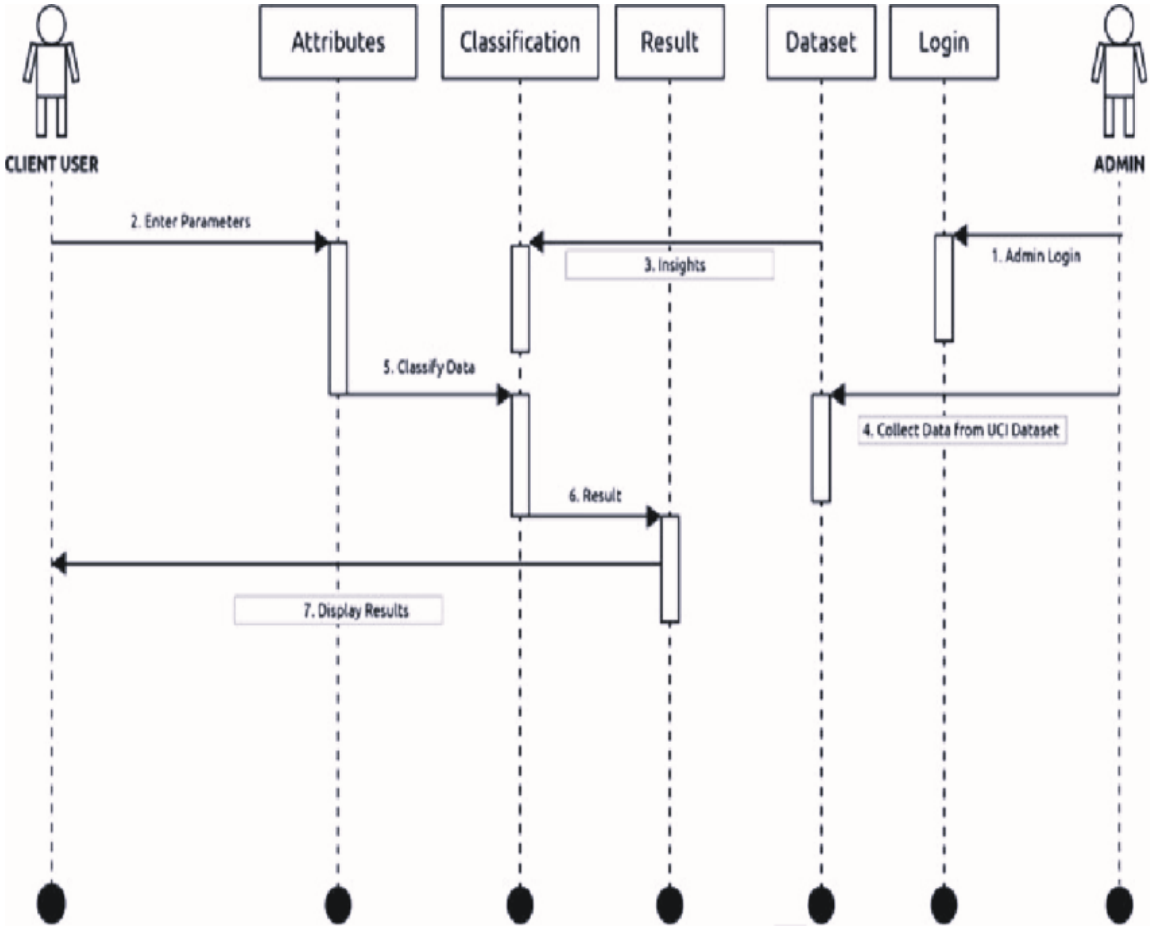
For predicting cardiovascular disease analyzing bp and cholesterol report is required to analyze and predict disease. Cardiovascular data set analysis will be conducted using various supervised machine learning classifier techniques. Based on the accuracy of different algorithm, best accuracy algorithm will be chosen to fetch the result.

Cleaned data is used as training data and test data, which is fed as input to the algorithm.
The algorithm extracts the features from different dataset to classify the data according to the labels.
To check the accuracy of the prediction, test data is fed to the algorithm.

Based on the feature extracted, probability will be generated for test data by comparing the features of both. Highest probability value will be classified to that particular label whether it is heart disease or not

Web app is developed using Django (back end) and web application is designed using HTML5 (front end) and CSS, where the chosen ML model will be linked with web app and HTML. The users features data will be entered in web app front end and the back end will process the data using model and result will be designed on screen

3.2 DFD/ER/UML DIAGRAM



3.3 DATA SET DESCRIPTIONS:

The dataset consists of 70 000 records of patient data, 11 features + target.

There are 3 types of input features:

- Objective: factual information;
- Examination: results of medical examination;
- Subjective: information given by the patient.

SL. No.	Attribute	Value type
01	Age	Numerical
02	Height	Numerical
03	Weight	Numerical
04	Gender	Nominal
05	Systolic blood pressure	Numerical
06	Diastolic blood pressure	Numerical
07	Cholesterol	Nominal
08	Glucose	Nominal
09	Smoking	Nominal
10	Alcohol intake	Nominal
11	Physical activity	Nominal
12	Presence or absence of cardiovascular disease	Nominal

3.4 DATA PREPROCESSING TECHNIQUES

Raw data from the real world is frequently incomplete, unreliable, and devoid of specific behaviors or trends. They're also likely to have a lot of mistakes in them. As a result, they are pre-processed into a format that the machine learning algorithm can use for the model once they have been collected. The data pre-processing phase should be given a lot of attention in order to get the best model quality. It includes several tasks employed in the process to make the data more relevant. In this study, we followed the following steps in order to preprocess the data. At first, there were many unclear values which actually didn't had any significant meaning. So, we removed those unclear values to get better results from this process by reducing the datasets attributes. Following that, replaced missing value because there were so many missing values in our dataset.

We took different steps to handle missing values for example, filling missing values with median, mode. In addition, the categorical data was encoded into integer format so that data with transformed category values may be fed into models to improve prediction accuracy. Furthermore, we handled the

imbalance data from our datasets in which the target class had an unequal distribution of observations. For balancingour dataset, we employed the resampling technique. Finally, we spilled the datasets into training and test sets. The training dataset was utilized to fit the model and test sets were used to make predictions and compare them to the predicted values.

3.5 METHODS & ALGORITHMS

SUPPORT VECTOR MACHINE (SVM):

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 support vectors, and hence the algorithm is termed as Support Vector Machine.

Support vector machines (SVMs) are powerful yet flexible supervised machine learning algorithms which are used both for classification and regression. But generally, they are used in classification problems. In the 1960s, SVMs were first introduced but later they got refined in 1990. SVMs have their unique way of implementation as compared to other machine learning algorithms. Lately, they are extremely popular because of their ability to handle multiple continuous and categorical variables.

NAIVE BAYES ALGORITHM:

Naive Bayes algorithm is a supervised learning algorithm, which is based on Bayes theorem and used for solving classification problems. It is mainly used in text classification that includes a high-dimensional training dataset.

Naive Bayes Classifier is one of the simple and most effective Classification algorithms which helps in building the fast machine learning models that can make quick predictions.

It is a probabilistic classifier, which means it predicts on the basis of the probability of an object. Some popular examples of Naïve Bayes Algorithm are spam filtration, Sentimental analysis, and classifying articles.

It is a classification technique based on Bayes' Theorem with an assumption of independence among predictors. In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature.

The Naive Bayes model is easy to build and particularly useful for very large data sets. Along with simplicity, Naive Bayes is known to outperform even highly sophisticated classification method.

DECISION TREE 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.

The Decision Tree Algorithm belongs to the family of supervised machine learning algorithms. It can be used for both a classification problem as well as for a regression problem.

The goal of this algorithm is to create a model that predicts the value of a target variable, for which the decision tree uses the tree representation to solve the problem in which the leaf node corresponds to a class label and attributes are represented on the internal node of the tree.

RANDOM FOREST ALGORITHM:

Random Forest is a supervised learning algorithm. It is an extension of machine learning classifiers which include the bagging to improve the performance of Decision Tree. It combines tree predictors, and trees are dependent on a random vector which is independently sampled. The distribution of all trees are the same. Random Forests splits nodes using the best among of a predictor subset that are randomly chosen from the node itself, instead of splitting nodes based on the variables. The time complexity of the worst case of learning with Random Forests is O(M(dnlogn)), where M is the number of growing trees, n is the number of instances, and d is the data dimension.

It can be used both for classification and regression. It is also the most flexible and easy to use algorithm. A forest consists of trees. It is said that the more trees it has, the more robust a forest is.

Random Forests create Decision Trees on randomly selected data samples, get predictions from each tree

and select the best solution by means of voting. It also provides a pretty good indicator of the feature importance.

Random Forests have a variety of applications, such as recommendation engines, image classification and feature selection. It can be used to classify loyal loan applicants, identify fraudulent activity and predict diseases. It lies at the base of the Boruta algorithm, which selects important features in a dataset. 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.

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.

In Logistic regression, instead of fitting a regression line, we fit an "S" shaped logistic function, which predicts two maximum values (0 or 1). The curve from the logistic function indicates the likelihood of something such as whether the cells are cancerous or not, a mouse is obese or not based on its weight, etc. Logistic Regression is a significant machine learning algorithm because it has the ability to provide probabilities and classify new data using continuous and discrete datasets.

K – NEAREST NEIGHBOURS (KNN) :

Fix and Hodge proposed the KNN model which is one of the simple and most widely used supervised classification algorithm. It works by calculating distance between the test and training datapoints inorder to identify its nearest neighbours. The new sample is then assigned to the class of its closet neighbor. In KNN, K represents the number of closest neighbors. The classification is greatly influenced by this value of ‘K’KNN works with numeric data which uses various measures such as Euclidean, Manhattan.

3.6 BUILDING A MODEL

- Collect and annotate a dataset which consist records of Cardio disease. The dataset should be large enough to provide the model with sufficient examples to learn from, It is so large that it takes a long time to train the model
- Preprocess the data so that the raw data which we collected from repository is converted in to a clean data set.
- Choose the model, here we used classification model support vector machine, decision tree, random forest, k-nearest neighbours so we can predict the output in the form of categories like whether there is disease or not.
- Train the model on the preprocessed data on chosen machine learning model. You may need to experiment with different values for the hyper parameters, such as the learning rate and batch size, to find the best settings for your dataset.
- Test the trained model on a separate dataset to evaluate its performance. You can use metrics such as accuracy, precision, recall, and F1 score to assess the model's performance.
- If the model's performance is not satisfactory, you may need to fine-tune the model by adjusting the model architecture, hyper parameters, or training data.
- Once the model is performing well, you can now test the model by providing required parameters as input to the model so the predicted output is shown.

3.7 EVALUATION

In the context of machine learning, performance metrics refer to how well an algorithm performs depending on various criteria such as precision, accuracy, recall, F1 score and so on. The next sections go through several performance metrics.

ACCURACY

The percentage of correct test data predictions referred to as accuracy. It is easy to calculate by dividing the total number of forecasts by the number of correct guesses.

PRECISION

The precision score used to assess the model's accuracy in counting genuine positives correctly among all positive predictions.

RECALL

The recall score used to assess the model's performance in terms of accurately counting true positives among all actual positive values.

F1 SCORE

F1-score is the harmonic mean of precision and recall score, and utilized as a metric in situations when choosing either precision or recall score can result in a model with excessive false positives or false negatives.

	precision	recall	f1-score	support
0	0.70	0.74	0.72	10352
1	0.73	0.69	0.71	10648
accuracy			0.72	21000
macro avg	0.72	0.72	0.72	21000
weighted avg	0.72	0.72	0.72	21000

4 DEPLOYMENT & RESULTS

4.1 INTRODUCTION

For deploying CARDIOVASCULAR DISEASE DETECTION, we have taken dataset of cardiovascular, so we can trainour model to efficiently predict the disease to do that we have trained our model with different ML algorithms. After successfully completing execution of model there will a pickle or file generated which are basically binary file this file is used to deploy our model into a website by using python Django (back-end) and web application is designed using HTML5 (front end) and CSS, where the chosen ML model will be linked with web app and HTML. The users features will be entered in web app front end and the back end will process the data using model and result will be designed on screen.

4.2 SOURCE CODE:

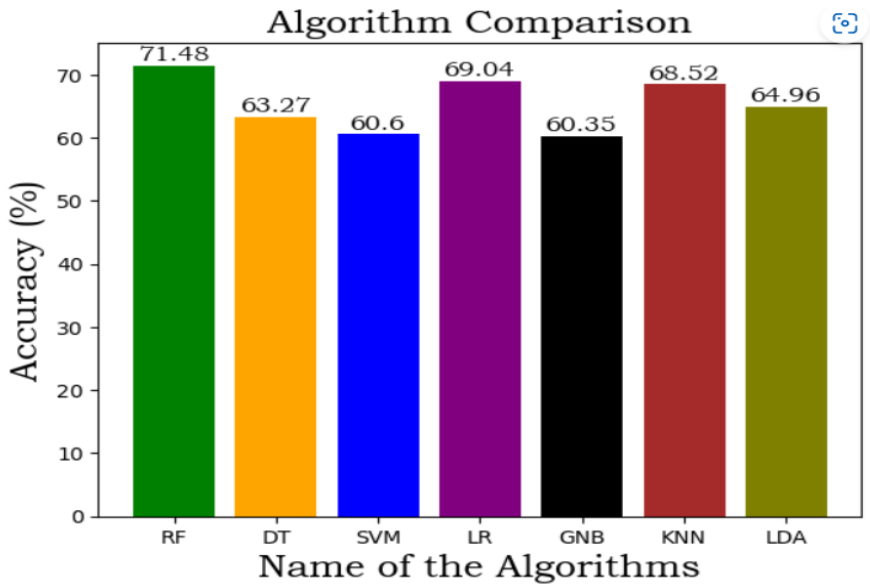
```
#Importing Libraries
import pandas as pd
from pandas.plotting import scatter_matrix
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns from sklearn.metrics
import accuracy_score from sklearn.metrics
import confusion_matrix from sklearn.metrics
import classification_report
 #Importing Dataset
df = pd.read_csv('cardio_train.csv')
# dividing features and columns
x = df.drop(['cardio', 'id'], axis = 'columns')
y = df['cardio']
# Dividing into Training and Testing Data
from sklearn.model_selection import train_test_split
xtrain, xtest, ytrain, ytest = train_test_split(x, y, test_size = .30, random_state = 1)
#Showing xtrain
xtrain
#Showing ytrain
 ytrain
```

```
#Model Developing using Random Forest
from sklearn.ensemble import RandomForestClassifier
rfc = RandomForestClassifier(n_estimators = 100)
rfc.fit(xtrain, ytrain)
pred = rfc.predict(xtest)
rfc.score(xtest, ytest)
cr = classification_report(ytest, pred)
print (cr)
cm = confusion_matrix(ytest, pred)
print (cm)

#Model Developing using Decision Tree
from sklearn.tree import DecisionTreeClassifier
 dtc = DecisionTreeClassifier()
 dtc.fit(xtrain, ytrain)
pred = dtc.predict(xtest)
dtc.score(xtest, ytest)
cr = classification_report(ytest, pred)
print (cr)
cm = confusion_matrix(ytest, pred)
 print (cm)
#Model Developing using Support Vector Machine
from sklearn.svm import SVC
svm = SVC()
svm.fit(xtrain, ytrain)
svm.score(xtest, ytest)
pred = svm.predict(xtest)
 cr = classification_report(ytest, pred)
 print (cr)
cm = confusion_matrix(ytest, pred)
 print (cm)

#Model Developing using Logistic Regression
```

```
from sklearn.linear_model
import LogisticRegression
lr = LogisticRegression()
lr.fit(xtrain, ytrain)
lr.score(xtest, ytest)
pred = lr.predict(xtest)
cr = classification_report(ytest, pred)
print (cr)
cm = confusion_matrix(ytest, pred)
print (cm)

#Model Developing using Gaussian Naive Bayes
from sklearn.naive_bayes
import GaussianNB gnb = GaussianNB()
gnb.fit(xtrain, ytrain)
gnb.score(xtest, ytest)
pred = gnb.predict(xtest)
cr = classification_report(ytest, pred)
print (cr)
cm = confusion_matrix(ytest, pred)
print (cm)

#Model Developing using K-Nearest Neighbors
from sklearn.neighbors import KNeighborsClassifier
knn = KNeighborsClassifier()
 knn.fit(xtrain, ytrain)
knn.score(xtest, ytest)
pred = knn.predict(xtest)
cr = classification_report(ytest, pred)
print(cr)
cm = confusion_matrix(ytest, pred)
print (cm)

#Model Developing using Linear Discriminant Analysis
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
lda = LinearDiscriminantAnalysis()
lda.fit(xtrain, ytrain)
```

```
lda.score(xtest, ytest)
pred = lda.predict(xtest)
cr = classification_report(ytest, pred)
print (cr)
cm = confusion_matrix(ytest, pred)
print (cm)

#Visualization of the Confusion Matrix
p = sns.heatmap(pd.DataFrame(cm), annot = True, cmap = 'YlGn', fmt = 'g')
plt.title('Confusion Matrix for Linear Discriminant Analysis')
plt.xlabel('Predicted Label')
plt.ylabel('Actual Label')
In  x = np.array(["RF", "DT", "SVM", "LR", "GNB", "KNN", "LDA"])
 y = np.array([72.04, 63.3, 60.1, 70.4, 59.1, 68.2, 64.6])
colors_list = ['Green','Orange', 'Blue', 'Purple', 'Black', 'Brown', 'Olive']
plt.xlabel('Name of the Algorithms', fontname="Bookman Old Style", fontsize=18)
plt.title('Algorithm Comparison', fontname="Bookman Old Style", fontsize=18)
plt.ylabel('Accuracy (%)', fontname="Bookman Old Style", fontsize=18)
 pb = plt.bar(x, y, color = colors_list)
for i in range(len(x)):
plt.text(i, y[i], y[i] , ha = "center", va = "bottom", fontname="Bookman Old Style", fontsize = 12)
plt.show()
```

4.3 FINAL RESULT:



5.CONCLUSION

5.1 PROJECT CONCLUSION

This study proposed classifier-based machine and deep learning approaches to predict cardiovascular disease. For this study, we used 70000 patients' data with various forms of cardiovascular disease. The accuracy of the models was used as a measure of their performance. In addition, we selected additional informative characteristics that influence the models' performance. According to the results, the ML Ensemble model was the most accurate in predicting cardiovascular disease. We have added a few procedures to get the dataset ready for analysis.

5.2 FUTURE SCOPE

We may employ various strategies to find the best features for our future work. For a better and more accurate assessment, additional datasets may be employed. Finally, deep learning and reinforcement learning approaches may tackle the prediction issue to detect cardiovascular disease more efficiently.