

HEART DISEASE PREDICTION AND RECOMMENDATION SYSTEM USING MACHINE LEARNING

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

PRIYANSHU

(reg no. 38110430)



**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
JEPPIAAR NAGAR, RAJIV GANDHI
SALAI, CHENNAI – 600 119**

APRIL - 2022



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 bonafidework of **Priyanshu** (Reg. No. 38110430) who carried out the project entitled "**Heart Disease Prediction and Recommendation System using Machine Learning**" under our supervision from November, 2021 to April 2022.

Internal Guide

Dr. A. Mary Posonia, M.E., Ph.D.,

Head of the Department

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

Dr. S. VIGNESHWARI, M.E., Ph.D.,

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I, **Priyanshu** hereby declare that the Project Report entitled “**Heart Disease Prediction and Recommendation System using Machine Learning**” is done by me under the guidance of Associate Professor **Dr. A. Mary Posonia, M.E., Ph.D.**, of the Department of CSE at Sathyabama Institute of Science and Technology and is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering.

DATE:

PLACE:CHENNAI

SIGNATURE OF THE CANDIDATES

1.

2.

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to 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.**, and **Dr. S. Vigneshwari, M.E., Ph.D.**, Heads of the Department, Dept. 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. Mary Posonia, M.E., Ph.D.**, for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

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

ABSTRACT

There are millions of people right now experiencing suffering through heart disease. Prediction of these disease is very necessary for the survival of any individuals. In this paper the disease is accurately predicted by looking at the symptoms of a patient. Algorithmic models such as naïve bayes, random forest, logistic regression and KNN have been used to predict the disease. Once the disease is predicted by the system, it then recommend which type of doctor to consult. Over this an interactive interface is built for recommendation.

TABLE OF CONTENTS

CHAPTER No.	TITLE	PAGE No.
	ABSTRACT	V
	LIST OF FIGURES	Viii
	LIST OF TABLES	lx
1.	INTRODUCTION	12
	1.1 RELATED WORK	12
	1.2 PROPOSED SYSTEM	13
	1.3 ADVANTAGES OF PROPOSED SYSTEM	14
2.	MACHINE LEARNING MODELS	15
	2.1 K-NEAREST NEIGHBOR	15
	2.2 RANDOM FOREST CLASSIFICATION	15
	2.3 DECISION TREE CLASSIFICATION	16
	2.4 RECOMMENDATION SYSTEM	16
3.	RESULTS AND DISCUSSIONS	17
4.	CONCLUSION	22
	REFERENCES	23
	APENDIX	26

LIST OF FIGURES

FIGURE No.	FIGURE NAME	PAGE No.
1.1	PROPOSED SYSTEM ARCHITECTURE	13
3.1	HISTOGRAM DEPICTION OF ATTRIBUTES OF DATASET	19
3.2	NUMBER OF PATIENT WHICH DON'T HAVE HEART DISEASE	20
3.3	GRAPHICAL REPRESENTATION OF KN	20
3.4	WELCOME PAGE FOR USERS	21
3.5	MEDICAL REPORT FOR VERIFICATION	21
3.6	ANALYSIS OF MULTIPLE PATIENTS TO AVOID LATENCY	22
3.7	DOCTOR APPOINTMENT PORTAL	22

LIST OF TABLES

TABLE No.	TABLE NAME	PAGE No.
1	MODEL NAME AND THEIR MEAN SCORE	21

CHAPTER 1

INTRODUCTION

Machine learning is a modern approach that can help in determination of disease using model information or sample/past data. It has two dataset: training and test set. The main objective of this project is to provide a user friendly platform as a front end and combine with ML models using python. It is a urgent need for study and develop a framework which will beneficiary for end user to anticipate the diseases and predict the type of doctor they should visit. Machine learning framework allows the prediction to be more accurate.

The main limitation of the existing system is that we have prediction framework only for specific diseases based on their medical parameters not on the their symptoms (e.g. chest pain, heavy breathing).

These models only apply in the areas where traditional healthcare system already exist.

Also only limited number of users can access the doctor recommendation system. Meanwhile the user interface is also not effective.

There are also noted difficulties like having trouble in finding best specialist for a specific disease.

1.1 RELATED WORK

I have reviewed five papers from different other sources to make this project. I studied the relationship between different algorithms and their performance in different scenarios for the prediction of disease.

[1] Aim of this paper is to improve the slow and meticulous treatment method by introducing the ML technology. It is a comprehensive paper on the prediction of heart diseases. Prediction of the disease with the help of patient's symptoms is done. A complete hospital data is also maintained. Author Mrs. Shital P and others have performed with experimentation in different factors and obtained 88.7% accuracy.

[2] In this paper then are given attributes in the dataset and it is studied on

supervised binary classification. Datasets included in this paper are plasma glucose concentration blood pressure (mm Hg), bmi, age, etc. The improvement of software and site is in detailed manner.

[3] By this paper, it was concluded that the framework should be efficiently optimized in order to anticipate the disease, before visiting the doctor or performing a case of special diagnosis. This paper noted the accuracy being 95.28%, for random forest and 88% for naïve bayes.

[4] This paper studies that the innovation has being scaled in healthcare industries and the ways to improve it further. The medical care industry gathers patient's symptoms and history by applying data mining and the ML techniques are applied to predict the disease.

[5] It provided a heart disease related supervised learning algorithm. The algorithm included here are SVM, KNN, and Naïve bayes. The dataset contains 3000 objects and 14 features. Naïve bayes reported 86.6%, decision tree 78.69% and KNN reported 77.85% accuracy.

1.2 PROPOSED SYSTEM

In this research I have discovered the answer for the issues looking in existing framework. In this system I have used three ML models namely KNN, Random Forest Classifier and Decision Tree Classifier to study the accuracy in predicting the disease using various symptoms. On top of that an User friendly interface is also built for patients to freely interact.

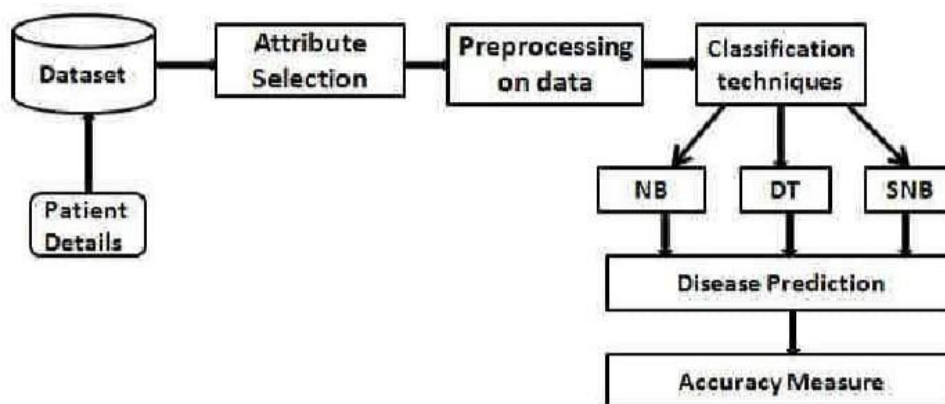


Fig 1.1: - Proposed System Architecture

1.3 ADVANTAGES OF PROPOSED SYSTEM

- 1) The system could reduce medical errors and improve patient results.
- 2) Simple method of getting to the application for Disease Prediction and Doctor Recommendation System
- 3) General Disease Prediction based on symptom and specific Disease Prediction based on Medical parameters gives the output on the same interface.

CHAPTER 2

MACHINE LEARNING MODELS

2.1 K-NEAREST NEIGHBOR

A. KNN is used as it is known to provide high accuracy on ml models. The quality of measure is dependent on the distance between k values.

- 1) Value of k is taken from range 1 to 21
 - 2) For each value of k, score is calculated
 - 3) The average of the scores is appended
 - 4) After applying cross validation a graph is plotted for score vs k value
- value of k is generated using

$$k = \text{sqrt}(N)/2$$

where N is total number of samples.

2.2 RANDOM FOREST CLASSIFICATION

Random Forest is used here as a classifier. It creates a random group of decision trees and merges them together. The more is the number of trees, the more will be accuracy.

- 1) Random Forest Classifier module is imported from scikitlearn.
- 2) Value of the decision tree is arbitrarily chosen as 10.
- 3) Cross validation score is applied
- 4) The mean of all the scores generated is calculated which gives us the accuracy of the model.

2.3 DECISION TREE CLASSIFICATION

Decision tree is used here to create training model. To select the best attribute for the root node Gini index is used. It is a measure of purity or impurity while creating the decision tree.

- 1) By applying the decision tree classifier through scikitlearn a decision tree is generated with the best attributes.

2) Data is splitted using entropy on an attribute

$$\text{Gain}(T, X) = \text{Entropy}(T) - \text{Entropy}(T, X)$$

T – Target Variable

X- Feature to be split on

3) The decision tree algorithm is fitted into the training set

4) The test set results are predicted and its accuracy is generated.

5) Finally we visualize the test set

6) The results are calculated using Gini index and it is in the form of confusion matrix.

formula for Gini index is

$$\text{Gini} = 1 - \sum_{i=1}^n (p_i)^2$$

where pi is the probability of an object being classified to a class.

2.4 RECOMMENDATION SYSTEM

Recommendation system is built using gradio module which allows to build ML user interfaces conveniently.

- a. Welcome page is built by defining a function “greet” and assigning the user interface value
- b. iface.launch() is used to open the interface every time code is run.
- c. A system is developed to predict as well as assign doctor.

CHAPTER 3

RESULTS AND DISCUSSIONS

The dataset was used from the source <https://www.kaggle.com/ronitf/heart-disease-uci>

By reading the dataset and before the processing phase, histogram representation of the different attributes of the dataset has been derived. In the figure one can see the different correlations of the attributes with their measurements.

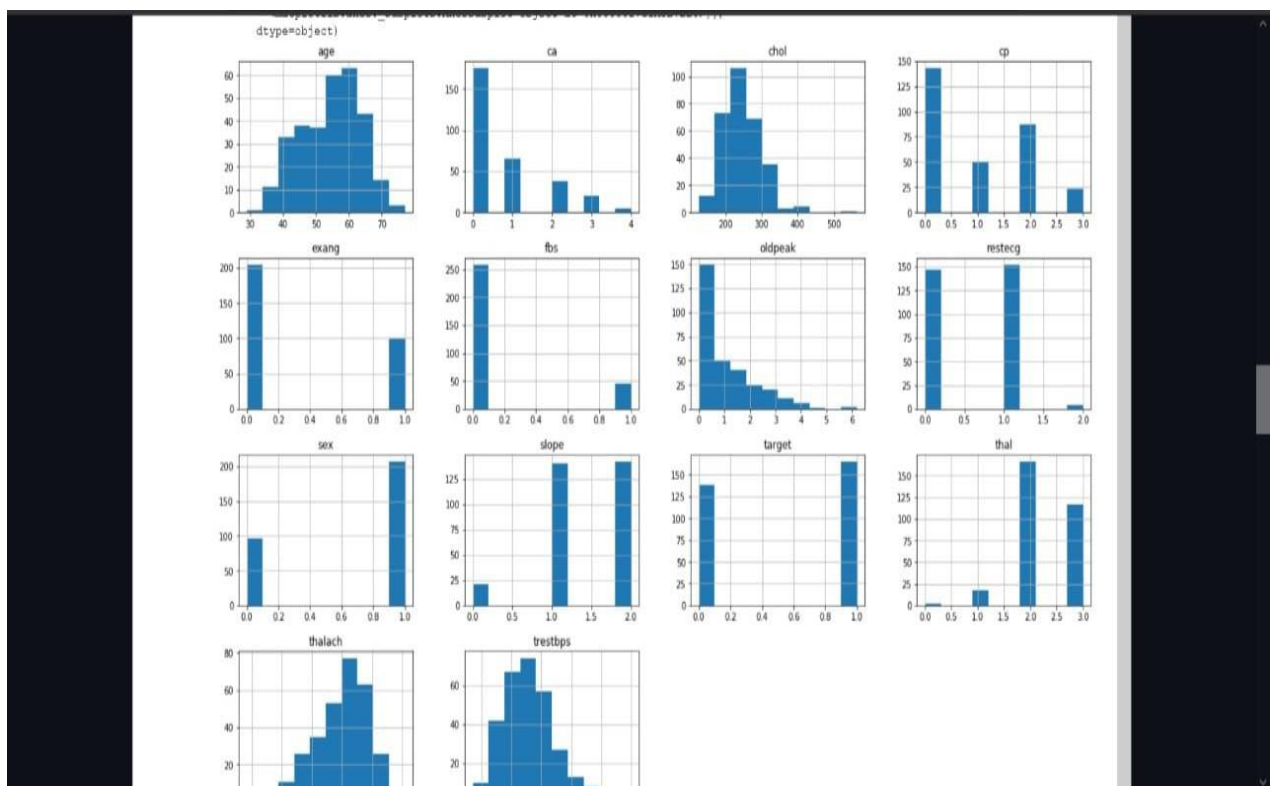


Fig 3.1 : histogram depiction of the attributes of dataset.

The different attributes of the datasets are listed below with their data types. Some of them are categorical data e.g. sex, cp, fbs, slope.

In order to measure the accuracy we already have the target result in our dataset (the target set should be of relatively equal size).

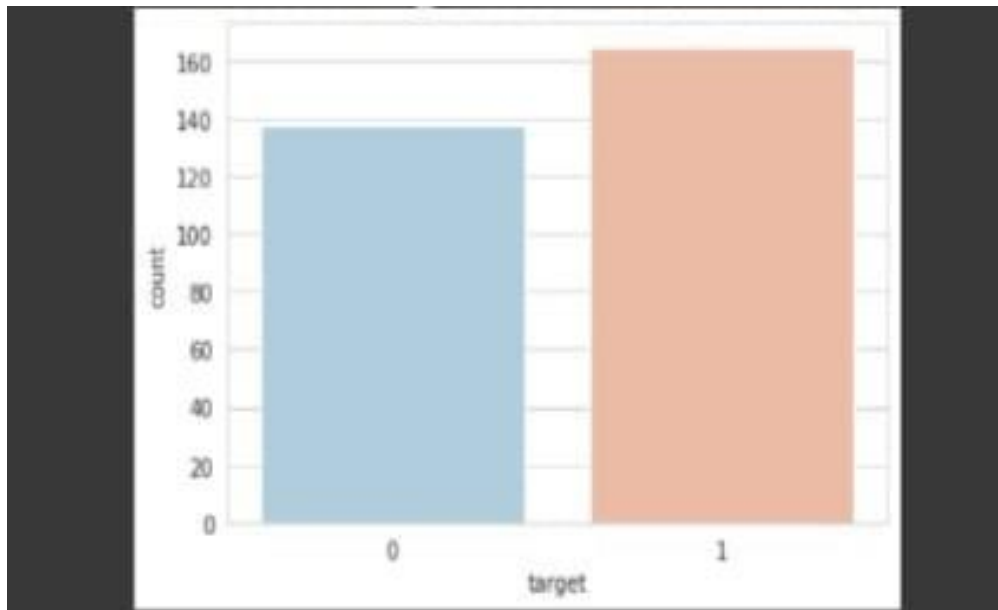


Fig 3.2: 0 represent the number of patient which don't have any heart disease and vice versa

And after the phase of data preprocessing the graph for KNN has been plotted for k in range 1 to 21 I.e. for 20 k values.

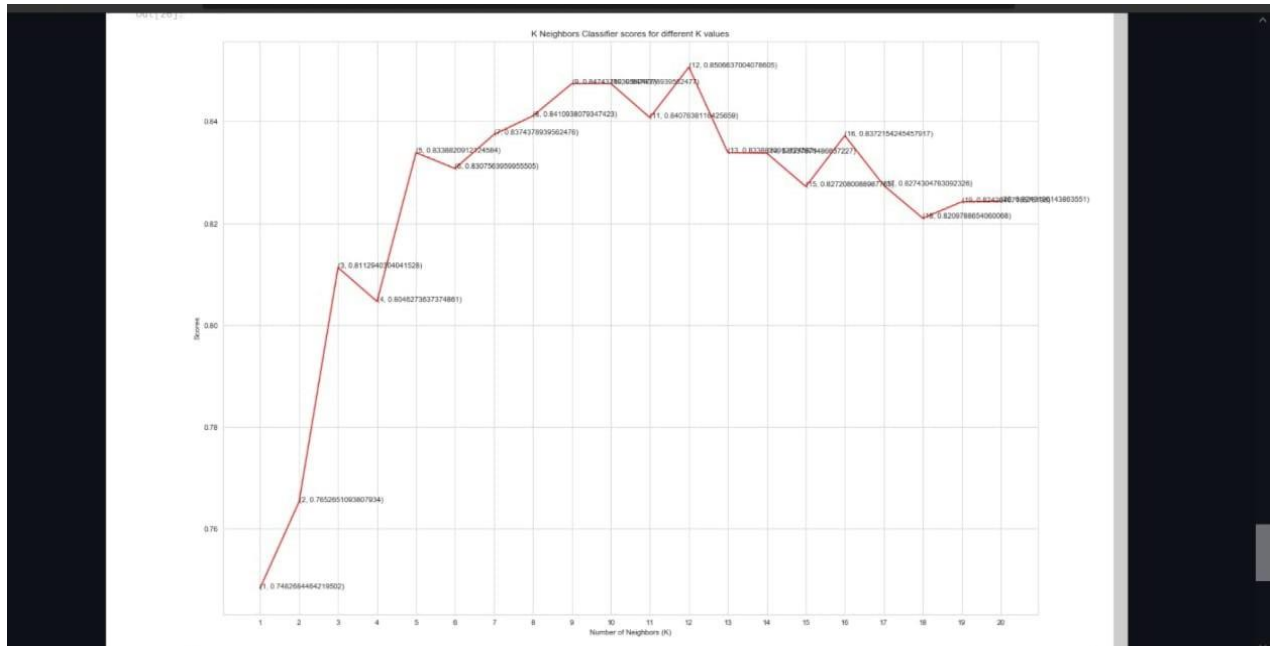


Fig 3.3: Graphical representation of KNN score for each value of k

The above graph is plotted for Neighbors Classifier scores for different K values.

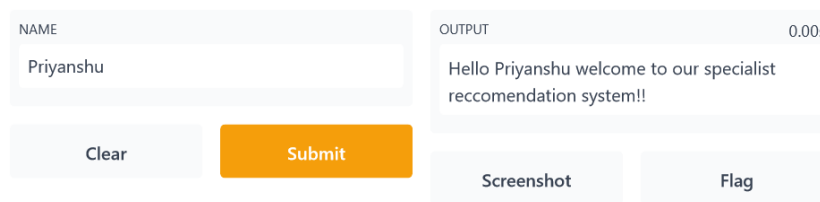
Similarly for other models as well accuracy is calculated and is noted down

in a tabular format given below.

Model Name	Accuracy Calculated (mean score)
K Nearest Neighbor	0.8448387096774195
Random Forest Classifier	0.7981720430107526
Decision Tree	0.815465853983447

Table 1: Model name and their mean score

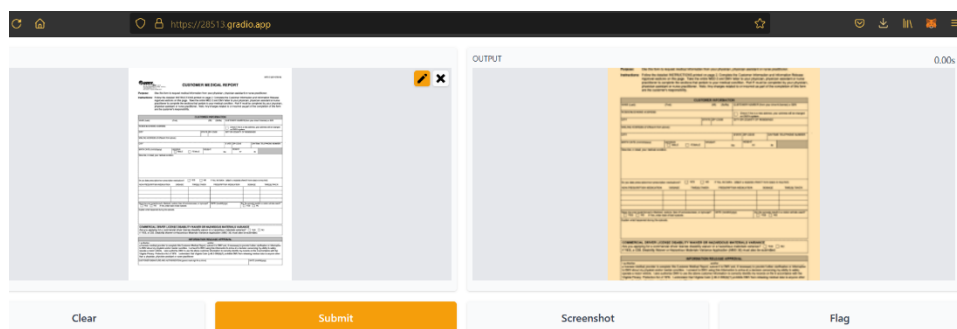
After the accuracy for different models were calculated. Recommendation system was built using gradio module.



The image shows a Gradio web interface. On the left, there is a text input field labeled 'NAME' containing the text 'Priyanshu'. Below the input field are two buttons: 'Clear' and 'Submit'. On the right, there is an 'OUTPUT' section with a timer showing '0.00s'. The output text reads: 'Hello Priyanshu welcome to our specialist recommendation system!!'. Below the output section are two buttons: 'Screenshot' and 'Flag'.

Fig 3.4: Welcome page for users

After that the user has to provide his/her medical report in order to verify that the data has been gathered from an actual Health Center. This phase is mainly for verification purpose. (Note: The paper is blurred on purpose as only verification is needed).



The image shows a Gradio web interface for medical report submission. The browser address bar shows 'https://20511.gradio.app'. The interface has two main columns. The left column contains a blurred image of a medical report form, a 'Clear' button, and a 'Submit' button. The right column contains an 'OUTPUT' section with a timer showing '0.00s', a blurred image of the same medical report form, a 'Screenshot' button, and a 'Flag' button. At the bottom of the interface, there is a footer that says 'view the api • built with' followed by the Gradio logo.

Fig 3.5: Medical Report Submission for patient verification

To avoid latency multiple patient's report from the same hospital can be compared and plotted on a graph to find out easily who has more chances of having any kind

of heart disease.

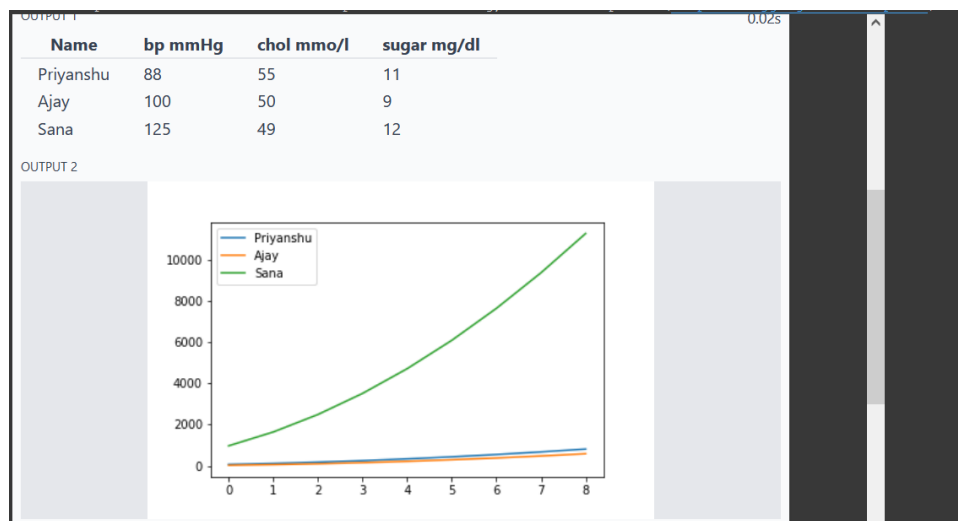


Fig 6: Analysis of multiple patient to avoid latency.

After the system gives output whether the patient has heart disease or not, he/she can schedule an appointment for the preferred time slot.

Figure 7 displays the Doctor Appointment Portal. It includes a form for selecting a time slot and a list of available doctors.

TIMESLOT

10:00 AM to 2:00 PM

OUTPUT 0.00s

Doctors available during 10:00 AM to 2:00 PM are

- John Smith MBBS
- Ravi Shankar MD
- R N Verma DM

Buttons: Clear, Submit, Screenshot, Flag

Fig7: Doctor Appointment Portal

CHAPTER 4

CONCLUSION

From the observations made it was found that on the given dataset K Nearest Neighbor recorded the highest accuracy of 85%, while the other two models failed to record such higher accuracy while limiting upto 81% and 78 % for Random Forest and Decision Tree Classifier respectively. Finally it can be said that KNN provides more accurate results every time we apply it as compared to other ML models. Recommendation system which analyses different patient data can also avoid latency and also provide faster result for the user.

REFERENCES

- [1] Mrs. Shital. P. Chattar, Prathamesh Wadile , Bhushan Saraf , Omkar Kardile and Aditya Pawar. " IOT Based Heart Disease Prediction." International Journal for Research in Applied Science & Engineering Technology (IJRASET) (2021):2321-9653.
- [2] Megha Vyas , Priya Kunwar, Vinod Joshi, Mihir Maheshwari and Mr. Girish Ameta. "Software for Disease Prediction." International Journal for Research in Applied Science & Engineering Technology(IJRASET) :2321-9653,2020.
- [3] Yuvraj Nikhate and M. V. Jonnalagedda. "Survey On Heart Disease Prediction using Machine Learning." International Journal of Creative Research Thoughts(2020) :2320-2882.
- [4] Raj H. Chauhan, Daksh N. Naik, Rinal A. Halpati, Sagarkumar J. Patel4 and Mr. A.D.Prajapati. " Disease Prediction Using Machine Learning."International Research Journal Of Engineering & Technology(IRJET):23950056,2020.
- [5] Sandhya B , Monika K. J , Vyshnavi and Usman K. " Disease Prediction and Hospital Recommendation using Machine Learning Algorithms." International Journal for Research in Applied Science & Engineering Technology(IJRASET): 2321-9653,2020.
- [6] Lamido Yahaya , Nathaniel David Oye and Etemi Joshua Garba. "A Comprehensive Review on Heart Disease Prediction Using Data Mining and Machine Learning Techniques."American Journal of Artificial Intelligence:2639-9717 ,vol 4, 2020.
- [7] Neelaveni, J., and MS Geetha Devasana. "Alzheimer disease prediction using machine learning algorithms." 2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS). IEEE, 2020
- [8] Terrada, Oumaima, et al. "Atherosclerosis disease prediction using Supervised Machine Learning Techniques." 2020 1st International Conference on Innovative Research in Applied Science, Engineering and Technology (IRASET). IEEE, 2020.
- [9] Fitriyani, Norma Latif, et al. "HDPM: An Effective Heart Disease Prediction Model for a Clinical Decision Support System." IEEE Access 8 (2020): 133034-133050.
- [10] Grampurohit, Sneha, and Chetan Sagarnal. "Disease Prediction using Machine Learning Algorithms." 2020 International Conference for Emerging Technology (INCET). IEEE,

APPENDIX

1. Importing the Dependencies

```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
```

```
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
```

2. Data Collection and Processing

```
# loading the csv data to a Pandas DataFrame
```

```
heart_data = pd.read_csv('/content/data.csv')
```

```
# print first 5 rows of the dataset
```

```
heart_data.head()
```

	age	sex	cp	trestbps	cho1	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
1	37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
2	41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
3	56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
4	57	0	0	120	354	0	1	163	1	0.6	2	0	2	1

```
# print last 5 rows of the dataset
```

```
heart_data.tail()
```

	age	sex	cp	trestbps	cho1	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
298	57	0	0	140	241	0	1	123	1	0.2	1	0	3	0
299	45	1	3	110	264	0	1	132	0	1.2	1	0	3	0
300	68	1	0	144	193	1	1	141	0	3.4	1	2	3	0
301	57	1	0	130	131	0	1	115	1	1.2	1	1	3	0
302	57	0	1	130	236	0	0	174	0	0.0	1	1	2	0

```
# number of rows and columns in the dataset
```

```
heart_data.shape
```

```
(303, 14)
```

```
# getting some info about the data
```

```
heart_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
```

```

RangeIndex: 303 entries, 0 to 302
Data columns (total 14 columns):
#   Column      Non-Null Count  Dtype
---  -
0   age         303 non-null    int64
1   sex         303 non-null    int64
2   cp          303 non-null    int64
3   trestbps    303 non-null    int64
4   chol        303 non-null    int64
5   fbs         303 non-null    int64
6   restecg     303 non-null    int64
7   thalach     303 non-null    int64
8   exang       303 non-null    int64
9   oldpeak     303 non-null    float64
10  slope       303 non-null    int64
11  ca          303 non-null    int64
12  thal        303 non-null    int64
13  target      303 non-null    int64
dtypes: float64(1), int64(13)
memory usage: 33.3 KB

```

```
# checking for missing values
```

```
heart_data.isnull().sum()
```

```

age         0
sex         0
cp          0
trestbps    0
chol        0
fbs         0
restecg     0
thalach     0
exang       0
oldpeak     0
slope       0
ca          0
thal        0
target      0

```

```
dtype: int64
```

```
# statistical measures about the data
```

```
heart_data.describe()
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
count	303 .00 000 0	303 .00 000 0	303 .00 000 0	303 .00 000 0	303 .00 000 0	303 .00 000 0	303 .00 000 0	303 .00 000 0	303 .00 000 0	303 .00 000 0	303 .00 000 0	303 .00 000 0	303 .00 000 0	303 .00 000 0
me	54.	0.6	0.9	131	246	0.1	0.5	149	0.3	1.0	1.3	0.7	2.3	0.5

an	366 337	831 68	669 97	.62 376 2	.26 402 6	485 15	280 53	.64 686 5	267 33	396 04	993 40	293 73	135 31	445 54
std	9.0 821 01	0.4 660 11	1.0 320 52	17. 538 143	51. 830 751	0.3 561 98	0.5 258 60	22. 905 161	0.4 697 94	1.1 610 75	0.6 162 26	1.0 226 06	0.6 122 77	0.4 988 35
min	29. 000 000	0.0 000 00	0.0 000 00	94. 000 000	126 .00 000 0	0.0 000 00	0.0 000 00	71. 000 000	0.0 000 00	0.0 000 00	0.0 000 00	0.0 000 00	0.0 000 00	0.0 000 00
25 %	47. 500 000	0.0 000 00	0.0 000 00	120 .00 000 0	211 .00 000 0	0.0 000 00	0.0 000 00	133 .50 000 0	0.0 000 00	0.0 000 00	1.0 000 00	0.0 000 00	2.0 000 00	0.0 000 00
50 %	55. 000 000	1.0 000 00	1.0 000 00	130 .00 000 0	240 .00 000 0	0.0 000 00	1.0 000 00	153 .00 000 0	0.0 000 00	0.8 000 00	1.0 000 00	0.0 000 00	2.0 000 00	1.0 000 00
75 %	61. 000 000	1.0 000 00	2.0 000 00	140 .00 000 0	274 .50 000 0	0.0 000 00	1.0 000 00	166 .00 000 0	1.0 000 00	1.6 000 00	2.0 000 00	1.0 000 00	3.0 000 00	1.0 000 00
ma x	77. 000 000	1.0 000 00	3.0 000 00	200 .00 000 0	564 .00 000 0	1.0 000 00	2.0 000 00	202 .00 000 0	1.0 000 00	6.2 000 00	2.0 000 00	4.0 000 00	3.0 000 00	1.0 000 00

```
# checking the distribution of Target Variable
heart_data['target'].value_counts()
```

```
1    165
0    138
Name: target, dtype: int64
1 --> Defective Heart
0 --> Healthy Heart
```

3. Splitting the Features and Target

```
X = heart_data.drop(columns='target', axis=1)
Y = heart_data['target']
```

```
print(X)
```

```
   age  sex  cp  trestbps  chol  ...  exang  oldpeak  slope  ca  thal
0    63   1   3      145   233  ...      0       2.3      0   0    1
1    37   1   2      130   250  ...      0       3.5      0   0    2
2    41   0   1      130   204  ...      0       1.4      2   0    2
3    56   1   1      120   236  ...      0       0.8      2   0    2
4    57   0   0      120   354  ...      1       0.6      2   0    2
..  ...  ...  ..      ...   ...  ...      ...      ...      ...  ..  ...
298  57   0   0      140   241  ...      1       0.2      1   0    3
299  45   1   3      110   264  ...      0       1.2      1   0    3
300  68   1   0      144   193  ...      0       3.4      1   2    3
301  57   1   0      130   131  ...      1       1.2      1   1    3
302  57   0   1      130   236  ...      0       0.0      1   1    2
```

```
[303 rows x 13 columns]
```

```
print(Y)
```

```
0      1
1      1
2      1
3      1
4      1
..
298    0
299    0
300    0
301    0
302    0
```

```
Name: target, Length: 303, dtype: int64
```

Splitting the Data into Training data & Test Data

```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2,
stratify=Y, random_state=2)
```

```
print(X.shape, X_train.shape, X_test.shape)
```

```
(303, 13) (242, 13) (61, 13)
```

4. Model Training

4.1 Logistic Regression

```
model = LogisticRegression()
```

```
# training the LogisticRegression model with Training data
```

```
model.fit(X_train, Y_train)
```

```
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
intercept_scaling=1, l1_ratio=None, max_iter=100,
multi_class='auto', n_jobs=None, penalty='l2',
random_state=None, solver='lbfgs', tol=0.0001, verbose=0,
warm_start=False)
```

4.2 K Nearest Neighbor

```
#Fitting K-NN classifier to the training set
```

```
from sklearn.neighbors import KNeighborsClassifier
```

```
classifier= KNeighborsClassifier(n_neighbors=5, metric='minkowski', p=2 )
```

```
classifier.fit(x_train, y_train)
```

4.3 Random Forest

#Fitting Random Forest classifier to the training set

```
from sklearn.ensemble import RandomForestClassifier
```

```
classifier= RandomForestClassifier(n_estimators= 10, criterion="entropy")
```

```
classifier.fit(x_train, y_train)
```

4.4 Decision Tree

#Fitting Decision Tree classifier to the training set

```
From sklearn.tree import DecisionTreeClassifier
```

```
classifier= DecisionTreeClassifier(criterion='entropy', random_state=0)
```

```
classifier.fit(x_train, y_train)
```

5. Model Evaluation

Accuracy Score

accuracy on training data

```
X_train_prediction = model.predict(X_train)
```

```
training_data_accuracy = accuracy_score(X_train_prediction, Y_train)
```

```
print('Accuracy on Training data : ', training_data_accuracy)
```

```
Accuracy on Training data :  0.8512396694214877
```

accuracy on test data

```
X_test_prediction = model.predict(X_test)
```

```
test_data_accuracy = accuracy_score(X_test_prediction, Y_test)
```

```
print('Accuracy on Test data : ', test_data_accuracy)
```

```
Accuracy on Test data :  0.819672131147541
```

6. Building a Predictive System

```
input_data = (62,0,0,140,268,0,0,160,0,3.6,0,2,2)
```

```
# change the input data to a numpy array
```

```
input_data_as_numpy_array= np.asarray(input_data)

# reshape the numpy array as we are predicting for only on instance
input_data_reshaped = input_data_as_numpy_array.reshape(1,-1)

prediction = model.predict(input_data_reshaped)

print(prediction)

if (prediction[0]== 0):
    print('The Person does not have a Heart Disease')
else:
    print('The Person has Heart Disease')

[0]
The Person does not have a Heart Disease
```


With growing development in the field of medical science alongside machine learning various experiments and researches has been carried out in these recent years releasing the relevant significant papers. The paper [1] propose heart disease prediction using KStar, J48, SMO, and Bayes Net and Multilayer perceptron using WEKA software. Based on performance from different factor SMO (89% of accuracy) and Bayes Net (87% of accuracy) achieve optimum performance than KStar, Multilayer perceptron and J48 techniques using k-fold cross validation. The accuracy performance achieved by those algorithms are still not satisfactory. So that if the performance of accuracy is improved more to give batter decision to diagnosis disease. [2]In a research conducted using Cleveland dataset for heart diseases which contains 303 instances and used 10-fold Cross Validation, considering 13 attributes, implementing 4 different algorithms, they concluded Gaussian Naïve Bayes and Random Forest gave the maximum accuracy of 91.2 percent. [3]Using the similar dataset of Framingham, Massachusetts, the experiments were carried out using 4 models and were trained and tested with maximum accuracy K Neighbors Classifier: 87%, Support Vector Classifier: 83%, Decision Tree Classifier: 79% and Random Forest Classifier: 84

