Disease Prediction Application Using Machine Learning

Jagruthi sangi

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING

AURORA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with Grade “A” by NAAC

ABSTRACT

The health care systems collects data and reports from the hospitals or patient's database by machine learning and data processing techniques which is employed to predict the disease so as to create reports supported the results which used for various kinds of predictions for disease and which is that the leading explanation for the human's death since past years. Medical reports and data had been extracted from various databases to predict a number of the required diseases which are commonly found in people nowadays breast cancer, heart disease and diabetes disease and make their life more critical to measure. Nowadays technology advancement within the health care industry has been helping people to create their process easier by suggesting hospitals and doctors to travel to for his or her treatment, where to admit and which hospitals are the simplest for the treating the desired disease. we've implemented this sort of system in our application to form people’s life simpler by predicting the disease by inputting certain data from their reports which can give the result positive or negative supported the disease prediction they are going to be having a choice to get recommendation of best hospitals with best doctors nearby from the past users or guardians..

LIST OF FIGURES

LIST OF TABLES

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

Properly analyzing clinical documents about patients' health anticipate the possibility of occurrence of various diseases. In addition, acquiring information regarding specialists of that particular disease as per the requirement facilitates proper and efficient diagnosis. This Project provides a novel method that uses data mining technique, namely, Logistic regression and random forest classification algorithm for prediction of disease. Using medical profiles such as heart rate, blood pressure through sensors and other externally observable symptoms such as fever, cold, headache etc. that patient has, prediction of likelihood of a disease is done. Logistic regression and random forest classification algorithm takes these symptoms and predicts disease. Furthermore, all the needful and adequate information regarding the predicted disease as well as the recommended doctors is provided. Recommendation (Future implementation) suggests the location , contact and other necessary details of the disease specialists based on the filters chosen by the user out of less fees, more experience, nearest location and feedback reviews of the doctors.

algorithm. Thus user can get appropriate treatment and necessary medical advice as fast as possible. Additionally, users provide their feedback for the recommended doctors which are then added for analysis in order to make further recommendations based on reviews.

Healthcare industry generates terabytes of data every year. The medical documents maintained are a pool of information regarding patients. The task of extracting useful in formation or quality healthcare is tricky and important. By analyzing these voluminous data we can predict the occurrence of the disease and safe guard people. Thus, an intelligent system for disease prediction plays a major role in controlling the disease and maintaining the good health status for people by providing accurate and trustworthy disease risk prediction.

CHAPTER 2

LITERATURE SURVEY

SL NO.	AUTHOR	YEAR	DESCRIPTION	PROS	CONS
1	M. Denil, D. Matheson, and N. De Freitas	2014	explored the relation between the willingness to recommend the hospital and other satisfaction identifiers. Author : Klink Enberg This paper finds that hospitals that focus on resources to improve aspects of care such as nurses and physician respect, respect, obedience, room hygiene, etc. This paper does not look at patient data. The literature in the HCAHPS database analysis is largely driven by hypothesis and only looks at specific aspects of patient satisfaction or census.	This paper focuses on proper hospital Hygiene and staff.	It does not focus on the patients point of view on easy access and bes choice of hospital.
2	Watson, F. Marir	1994	Using retrospect, they concluded that non-Spanish whites on average tend to go to hospitals that offer a better patient experience for all patients compared to	It provides the idea that patients always prefer best hospital first.	It lacks a technological easy access to the best hospitals.

			hospitals commonly used by African American, Hispanic, Asian / Pacific Islander, or multiracial patients		
3	W. Bergerud	1996	Investigated the relationship between postoperative morbidity and mortality and patients' perspectives of care in surgical patients. Author: Sheetz et al In their article, the satisfaction points used were used along with the registered Registry of the Michigan Collaborative Clinic as a measure of patient care ideas. A few studies have examined the relationship between one satisfaction question and one or more patient information.	It used the patients perspective to make improvement for the hospital.	Its lacks in doctors point of view who is the most imp in these systems.
4	Binal T. et al	2010	Healthcare decision support system forswine flu prediction using naïve bayes classifier, focuses onthe aspect of medical diagnosis by learning patterns throughthe	Healthcare support system for swine flue.	It lacks in accuracy in prediction of the disease.

			collected data for swine flu using naïve bayes classifierfor classifying the patients of swine flu into three categories(least possible, probable or most probable), resulting into anaccuracy of nearly 63.33%. Datasets used for thisclassification were limited in number		

CHAPTER 3

AIM AND SCOPE OF THE PRESENT INVESTIGATION

When we see around there are many patients that does not get the right treatment at the right time because of their lack of decision taking about the choice of hospital and doctors, they don’t know what you do now and end up very serious at the end.
The objective of the project is to provide the service to patients by suggesting them the best hospital to find their cure for their existing disease . The project is to provide a very easy solution for the patients to get recommendation to what doctor or hospital they need to go after diagnosed with a severe disease.This web application can find the solution to that, no need of thinking about what should be done after diagnosed with a severe disease. This web application handles reports to make predictions and give results accordingly to that, a best hospitals can be selected for their treatment and more lives can be saved.

The scope of the project is to provide a very easy solution for the patients to get recommendation to what doctor or hospital they need to go after diagnosed with a severe disease. In this project we will be using web development to develop a web application that will help us to achieve our target and with machine learning algorithms to predict the disease by using random forest classifier algorithm and recommend the best doctors and the best hospitals nearby by using collaborative filtering technique.

CHAPTER 4

EXPERIMENT OR MATERIALS AND METHODS, ALGORITHMS USED

4.1 EXISTING AND PROPOSED SYSTEM

4.1.1 EXISTING SYSTEM

- Several online health care system has invented new ideas to benefit people and so many online applications have features to give recommendations on hospital and doctors.
- But they have lack of reliability and accuracy where they need to do improvisations in the features and modules. Genuinely health care systems might not upload the opinions of people in some cases for the negative response and by doing manually while collecting feedbacks from the patients, might be patients hesitate to give complete opinion of doctors or hospitals in front of persons where we will find the lack of quality.
- In total we have not found all features and modules at a time in one application and there are different types of applications for different type of diseases where they have different applications separately for doctors and hospitals to give recommendations.

Disadvantages of Existing system:

1) Difficulties in finding the best doctors for a particular disease.
2) Tough to discover the hospitals based on the recommendation.

4.1.2 PROPOSED WORK

- In this research we have found the solution for the issues facing in existing system where we have proposed the accuracy, reliability and efficiency by developing the features of three diseases called Heart disease, cancer disease and diabetes where we will find most common diseases in people health and we have installed in one application with prediction of three diseases by analysing the symptoms collected from the patient's record and taking positive and negative opinions from patient's according to that we will give ratings to the hospitals and doctors from best to worst.
- Guardians opinions is also very much important and they can give feedback of them like how they were treating their patients? Was it friendly or strictly? And how the hospital management is? Was it clean? How is the hospitality ? When the feedback comes to online so that patients and guardians can give both positive and negative opinions completely without any hesitation.
- Based on that we can provide truthful recommendations of hospital and doctors for the people and can predict the results. According to that prediction of particular disease we will predict best suitable hospital and doctor to consult and to get admit into it.

Advantages Of Proposed System:

1) Easy way of accessing the application with best recommendations on both Hospitals and doctors.
2) Application has multiple options to make decision easily on diseases

4.1.3 SOFTWARE AND HARDWARE CONFIGURATIONS

Hardware: Desktop or Laptop or present generation devices.

Hardware requirements

Hardware	**Minimum requirements**
Computer	4 GHz minimum, multi-core processor
Memory (RAM)	At least 4GB, preferably higher, and 8ommensurate with concurrent usage
Hard disk space	At least 10 GB

Table 4.1: Hardware Requirements.

Software requirements

Software	**Minimum requirements**
Operating system	Windows Server 2012 R2 or above
Microsoft .Net Framework v4.6.1 (or higher)	The HelpMaster Web Portal has been written to use Microsoft IIS ASP.NET technology and as such requires the machine that IIS is running on to have the Microsoft .NET v4.6.1 (or higher) Framework installed as well as the ASP.Net 4.5 and .Net Extensibility 4.5 features enabled.

Table 4.2: Software Requirements.

- Hardware : Desktop or Laptop.
- Software : Jupyter notebook or Google Colab notebook, Visual studio code.
- Programming Language : Python Programming Language, PHP, HTML, CSS, Javascript.

4.2 DATASET DETAILS

4.2.1 DATASET NAMES

A data set (or dataset) is a collection of data. In the case of tabular data, a data set corresponds to one or more database tables, where every column of a table represents a particular variable, and each row corresponds to a given record of the data set in question. The data set lists values for each of the variables, such as height and weight of an object, for each member of the data set. Data sets can also consist of a collection of documents or files. The sources of the datasets are from Kaggle.com.

The datasets that are used are:

- Heart disease dataset – (heart_disease.csv)
- Diabetes disease dataset – (diabetes_disease.csv)
- Breast cancer dataset – (breast_cancer.csv)

4.2.2 HEART DISEASE DATASET DETAILS

The heart disease datasets consists of 303 rows 14 columns.

The fields: age, sex, cp, trestbp,s chol, fbs, restecg, thalach, exan,g oldpeak, slope, ca, thal. We have taken the 14th column as the target variable (target).

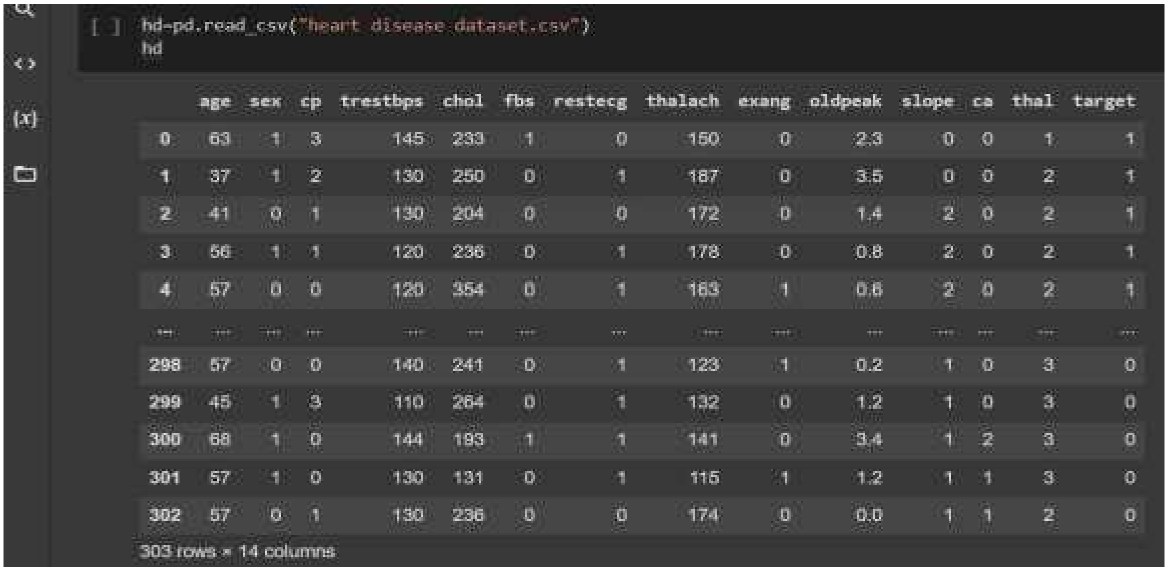
4.2.2.1 TABLE OF DATASET FIELDS

Column	**Non-Null Count**	**Data types**
Age	303 non-null	Int 64

Sex	303 non-null	Int 64
Cp	303 non-null	Int 64
Threstbps	303 non-null	Int 64
Chol	303 non-null	Int 64
Fbs	303 non-null	Int 64
Restecg	303 non-null	Int 64
Exang	303 non-null	Int 64
Oldpeak	303 non-null	Int 64
Slope	303 non-null	Int 64
Ca	303 non-null	Int 64
Thal	303 non-null	Int 64
Target	303 non-null	Int 64

Table 4.3: Heart disease data set fields.

DATA SET VISUALIZATION



```
[ ] hd-pd.read_csv("heart disease dataset.csv")
    hd
```

```
     age  sex  cp  trestbps  chol  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
...  ...  ...  ...      ...   ...  ...      ...      ...    ...      ...    ...  ...  ...     ...
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
303 rows × 14 columns
```

Fig 4.1: Heart disease dataset details.

4.2.3 DIABETES DISEASE DATASET DETAILS

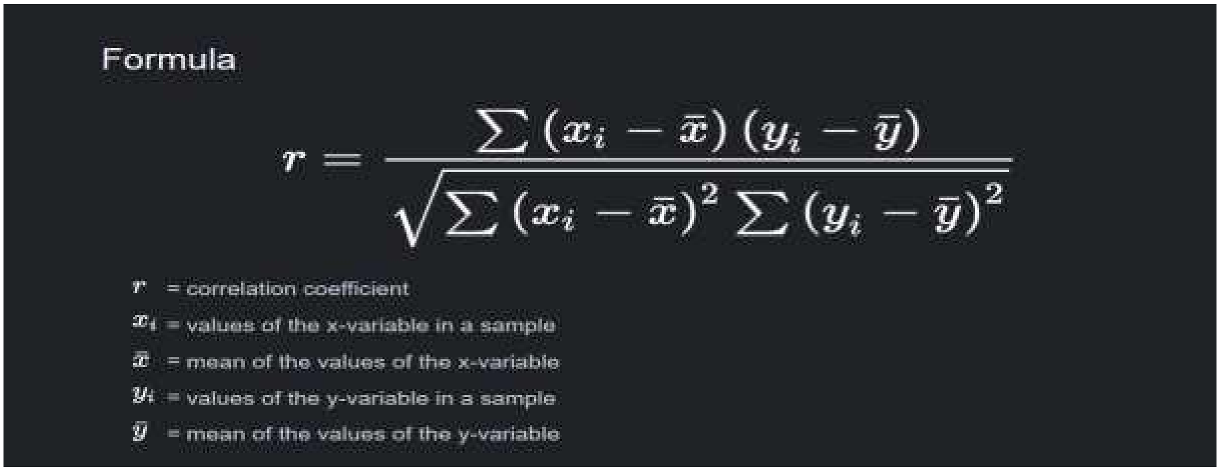
The diabetes disease dataset consists of 768 rows, 9 columns
Column field names : Pregnancies, Glucose , BloodPressure, SkinThickness, Insulin, BMI, DiabetesPedigreeFunction, Age.The last column is the target (outcome).

TABLE OF DATASET FIELDS

Column	**Non-Null count**	**Data Type**
Pregnancies	768 non-null	int64
Glucose	768 non-null	int64
Blood pressure	768 non-null	int64
Skin Thickness	768 non-null	int64
Insulin	768 non-null	int64
BMI	768 non-null	float64
Diabeta pedigree Function	768 non-null	float64
Age	768 non-null	int64
Outcome	768 non-null	int64

Table 4.4: Diabetes disease data set fields.

From the above 7 input fields we are only choosing the two input fields [Glucose, BMI] based on Correlation Pearson method.



Formula

$$r = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum (x_i - \bar{x})^2 \sum (y_i - \bar{y})^2}}$$

r = correlation coefficient

x_i = values of the x-variable in a sample

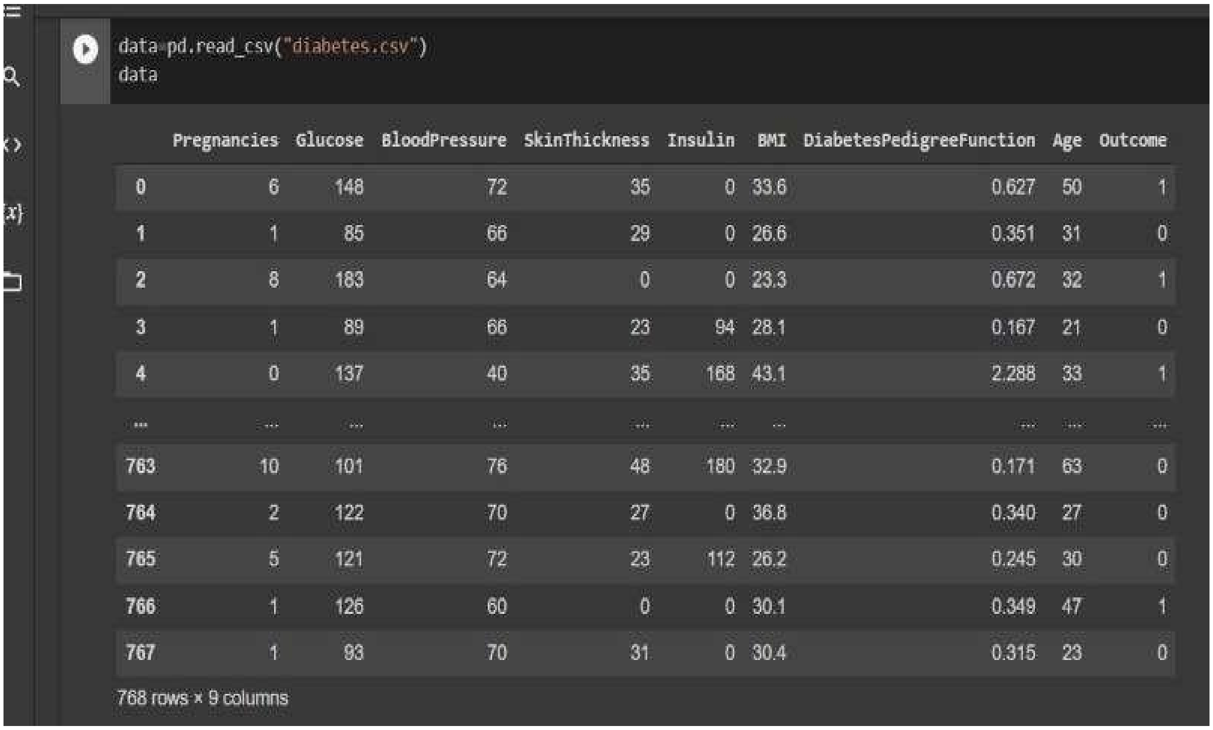
$\bar{x}$ = mean of the values of the x-variable

y_i = values of the y-variable in a sample

$\bar{y}$ = mean of the values of the y-variable

Fig 4.2: Correlation formula.

DATA SET VISUALIZATION



```
data=pd.read_csv("diabetes.csv")
data
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1
...	...	...	...	...	...	...	...	...	...
763	10	101	76	48	180	32.9	0.171	63	0
764	2	122	70	27	0	36.8	0.340	27	0
765	5	121	72	23	112	26.2	0.245	30	0
766	1	126	60	0	0	30.1	0.349	47	1
767	1	93	70	31	0	30.4	0.315	23	0

768 rows × 9 columns

Fig 4.3: Diabetes disease dataset details.

4.2.4 BREAST CNACER DATASET DETAILS

The breast cancer dataset consists of 569 rows, 31 columns.

Column field names : radius_mean, texture_mean, perimeter_mean, area_mean, smoothness_mean, compactness_mean, concavity_mean, concave points_mean, symmetry_mean, fractal_dimension_mean, radius_se, texture_se, perimeter_se, area_se, smoothness_se, compactness_se, concavity_se, concave points_se, symmetry_se, fractal_dimension_se, radius_worst, texture_worst, perimeter_worst, area_worst, smoothness_worst, compactness_worst, concavity_worst, concave points_worst, symmetry_worst, fractal_dimension_worst. Diagnosis will be the target variable (M=1, B=0)

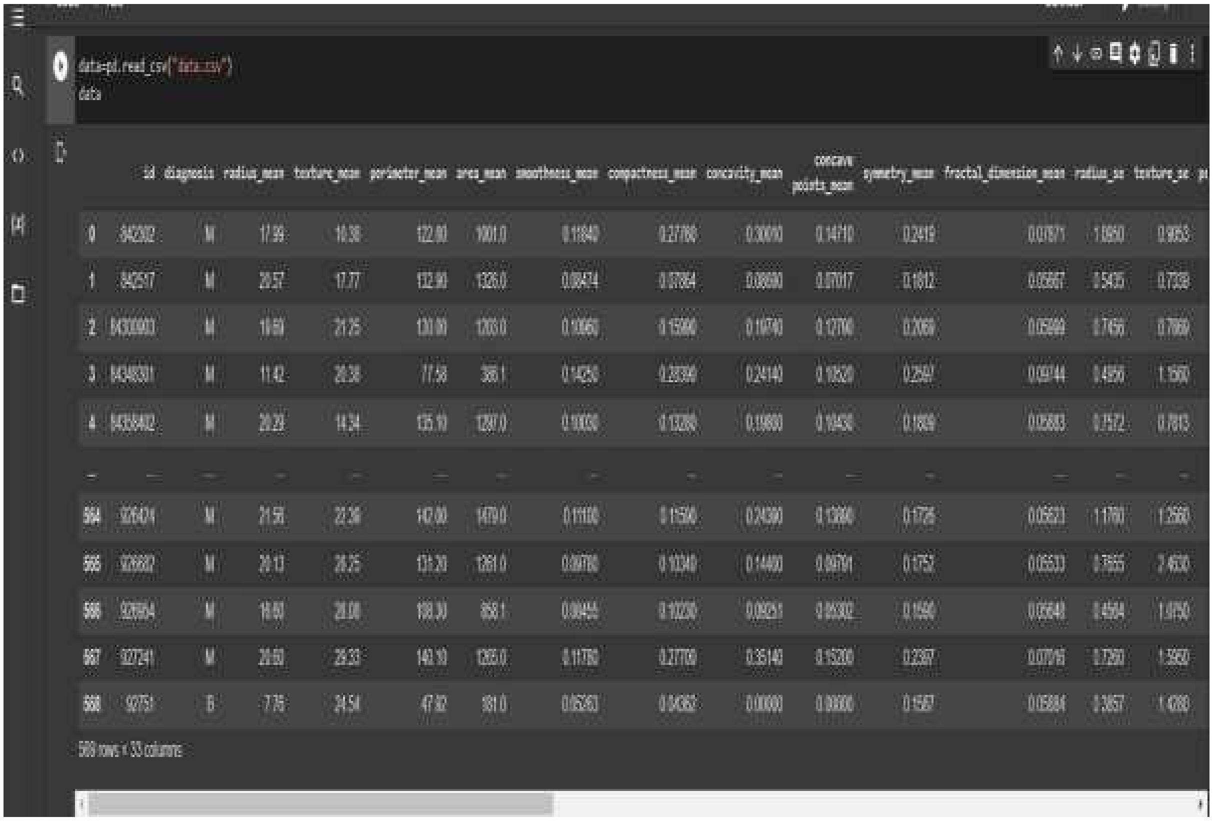
TABLE OF DATASET FIELDS

Column	**Non-Null count**	**Data types**
Diagnosis	569 non-null	float64
radius_mean	569 non-null	float64
texture_mean	569 non-null	float64
perimeter_mean	569 non-null	float64
area_mean	569 non-null	float64
smoothness_mean	569 non-null	float64
compactness_mean	569 non-null	float64
concavity_mean	569 non-null	float64
concave points_mean	569 non-null	float64
symmetry_mean	569 non-null	float64
fractal_dimension_mean	569 non-null	float64
radius_se	569 non-null	float64
texture_se	569 non-null	float64
perimeter_se	569 non-null	float64
area_se	569 non-null	float64
smoothness_se	569 non-null	float64
compactness_se	569 non-null	float64

concavity_se	569 non-null	float64
concave points_se	569 non-null	float64
symmetry_se	569 non-null	float64
fractal_dimension_se	569 non-null	float64
radius_worst	569 non-null	float64
texture_worst	569 non-null	float64
perimeter_worst	569 non-null	float64
area_worst	569 non-null	float64
smoothness_worst	569 non-null	float64
compactness_worst	569 non-null	float64
concavity_worst	569 non-null	float64
concave points_worst	569 non-null	float64
symmetry_worst	569 non-null	float64
fractal_dimension_worst	569 non-null	float64

Table 4.5: Breast cancer disease dataset details.

DATA SET VISUALIZATION



```
data=pd.read_csv("data.csv")
data
```

```
                                                                                                                 concave
      id        diagnosis  radius_mean  texture_mean  perimeter_mean  area_mean  smoothness_mean  compactness_mean  concavity_mean  points_mean  symmetry_mean  fractal_dimension_mean  radius_se  texture_se
0     842302    M          17.99        10.38         122.80          1001.0     0.11840          0.27760           0.30010         0.14710      0.2419         0.07871                 1.0950     0.9053
1     842517    M          20.57        17.77         132.90          1326.0     0.08474          0.07864           0.08690         0.07017      0.1812         0.05667                 0.5435     0.7339
2     84300903  M          19.69        21.25         130.00          1203.0     0.10960          0.15990           0.19740         0.12790      0.2069         0.05999                 0.7456     0.7869
3     84348301  M          11.42        20.38         77.58           386.1      0.14250          0.28390           0.24140         0.10520      0.2597         0.09744                 0.4956     1.1560
4     84358402  M          20.29        14.34         135.10          1297.0     0.10030          0.13280           0.19800         0.10430      0.1809         0.05883                 0.7572     0.7813
...   ...       ...        ...          ...           ...             ...        ...              ...               ...             ...          ...            ...                     ...        ...
564   926424    M          21.56        22.39         142.00          1479.0     0.11100          0.11590           0.24390         0.13890      0.1726         0.05623                 1.1760     1.2560
565   926682    M          20.13        28.25         131.20          1261.0     0.09780          0.10340           0.14400         0.09791      0.1752         0.05533                 0.7655     2.4630
566   926954    M          16.60        28.08         108.30          858.1      0.08455          0.10230           0.09251         0.05302      0.1590         0.05648                 0.4564     1.0750
567   927241    M          20.60        29.33         140.10          1265.0     0.11780          0.27700           0.35140         0.15200      0.2397         0.07016                 0.7260     1.5950
568   92751     B          7.76         24.54         47.92           181.0      0.05263          0.04362           0.00000         0.00000      0.1587         0.05884                 0.3857     1.4280

569 rows × 33 columns
```

Fig 4.4: Dataset used

4.3 METHODOLOGY

The user has to input the data where it will be stored in database and then according to their choice the prediction will be made. After collecting the user data from the database and the choice of predicting the disease is to be predicted. If negative then end the process and if positive the user will get hospital recommendations at which their best treatment can be done.

SYSTEM ARCHITECTURE

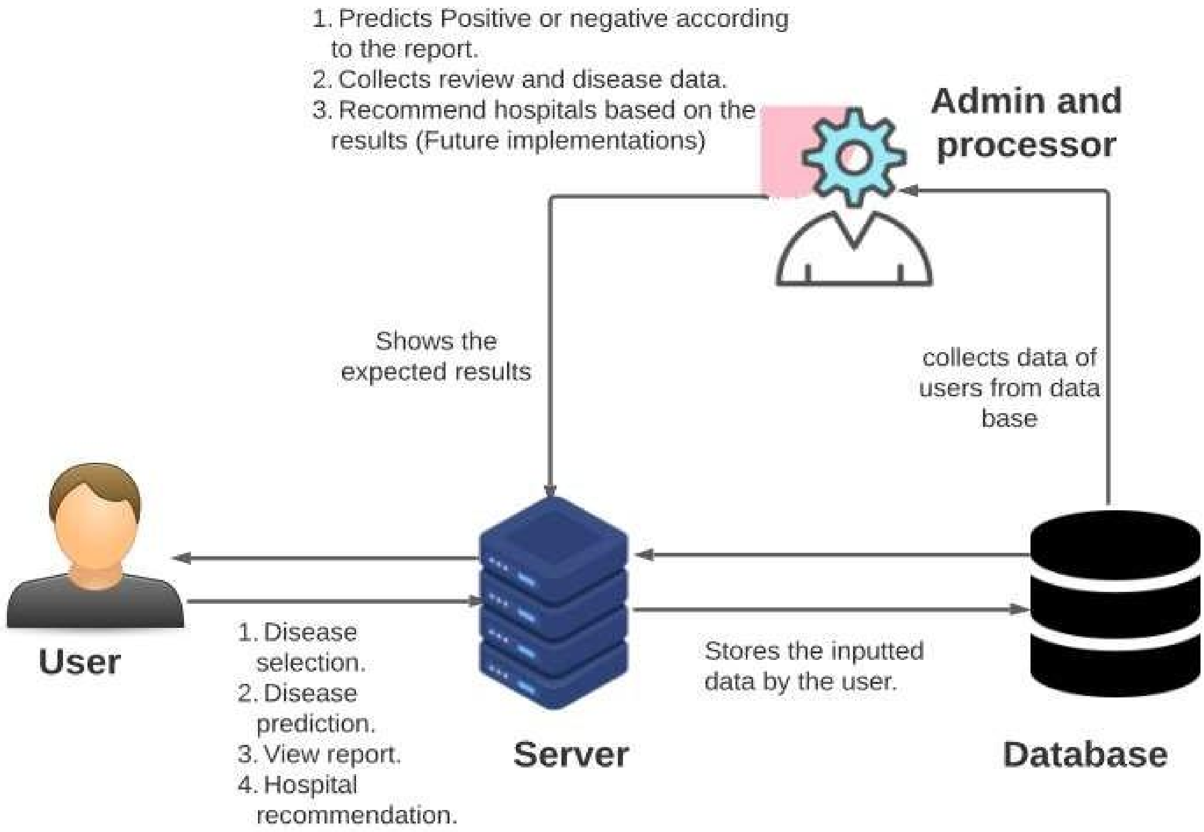

1. Predicts Positive or negative according to the report.
2. Collects review and disease data.
3. Recommend hospitals based on the results (Future implementations)
Admin and processor
Shows the expected results
collects data of users from data base
User
1. Disease selection.
2. Disease prediction.
3. View report.
4. Hospital recommendation.
Server
Stores the inputted data by the user.
Database

Fig 4.5: System Architecture.

System Architecture design-identifies the general hypermedia structure for the

WebApp. Architecture design is tied to the goals establish for a WebApp, the content to be presented, the users who will visit, and also the navigation philosophy that has been established. Content architecture, focuses on the way within which content objects and structured for presentation and navigation. WebApp architecture, addresses the way the applying is structure to manage user interaction, handle internal processing tasks, effect navigation, and present content. WebApp architecture is defined within the context of the event environment during which the appliance is to be implemented.

MODULES IMPLEMENTED

The user has to input the data where it will be stored in database and then according to their choice the prediction will be made. After collecting the user data from the database and the choice of predicting the disease is to be predicted. If negative then end the process and if positive the user will get hospital recommendations (future) at which their best treatment can be done.

- Application Flowchart.
- Data collection (from the user) to make dataset.
- Importing packages.
- Data pre-processing.
- Data fitting and training.
- Prediction as opted by the user.
- Result or output.

FLOWCHART DIAGRAM

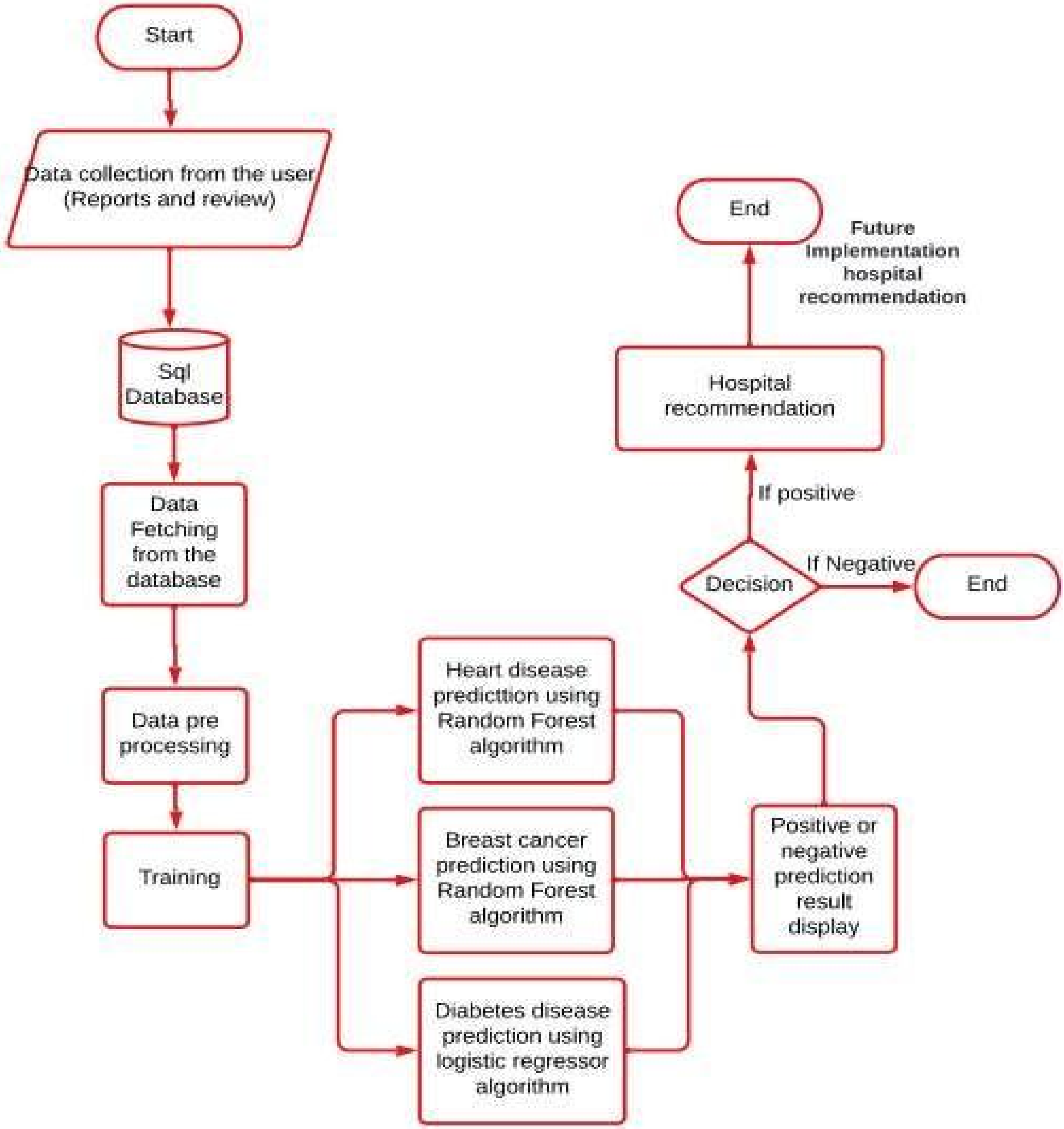


***Fig 4.6: Flowchart Diagram*.**

DATA COLLECTION

Data Collection is one of the most important tasks in building a machine learning model. We collect the specific data based on requirements from users to make the dataset. The dataset contains some unwanted data also. So first we need to pre-process the data and obtain perfect data set for algorithm.

PACKAGES IMPORTED

- Pandas : Pandas is a software library written for python for data manipulation and analysis. It offers data structures and operations for manipulating numerical tables and time series.

- Numpy: It is a library for the Python Programming Language, adding support for large, multiple-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.

- Scikit-learn Package: Scikit-learn is a free machine learning library for python. It features various algorithms like SVM, Random Forest , K-neighbours and Decision Tree.

- Confusion Matrix: A confusion matrix is a technique for summarizing the performance of a classification algorithm. Classification accuracy alone can be misleading if you have an unequal number of observations in each class or if you have more than two classes in your dataset.

  Syntax: from sklearn.metrics import confusion_matrix.

- Classification Report:The classification report visualizer displays the precision, recall, F1, and support scores for the model.

  Syntax: from sklearn.metrics import classification_report.

- Accuracy Score: Accuracy is one metric for evaluating classification models. Informally, accuracy is the fraction of predictions our model got right. Formally, accuracy has the following definition:

  Accuracy = Number of correct predictions / Total number of predictions.

  Syntax: from sklearn.metrics import accuracy_score.

DATA PRE-PROCESSING

It is the gathering of task related information based on some targeted variables to analyse and produce some valuable outcome. However, some of the data may be noisy, i.e. may contain inaccurate values, incomplete values or incorrect values. Hence, it is must to process the data before analysing it and coming to the results. Data pre-processing can be done by data cleaning, data transformation, data selection

Data pre processing is a process of preparing the raw data and making it suitable for a machine learning model. It is the first and crucial step while creating a machine learning model.

When creating a machine learning project, it is not always a case that we come across the clean and formatted data. And while doing any operation with data, it is mandatory to clean it and put in a formatted way. So for this, we use data pre processing task.

A real-world data generally contains noises, missing values, and maybe in an unusable format which cannot be directly used for machine learning models. Data pre processing is required tasks for cleaning the data and making it suitable for a machine learning model which also increases the accuracy and efficiency of a machine learning model.

It involves below steps:

- Getting the dataset
- Importing libraries
- Importing datasets
- Finding Missing Data
- Encoding Categorical Data
- Splitting dataset into training and test set
- Feature scaling

DATA TRAINING

Model fitting is a measure of how well a machine learning model generalizes to similar data to that on which it was trained. A model that is well-fitted produces more accurate outcomes. A model that is overfitted matches the data too closely. A model that is underfitted doesn't match closely enough

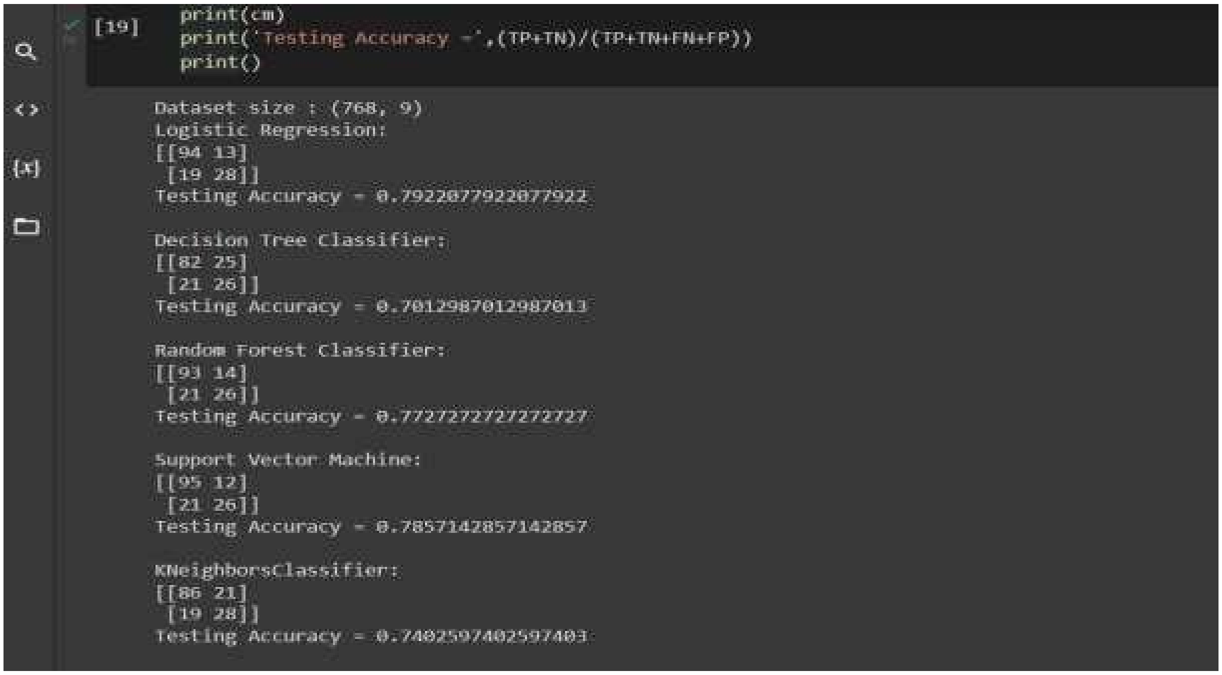
Training data is the initial dataset used to train machine learning algorithms. Models create and refine their rules using this data. It's a set of data samples used to fit the parameters of a machine learning model to training it by example. Training data is also known as training dataset, learning set, and training set. It's an essential component of every machine learning model and helps them make accurate predictions or perform a desired task.

ALGORITHM SELECTION

- The datasets has been tested with different supervised machine learning algorithms and it is found that the best solution with accuracy is given by

1. Diabetes – Logistic regression algorithm
2. Heart disease – Random Forest algorithm
3. Breast Cancer – Random Forest algorithm

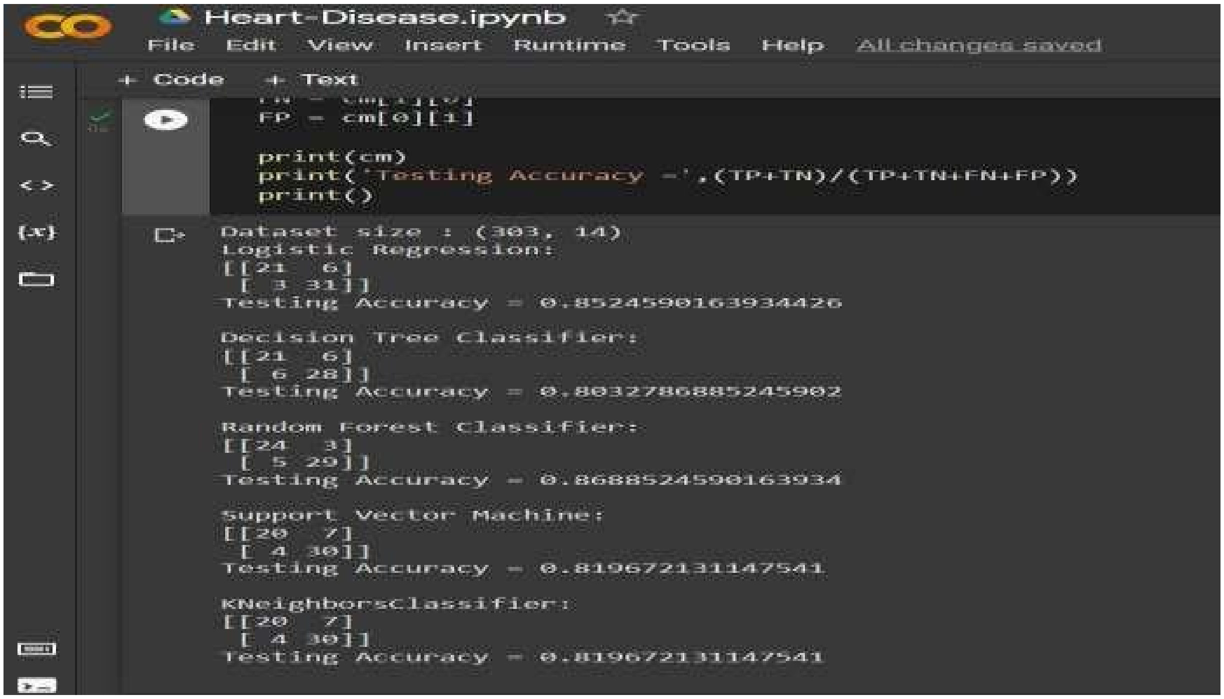
- For hospital recommendation used collaborative filtering algorithm



```
[19]    print(cm)
        print('Testing Accuracy =',(TP+TN)/(TP+TN+FN+FP))
        print()
```

```
Dataset size : (768, 9)
Logistic Regression:
[[94 13]
 [19 28]]
Testing Accuracy = 0.7922077922077922

Decision Tree Classifier:
[[82 25]
 [21 26]]
Testing Accuracy = 0.7012987012987013

Random Forest Classifier:
[[93 14]
 [21 26]]
Testing Accuracy = 0.7727272727272727

Support Vector Machine:
[[95 12]
 [21 26]]
Testing Accuracy = 0.7857142857142857

KNeighborsClassifier:
[[86 21]
 [19 28]]
Testing Accuracy = 0.7402597402597403
```

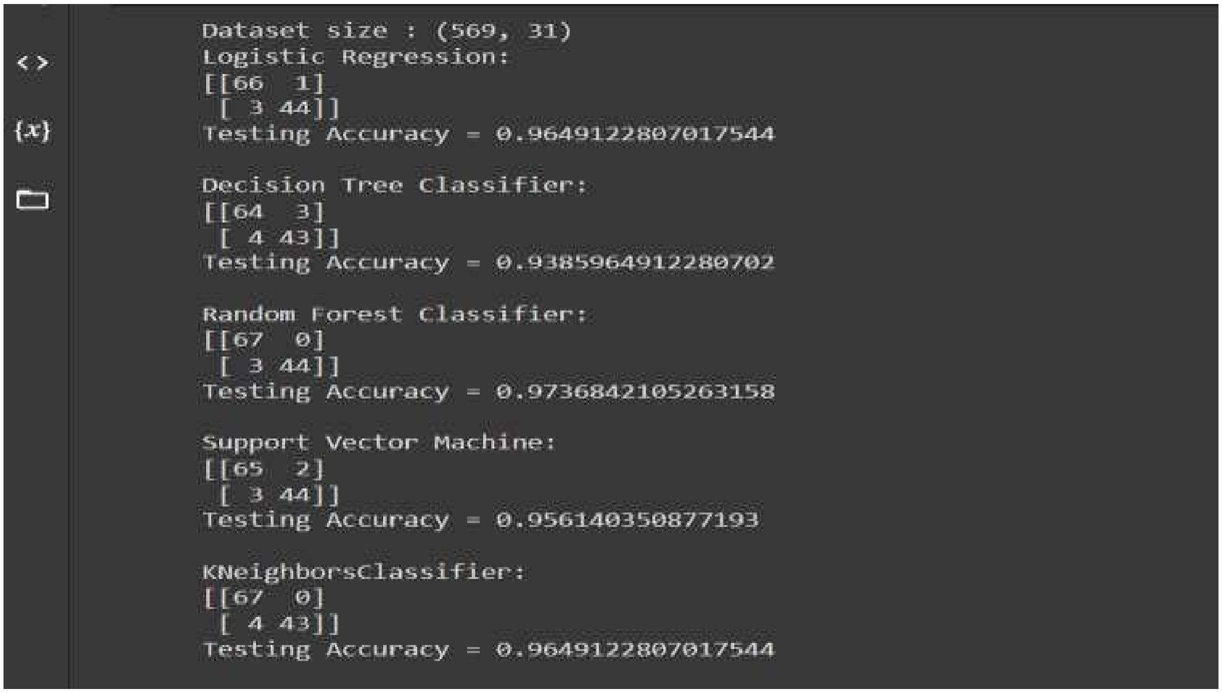
Fig 4.7: Algorithm selection for diabetes disease prediction.



```
FN = cm[1][0]
FP = cm[0][1]

print(cm)
print('Testing Accuracy =',(TP+TN)/(TP+TN+FN+FP))
print()
```

```
Dataset size : (303, 14)
Logistic Regression:
[[21  6]
 [ 3 31]]
Testing Accuracy = 0.8524590163934426

Decision Tree Classifier:
[[21  6]
 [ 6 28]]
Testing Accuracy = 0.8032786885245902

Random Forest Classifier:
[[24  3]
 [ 5 29]]
Testing Accuracy = 0.8688524590163934

Support Vector Machine:
[[20  7]
 [ 4 30]]
Testing Accuracy = 0.819672131147541

KNeighborsClassifier:
[[20  7]
 [ 4 30]]
Testing Accuracy = 0.819672131147541
```

Fig 4.8: Algorithm selection for heart disease prediction.



```
Dataset size : (569, 31)
Logistic Regression:
[[66  1]
 [ 3 44]]
Testing Accuracy = 0.9649122807017544

Decision Tree Classifier:
[[64  3]
 [ 4 43]]
Testing Accuracy = 0.9385964912280702

Random Forest Classifier:
[[67  0]
 [ 3 44]]
Testing Accuracy = 0.9736842105263158

Support Vector Machine:
[[65  2]
 [ 3 44]]
Testing Accuracy = 0.956140350877193

KNeighborsClassifier:
[[67  0]
 [ 4 43]]
Testing Accuracy = 0.9649122807017544
```

Fig 4.9: Algorithm selection for breast cancer prediction.

PREDICTION AS OPTED BY THE USED

- Prediction of disease has three options
  - Heart disease prediction
  - Diabetes disease prediction
  - breast cancer prediction
- Where Random forest algorithm is used for heart and breast cancer prediction and logistic regressor algorithm is used for diabetes prediction for these following algorithm gives the best accuracy rate for these datasets.

4.4 ALGORITHM MODELS

4.4.1 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. Logistic Regression can be used to classify the observations using different types of data and can easily determine the most effective variables used for the classification. The below image is showing the logistic function:

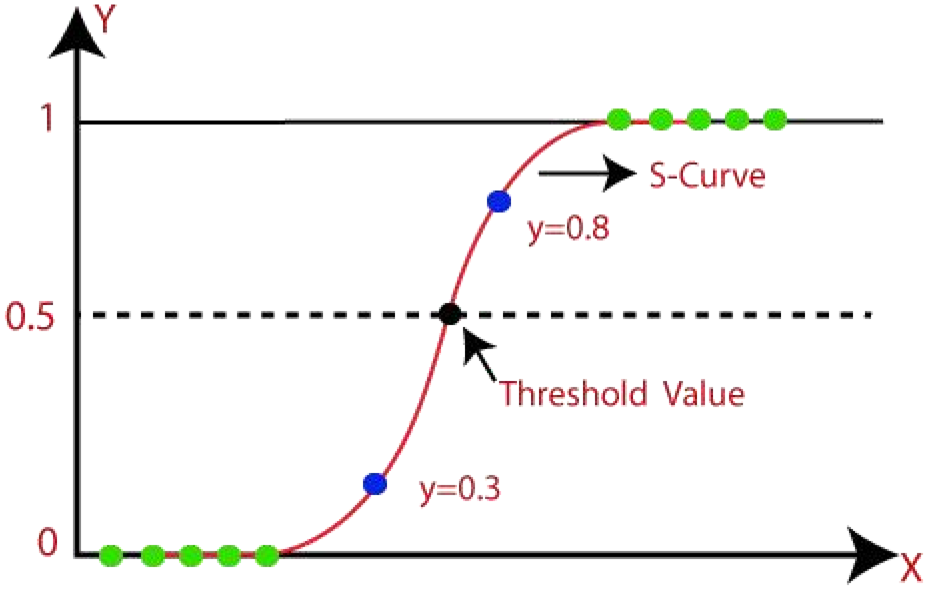


Fig 4.10: Logistic Regression graph.

The sigmoid function is a mathematical function used to map the predicted values to probabilities. It maps any real value into another value within a range of 0 and 1. The value of the logistic regression must be between 0 and 1, which cannot go beyond this limit, so it forms a curve like the "S" form. The S-form curve is called the Sigmoid function or the logistic function.In logistic regression, we use the concept of the threshold value, which defines the probability of either 0 or 1. Such as values above the threshold value tends to 1, and a value below the threshold values tends to 0.

For Diabetes disease prediction: LOGISTIC REGRESSION:

Logistic regression is a supervised learning classification algorithm used to predict the probability of a target variable. The nature of target or dependent variable is dichotomous, which means there would be only two possible classes. ... Mathematically, a logistic regression model predicts P(Y=1) as a function of X.

Syntax: from sklearn.linear_model import LogisticRegression obj=Logistic regression()LOGISTIC REGRESSION

Logistic regression equation:

$P=e\beta0+\beta1X1/1+e\beta0+\beta1X1$

When all the feature plugged in ;

$logit(p)=log(p/(1-p))=\beta0+\beta1*Sexmale+\beta2*age+\beta3*cigsPerDay+\beta4*totChol+\beta5*sysBP+\beta6*glucose$

To implement the Logistic Regression using Python, we will use the same steps as we have done in previous topics of Regression. Below are the steps:

- Data Pre-processing step
- Fitting Logistic Regression to the Training set
- Predicting the test result
- Test accuracy of the result(Creation of Confusion matrix)
- Visualizing the test set result.

4.4.2 RANDOM FOREST ALGORITHM

For heart disease and breast cancer prediction RANDOM FOREST CLASSIFIER: Random forest is a supervised learning algorithm which is used for both classification as well as regression. But however, it is mainly used for classification problems. As we know that a forest is made up of trees and more trees means more robust forest. Similarly, random forest algorithm creates decision trees on data samples and then gets the prediction from each of them and finally selects the best solution by means of voting. It is an ensemble method which is better than a single decision tree because it reduces the over-fitting by averaging the result.

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.

The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.

Implementation Steps are given below:

- Data Pre-processing step
- Fitting the Random forest algorithm to the Training set
- Predicting the test result
- Test accuracy of the result (Creation of Confusion matrix)
- Visualizing the test set result.

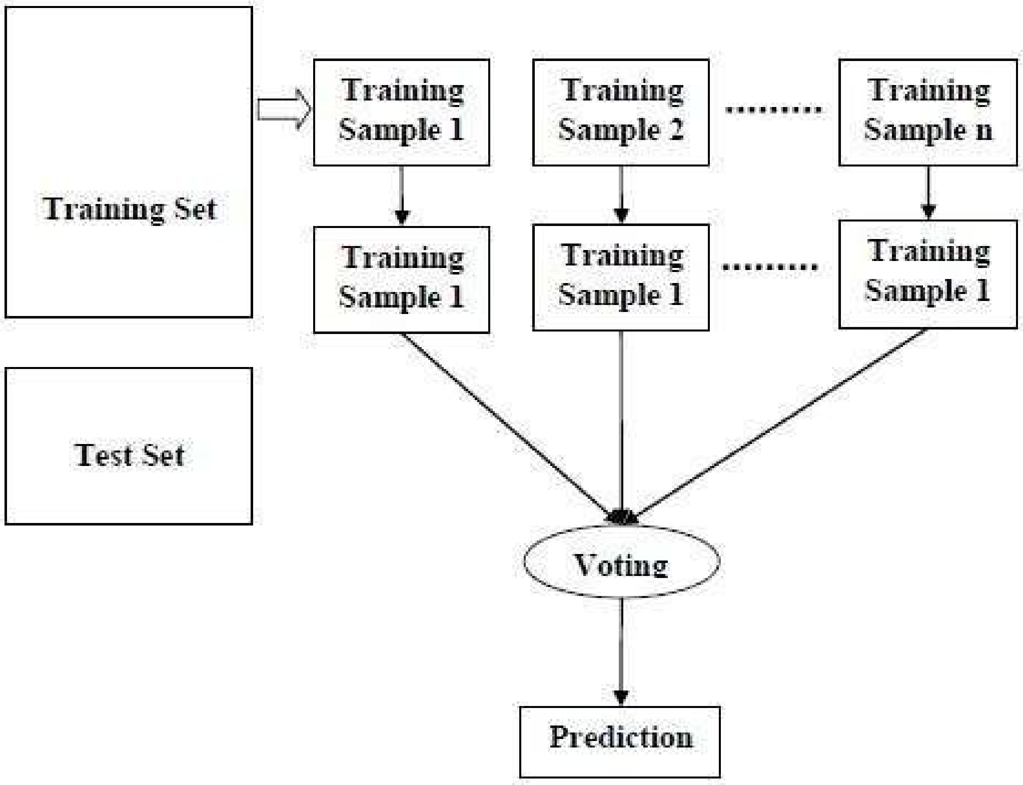


Fig 4.11: Random Forest Algorithm Architecture.

Random Forest works in two-phase first is to create the random forest by combining N decision tree, and second is to make predictions for each tree created in the first phase.

The Working process can be explained in the below steps and diagram:

Step-1: Select random K data points from the training set.

Step-2: Build the decision trees associated with the selected data points (Subsets).

Step-3: Choose the number N for decision trees that you want to build.

Step-4: Repeat Step 1 & 2.

Step-5: For new data points, find the predictions of each decision tree, and assign the new data points to the category that wins the majority votes.

Normal woods can be a gathering of trees. Here, the independence is partitioned into vectors, and each tree gives an underlying stage division called a x distribution. Customary timberlands give a gathering of guaranteed trees to make a fundamental variety of trees, and Breiman picked the best strategy, the technique for cooking or grouping each tree in one of the Random Forests, and Breiman followed the accompanying advances: Randomly organized N archives, yet additionally supplanted, as should be visible from the first numbers, this is a boot test. An illustration of this is tree establishing preparing. In the event that there is another M info, m << M chooses something similar for every hub, and m is a variable chosen from M, so a positive detachment from m addresses the property to be utilized for separation. The consistent worth of m during woods improvement. Each tree develops as large as could be expected. try not to cut. In this manner many trees are brought into the woods; The quantity of trees anticipated by the ntree boundary. The greatest number of factors (m) chose for every hub is again called "mtry" or k. The profundity of the tree can be constrained by hub boundaries (for instance, the quantity of leaves), and now and then by something like one. As referenced above, it streams from every one of the trees that fill in the backwoods to decide the degree of substitution in the wake of preparing or catching the woodland. Each tree gives another example class to casting a ballot. All tree ideas were merged and the greater part (larger part vote) grouping was affirmed at another level. Going on here, the woodland characterizes a tree backwoods assembled utilizing the RI timberland. In the ranger service area, each tree was chosen and a freight test was made for substitution, yet around 1/3 of the first material was absent. This rundown of models is called OOB (Out of pocket) data. Each tree has its own OOB data, which is utilized to look at the breaks in each tree in the timberland, and is known as the OOB break estimation.

CHAPTER 5

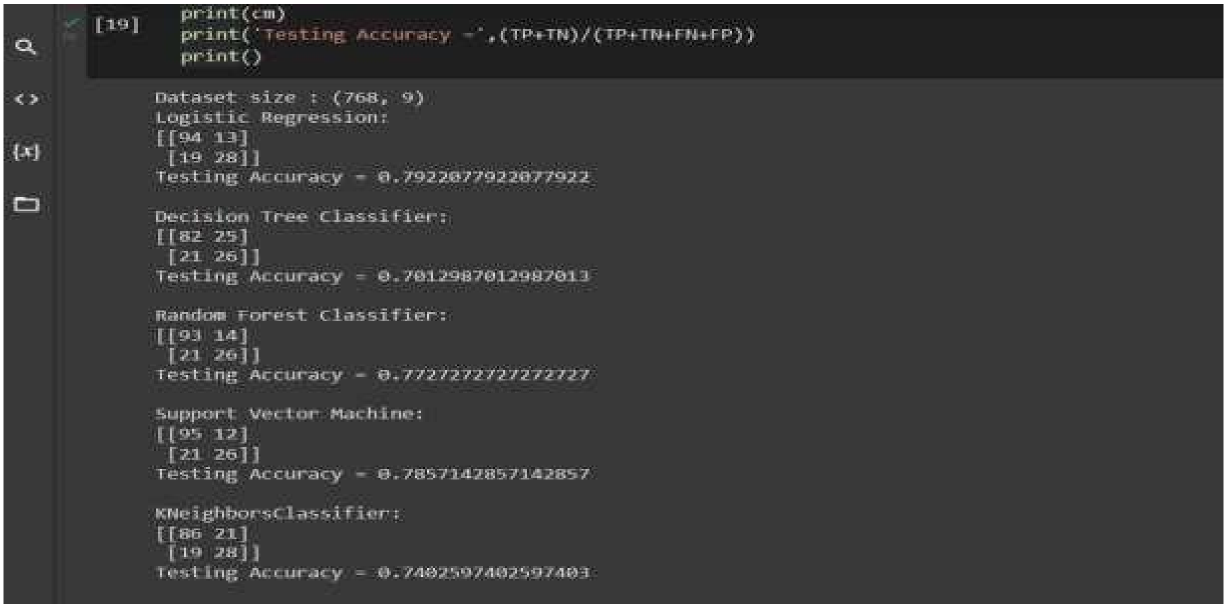
RESULTS AND DISCUSSION

5.1 RESULTS

When we see around there are many patients that does not get the right treatment at the right time because of their lack of decision taking about the choice of hospital and doctors, they don’t know what you do now and end up very serious at the end. The objective of the project is to provide the service to patients by suggesting them the best hospital to find their cure for their existing disease . The project is to provide a very easy solution for the patients to get recommendation to what doctor or hospital they need to go after diagnosed with a severe disease. This web application can find the solution to that, no need of thinking about what should be done after diagnosed with a severe disease. This web application handles reports to make predictions and give results accordingly to that, a best hospitals can be selected more for their treatment and more lives can be saved. After easy login or registering into the app the patient can predict their disease after inputting certain reports from their medical diagnosis report which will display accurately that the patient has the particular disease or not it will show in form of positive or negative. After the Prediction they will be having an option to get recommended hospital which are best for the treatment of their disease nearby. By this way the app can save many more lives more before its too late to get the treatment.

SCREENSHOTS OF RESULT

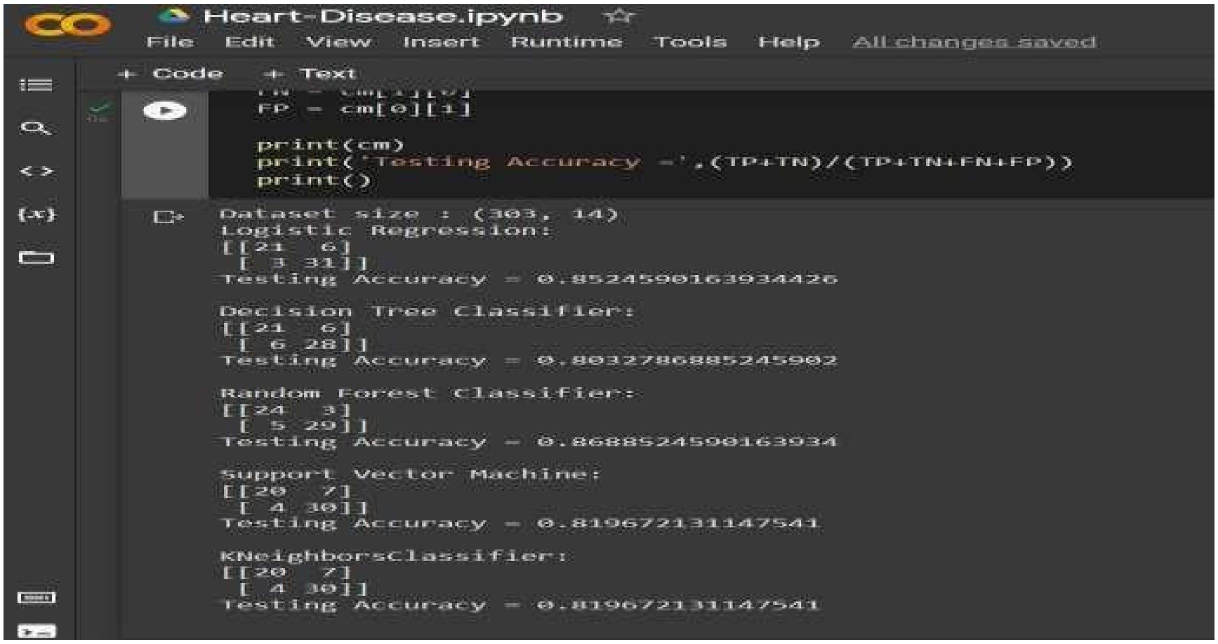
DIABETES DISEASE PREDICTION RESULTS



```
[19]    print(cm)
        print('Testing Accuracy =',(TP+TN)/(TP+TN+FN+FP))
        print()

     Dataset size : (768, 9)
     Logistic Regression:
     [[94 13]
      [19 28]]
     Testing Accuracy = 0.7922077922077922

     Decision Tree Classifier:
     [[82 25]
      [21 26]]
     Testing Accuracy = 0.7012987012987013

     Random Forest Classifier:
     [[93 14]
      [21 26]]
     Testing Accuracy = 0.7727272727272727

     Support Vector Machine:
     [[95 12]
      [21 26]]
     Testing Accuracy = 0.7857142857142857

     KNeighborsClassifier:
     [[86 21]
      [19 28]]
     Testing Accuracy = 0.7402597402597403
```

Fig 5.1: Diabetes disease prediction result.

HEART DISEASE PREDICTION RESULT

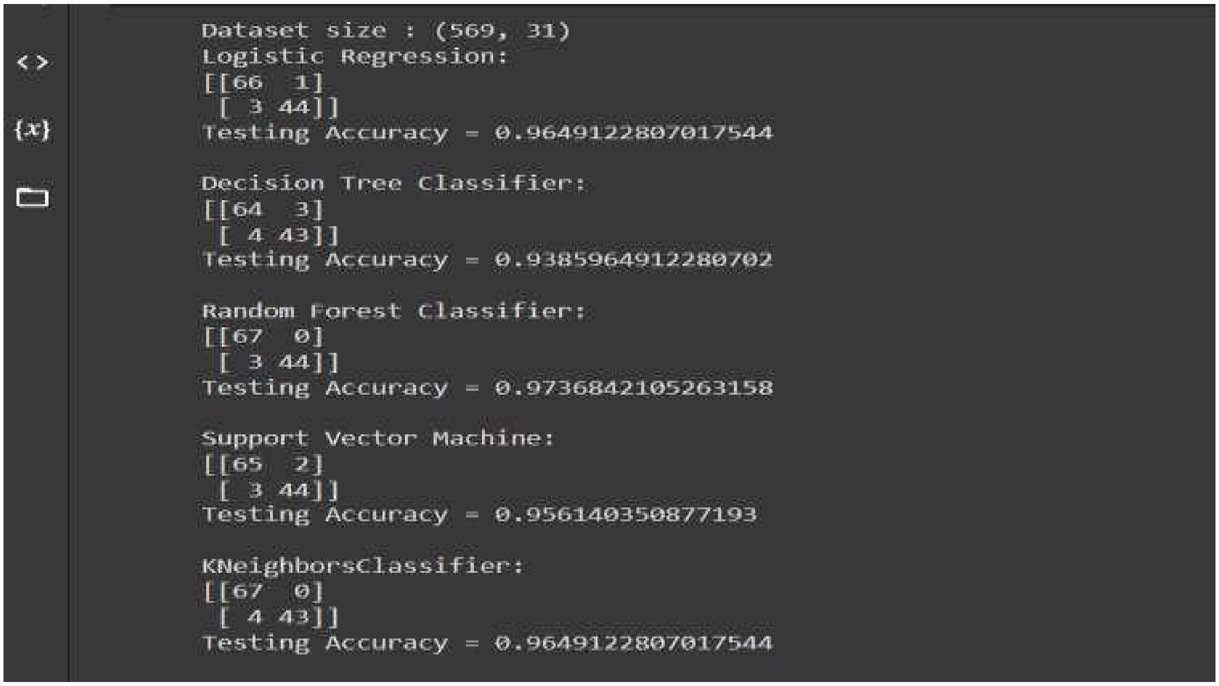


```
    FN = cm[1][0]
    FP = cm[0][1]

    print(cm)
    print('Testing Accuracy =',(TP+TN)/(TP+TN+FN+FP))
    print()
```

```
Dataset size : (303, 14)
Logistic Regression:
[[21  6]
 [ 3 31]]
Testing Accuracy = 0.8524590163934426

Decision Tree Classifier:
[[21  6]
 [ 6 28]]
Testing Accuracy = 0.8032786885245902

Random Forest Classifier:
[[24  3]
 [ 5 29]]
Testing Accuracy = 0.8688524590163934

Support Vector Machine:
[[20  7]
 [ 4 30]]
Testing Accuracy = 0.819672131147541

KNeighborsClassifier:
[[20  7]
 [ 4 30]]
Testing Accuracy = 0.819672131147541
```

Fig 5.2: Heart disease prediction result.

BREAST CANCER PREDICTION RESULT

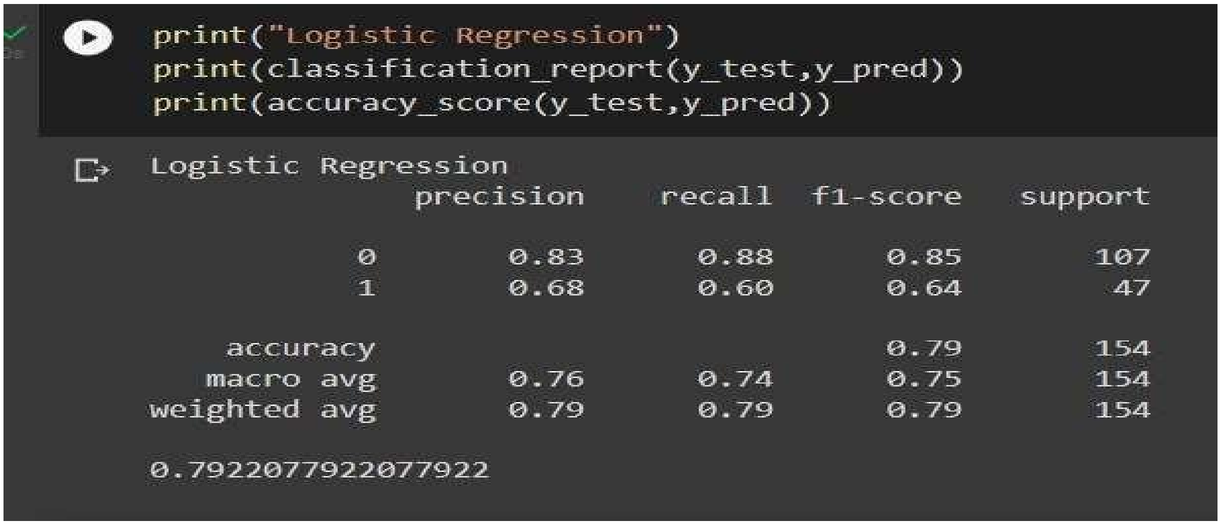


```
Dataset size : (569, 31)
Logistic Regression:
[[66  1]
 [ 3 44]]
Testing Accuracy = 0.9649122807017544

Decision Tree Classifier:
[[64  3]
 [ 4 43]]
Testing Accuracy = 0.9385964912280702

Random Forest Classifier:
[[67  0]
 [ 3 44]]
Testing Accuracy = 0.9736842105263158

Support Vector Machine:
[[65  2]
 [ 3 44]]
Testing Accuracy = 0.956140350877193

KNeighborsClassifier:
[[67  0]
 [ 4 43]]
Testing Accuracy = 0.9649122807017544
```

Fig 5.3: Breast cancer prediction result.

5.2 PERFORMANCE EVALUATION

DIABETES DISEASE PREDICTION PERFORMANCE ANALYSIS

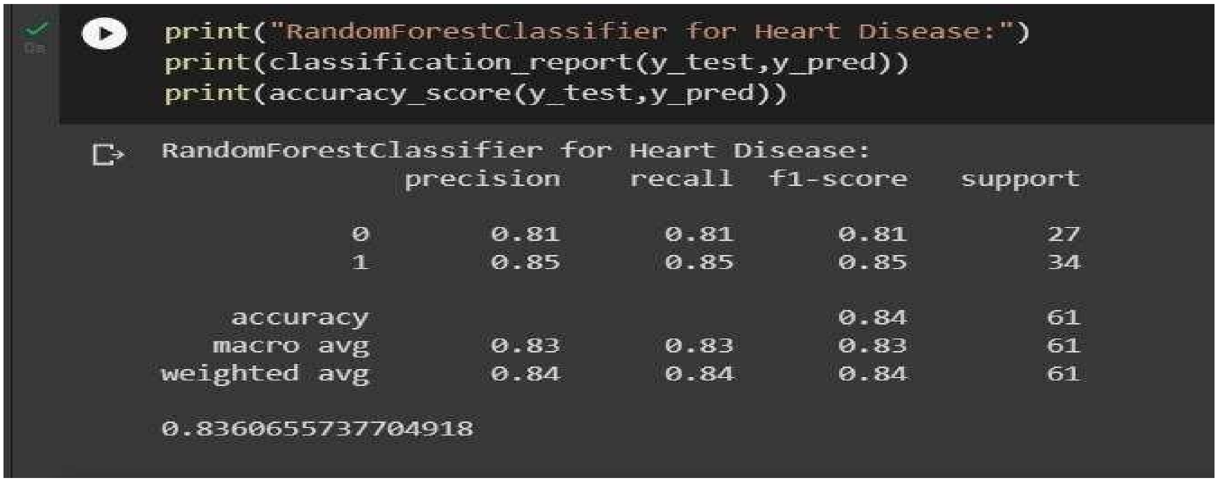


```
print("Logistic Regression")
print(classification_report(y_test,y_pred))
print(accuracy_score(y_test,y_pred))
```

```
Logistic Regression
              precision    recall  f1-score   support

           0       0.83      0.88      0.85       107
           1       0.68      0.60      0.64        47

    accuracy                           0.79       154
   macro avg       0.76      0.74      0.75       154
weighted avg       0.79      0.79      0.79       154

0.7922077922077922
```

Fig 5.4: Performance analysis of logistic regression for diabetes disease prediction.

HEART DISEASE PREDICTION PERFORMANCE ANALYSIS

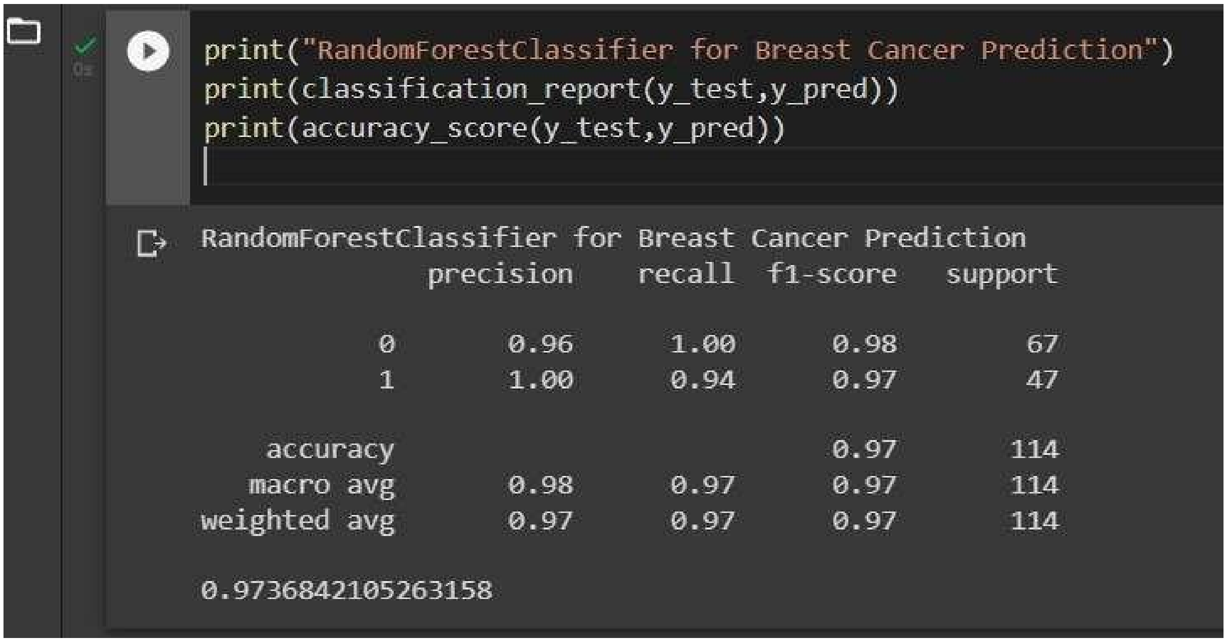


```
print("RandomForestClassifier for Heart Disease:")
print(classification_report(y_test,y_pred))
print(accuracy_score(y_test,y_pred))
```

```
RandomForestClassifier for Heart Disease:
              precision    recall  f1-score   support

           0       0.81      0.81      0.81        27
           1       0.85      0.85      0.85        34

    accuracy                           0.84        61
   macro avg       0.83      0.83      0.83        61
weighted avg       0.84      0.84      0.84        61

0.8360655737704918
```

Fig 5.5: Performance analysis of Random Forest classifier for heart disease prediction.

PERFORMANCE ANALYSIS OF BREASE CANCER PREDICTION



```
print("RandomForestClassifier for Breast Cancer Prediction")
print(classification_report(y_test,y_pred))
print(accuracy_score(y_test,y_pred))
```

```
RandomForestClassifier for Breast Cancer Prediction
              precision    recall  f1-score   support

           0       0.96      1.00      0.98        67
           1       1.00      0.94      0.97        47

    accuracy                           0.97       114
   macro avg       0.98      0.97      0.97       114
weighted avg       0.97      0.97      0.97       114

0.9736842105263158
```

***Fig 5.6: Performance analysis of random forest classifier for breast cancer prediction*.**

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 SUMMARY AND CONCLUSION

Earlier days in hospitals they lack in technological aspects for testing and issuing the reports which might take one day or may be more than that to issue the report for the lab related work that are being executed manually to predict the disease also they lack in efficiency and accuracy. But nowadays we have ample amount of data to show that these similar aspects or components can lead to this disease (exception may occur), so with the help of machine learning we have tried to implement similar system to predict the above stated disease which are most commonly found in person these days. In this application we have tried to implement a similar system which focuses on the three most deadly disease heart disease, breast cancer and diabetes diseases. We have implemented an effective way to reduce the dimensionality, reducing and eliminating the irrelevant data and increasing the accuracy. After the prediction of the disease a positive and negative report will be displayed according to which the patients can get best and nearby hospitals recommendations. It is to make easier way for patients to find the hospitals with good quality care of doctors. In total we are implementing our innovation ideas to give benefits to the people who are suffering from the health issues and they can make use of this application where they will find all good options at a time in one appeal. Opinions given by people on hospitals and doctors plays an important role and easily they can make decision. The goal was to use such associations to create a patient satisfaction based the recommendation system for hospitals.

6.2 FUTURE WORK

Future implementation is to recommend hospitals based on users review with the algorithm Collaborative Filtering: The motivation behind the CF calculation is to ascertain the benefits of a specific item for another item that is offered or for a chose client in view of the client's related knowledge and afterward the thoughts of different clients.

In view of authenticity and effortlessness, we expect to pay attention to what different clients share for all intents and purpose and love comparable preferences. Consolidating the inclinations of the two clients is considered by the orientation correspondence of the past. All CF techniques share the capacity to anticipate or give groundbreaking plans to individual clients who will appreciate utilizing past clients. Central issues depend on the possibility of connecting customers or items, and network is characterized as the demonstration of contracting between the first or the best. The two most significant CF modes are typically executed as client based objects, while the joined solicitation technique is separated into two gatherings: memory-based and model-based.

A memory-based approach is a heuristic in view of an assortment of things that clients have recently esteemed and remarked on. This expertise requires all scores, things, and clients to retain. The system depends on the utilization of an evaluation group to track down a model and make speculations. This innovation takes into account a normalized web-based evaluation technique. The CF strategy utilizes the thoughts of different networks to draw in clients. By and large, thoughts for the individuals who use them are utilized to gather the flavor of different clients. Hence, the CF expects that purchasers who have concurred in the past will probably settle on what's to come.

The CF framework requires a lot of information handling, including broadband, for example, web-based business and web facilitating.

Throughout the course of recent years, CF has advanced and has at long last become perhaps the most well-known method for significantly impacting the manner in which you approach directing. Today, PCs, as well as the Internet, assist us with contemplating the thoughts of an extraordinary spot with numerous individuals. People can profit from local gatherings, permitting them to acquire information from different clients and gaining from an assortment of items. Also, data can assist clients with making their own thoughts or check significant items out. Specifically, CF methods are utilized to assist clients with observing new items they might like, get guidance on explicit items, and associate with different clients who have comparative issues.

REFERENCE

[1] M. Denil, D. Matheson, and N. De Freitas, “Narrowing the Gap: Random Forests In Therein, M., Matheson, D., & De Freitas, N. (2014). Narrowing the Gap:

[2] W. Bergerud, “Introduction to logistic regression models with worked forestry examples: biometrics information handbook no. 7,” no. 7, p. 147, 1996.

[3] Watson, F. Marir "Using retrospect, they concluded that non-Spanish whites on average tend to go to hospitals that offer a better patient experience for all patients compared to hospitals commonly used by African American, Hispanic, Asian / Pacific Islander, or multiracial
patients" 1994.

[4] Binal A. Thakkar, Mosin I. Hasan, Mansi A. Desai, "Healthcare decision support system for swine flu prediction using naïve bayes classifier",IEEE", 101-105,2010.

[5] Disease Prediction and hospital recommendation using machine learning algorithm, www.academia.edu

[6] Random forest algorithm, javapoint.com

[7] Logistic regression algorithm, javapoint.com

[8] Youtube.com, for application reference

[9] Disease prediction and doctor recommendation system, International Research Journal of Engineering and Technology (IRJET)

APPENDICES

A. SAMPLE CODE

1. DIABETES DISEASE PREDICTION SOURCE CODE

```
# Importing packages.

import numpy as np
import pandas as pd

import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import accuracy_score,confusion_matrix,classification_report

from sklearn.neighbors import KNeighborsClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score,confusion_matrix
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import  RandomForestClassifier
from sklearn.svm import SVC

import joblib

data=pd.read_csv("diabetes.csv")   # Reading Dataset
corr=data.corr(method='pearson')   # Checking Correlation

diabetes_positive_count = len(data.loc[data['Outcome'] == 1])
diabetes_negative_count = len(data.loc[data['Outcome'] == 0])
print("total positve count:{0} and total negative
count:{1}".format(diabetes_positive_count,diabetes_negative_count))

cmap=sns.diverging_palette(220,10,as_cmap=True)
sns.heatmap(corr,cmap=cmap,vmax=.3,square=True,linewidths=6,cbar_kws={"shrink":.5})
colormap=plt.cm.viridis

plt.figure(figsize=(12,12))
plt.title('Pearson Correlation of Features', y=1.05, size=15)
sns.heatmap(data.corr(),linewidths=0.1,vmax=1.0, square=True, cmap=colormap,
linecolor='white',annot=True)

x=data.iloc[:,0:-1]
y=data.iloc[:,-1]

x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=0)

sc = StandardScaler()
x_train=sc.fit_transform(x_train)
x_test=sc.fit_transform(x_test)

log = LogisticRegression()
```

```
log.fit(x_train, y_train)

y_pred=log.predict(x_test)

print(classification_report(y_test,y_pred))
print(accuracy_score(y_test,y_pred))

# Saving Diabetes predct Train Model
filename = 'Diabetes-pred_model.sav'
joblib.dump(log, filename)

# load the model from disk
loaded_model = joblib.load(filename)

# Use the loaded model to make predictions
loaded_model.predict(x_test)
```

2. HEART DISEASE PREDICTION SOURCE CODE

```
import numpy as np
import pandas as pd

import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import accuracy_score,confusion_matrix,classification_report

from sklearn.neighbors import KNeighborsClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score,confusion_matrix
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import  RandomForestClassifier
from sklearn.svm import SVC
```

```
import joblib

hd=pd.read_csv("heart disease dataset.csv")

print(hd.columns)
print(hd.info())

corr=hd.corr("pearson")
print(corr)

x=hd.iloc[:,0:-1]
y=hd.iloc[:,-1]

x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=0)

sc = StandardScaler()
x_train=sc.fit_transform(x_train)
x_test=sc.fit_transform(x_test)

rfc=RandomForestClassifier(n_estimators=10,criterion='entropy',random_state=0)
rfc.fit(x_train,y_train)
y_pred = rfc.predict(x_test)
score2 = rfc.score(x_test,y_test)
print(score2)

#Saving Heart Disease Pred model
filename = 'Heart_Disease-pred_model.sav'
joblib.dump(rfc, filename)

# load the model from disk
loaded_model = joblib.load(filename)

# Use the loaded model to make predictions
loaded_model.predict(x_test)
```

3. BREAST CANCER PREDICTION SOURCE CODE

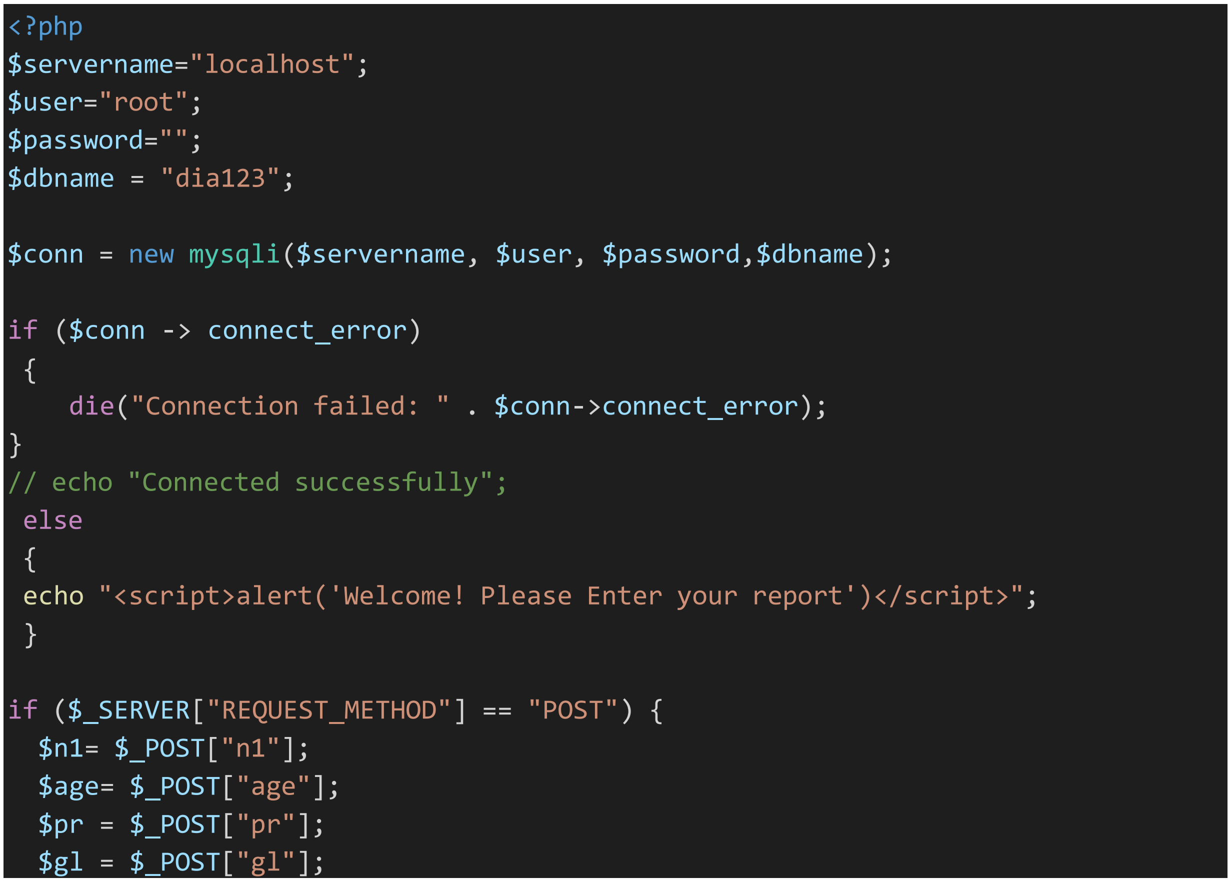
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline

from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import accuracy_score,confusion_matrix,classification_report

from sklearn.neighbors import KNeighborsClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score,confusion_matrix
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import  RandomForestClassifier
from sklearn.svm import SVC

import joblib
```

```
data=pd.read_csv("data.csv")
print(data)

s=LabelEncoder()
data.iloc[:,0]=s.fit_transform(data.iloc[:,0].values)        # 1=M,0=B
data.iloc[:,0]

print(data.corr())

plt.figure(figsize=(10,10))
sns.heatmap(data.iloc[:,:12].corr(),annot=True,fmt='.0%')

x=data.iloc[:,1:-1].values
y=data.iloc[:,0].values

x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=0)

sc = StandardScaler()
x_train=sc.fit_transform(x_train)
x_test=sc.fit_transform(x_test)

#RandomforestClassifier
rfc=RandomForestClassifier(n_estimators=10,criterion='entropy',random_state=0)
rfc.fit(x_train,y_train)

y_pred=rfc.predict(x_test)

print("RandomForestClassifier for Breast Cancer Prediction")
print(classification_report(y_test,y_pred))
print(accuracy_score(y_test,y_pred))

# Saving Breast Cancer predct Train Model
filename = 'Breast_Cancer-pred_model.sav'
joblib.dump(rfc, filename)

# load the model from disk
loaded_model = joblib.load(filename)
```

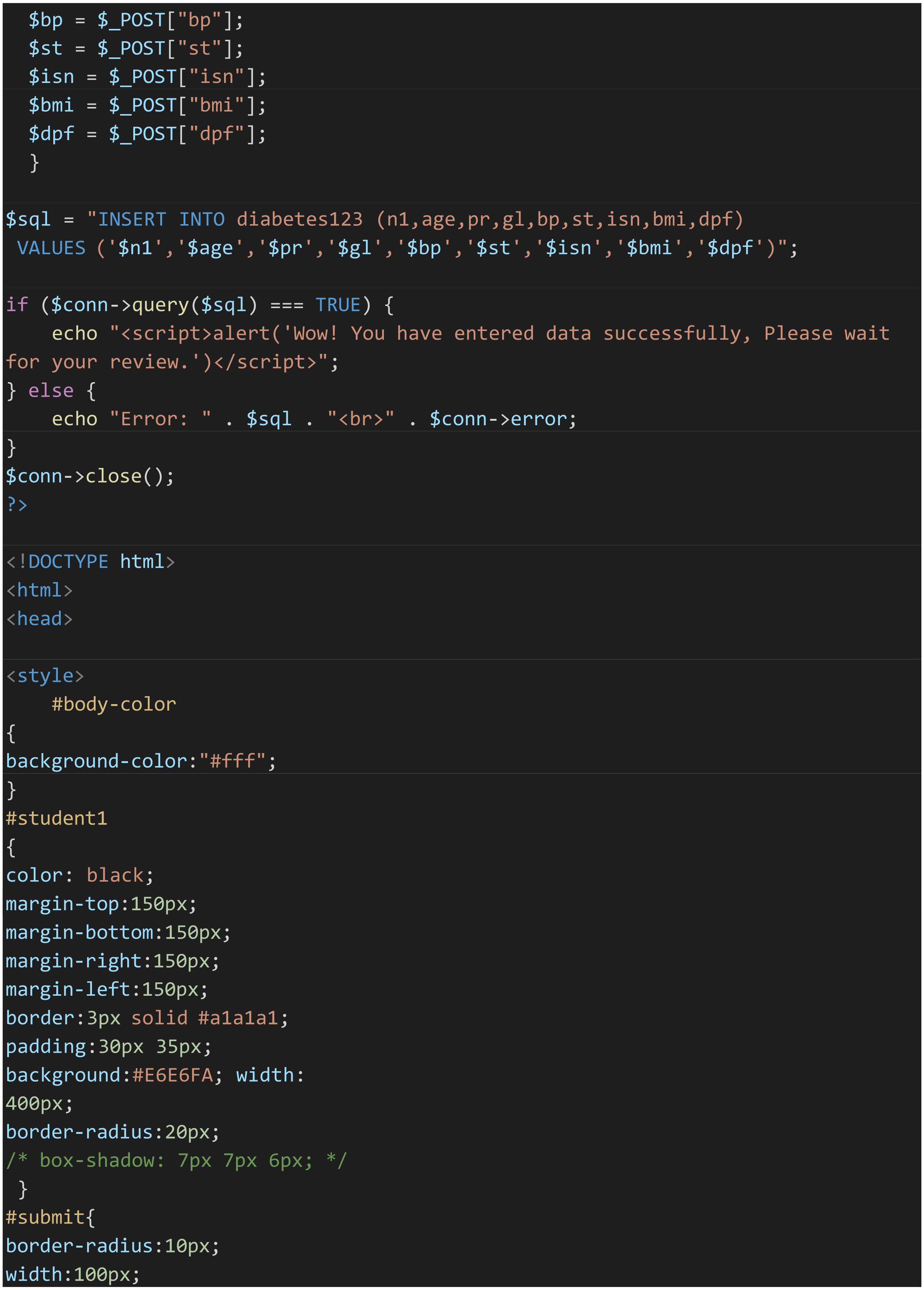
Use the loaded model to make predictions

loaded_model.predict(x_test)

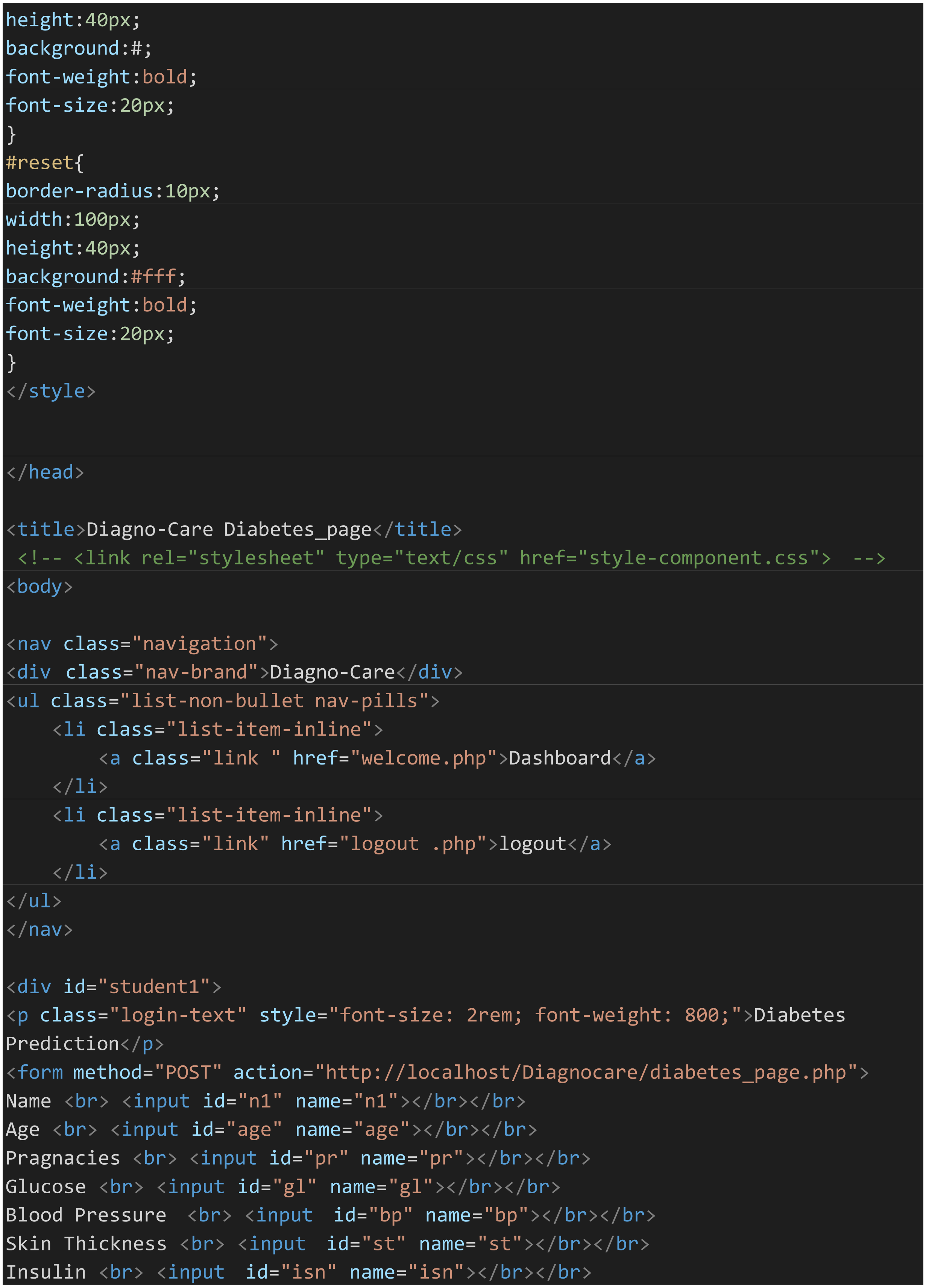
4. APPLICATION SOURCE CODE SAMPLE (DABETES DISEASE DATA ENTRY INTERFACE)



```php
<?php
$servername="localhost";
$user="root";
$password="";
$dbname = "dia123";

$conn = new mysqli($servername, $user, $password,$dbname);

if ($conn -> connect_error)
 {
    die("Connection failed: " . $conn->connect_error);
}
// echo "Connected successfully";
 else
 {
 echo "<script>alert('Welcome! Please Enter your report')</script>";
 }

if ($_SERVER["REQUEST_METHOD"] == "POST") {
  $n1= $_POST["n1"];
  $age= $_POST["age"];
  $pr = $_POST["pr"];
  $gl = $_POST["gl"];
```



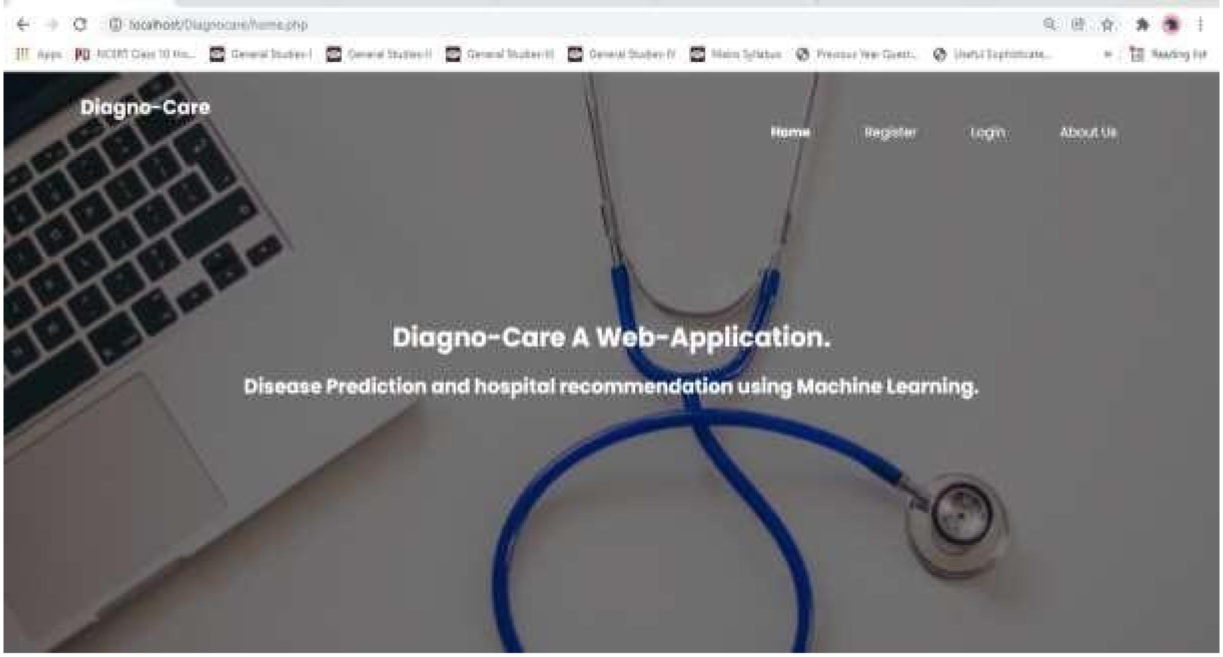
```
  $bp = $_POST["bp"];
  $st = $_POST["st"];
  $isn = $_POST["isn"];
  $bmi = $_POST["bmi"];
  $dpf = $_POST["dpf"];
  }

$sql = "INSERT INTO diabetes123 (n1,age,pr,gl,bp,st,isn,bmi,dpf)
 VALUES ('$n1','$age','$pr','$gl','$bp','$st','$isn','$bmi','$dpf')";

if ($conn->query($sql) === TRUE) {
    echo "<script>alert('Wow! You have entered data successfully, Please wait
for your review.')</script>";
} else {
    echo "Error: " . $sql . "<br>" . $conn->error;
}
$conn->close();
?>

<!DOCTYPE html>
<html>
<head>

<style>
    #body-color
{
background-color:"#fff";
}
#student1
{
color: black;
margin-top:150px;
margin-bottom:150px;
margin-right:150px;
margin-left:150px;
border:3px solid #a1a1a1;
padding:30px 35px;
background:#E6E6FA; width:
400px;
border-radius:20px;
/* box-shadow: 7px 7px 6px; */
 }
#submit{
border-radius:10px;
width:100px;
```



```
height:40px;
background:#;
font-weight:bold;
font-size:20px;
}
#reset{
border-radius:10px;
width:100px;
height:40px;
background:#fff;
font-weight:bold;
font-size:20px;
}
</style>


</head>

<title>Diagno-Care Diabetes_page</title>
 <!-- <link rel="stylesheet" type="text/css" href="style-component.css">  -->
<body>

<nav class="navigation">
<div  class="nav-brand">Diagno-Care</div>
<ul class="list-non-bullet nav-pills">
    <li class="list-item-inline">
        <a class="link " href="welcome.php">Dashboard</a>
    </li>
    <li class="list-item-inline">
        <a class="link" href="logout .php">logout</a>
    </li>
</ul>
</nav>

<div id="student1">
<p class="login-text" style="font-size: 2rem; font-weight: 800;">Diabetes
Prediction</p>
<form method="POST" action="http://localhost/Diagnocare/diabetes_page.php">
Name <br> <input id="n1" name="n1"></br></br>
Age <br> <input id="age" name="age"></br></br>
Pragnacies <br> <input id="pr" name="pr"></br></br>
Glucose <br> <input id="gl" name="gl"></br></br>
Blood Pressure  <br> <input  id="bp" name="bp"></br></br>
Skin Thickness <br> <input  id="st" name="st"></br></br>
Insulin <br> <input  id="isn" name="isn"></br></br>
```

```
BMI <br> <input  id="bmi" name="bmi"></br></br>
Diabetes Pedigree Function <br> <input  id="dpf" name="dpf"></br></br>


</br></br>
<div class="input-group">
<input type="submit" id="submit" value="Submit">
<input type="reset" id="reset" value="Reset">
</div>
</form>
</div>
</body>
</html>
```

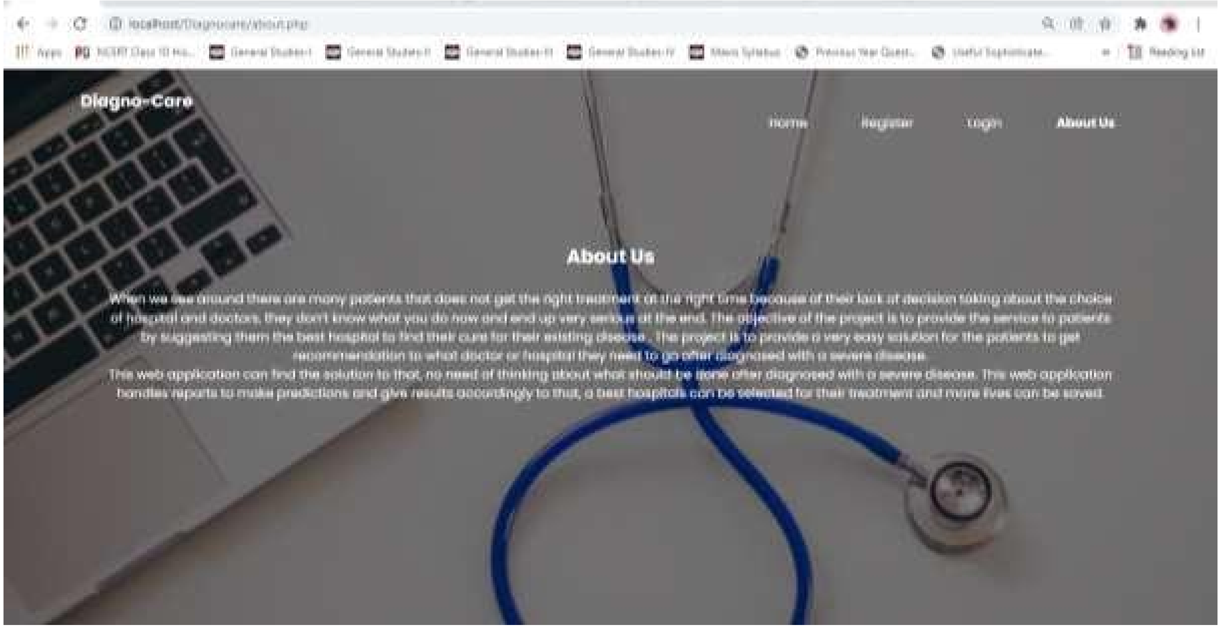
B. SCREENSHOTS



Diagno-Care A Web-Application.

Disease Prediction and hospital recommendation using Machine Learning.

Screenshot 1: Application Homepage (There are options for the user to register, login and to know about the application)

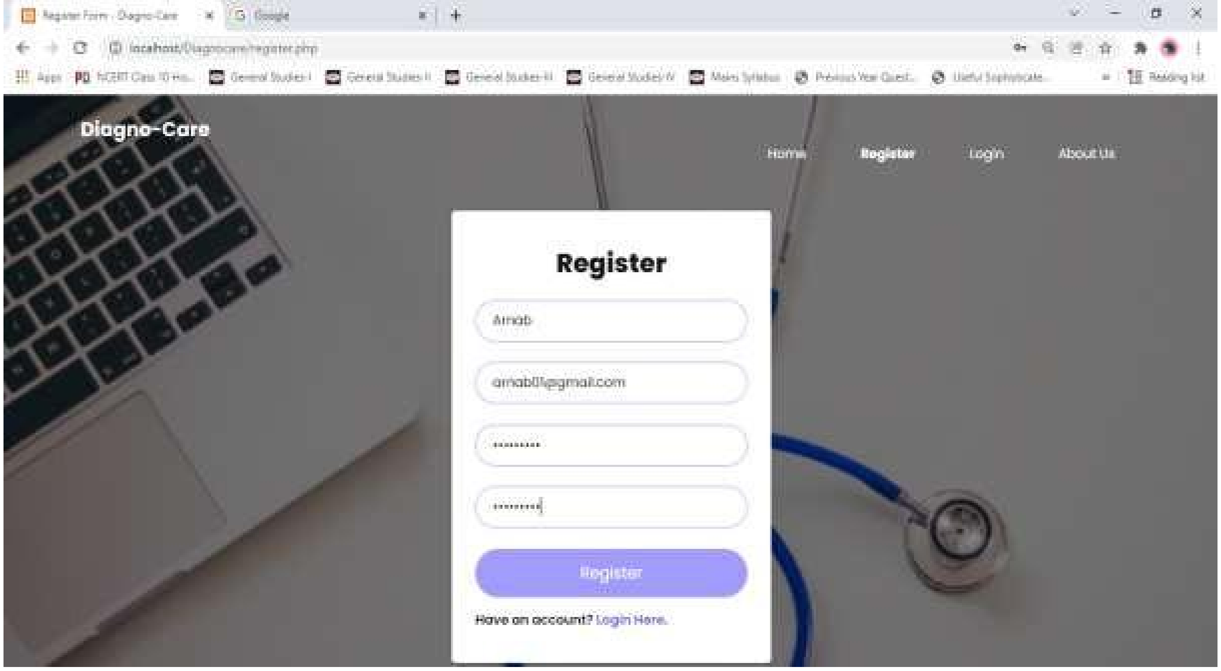


About Us

When we see around there are many patients that does not get the right treatment at the right time because of their lack of decision taking about the choice of hospital and doctors, they don't know what you do now and end up very serious at the end. The objective of the project is to provide the service to patients by suggesting them the best hospital to find their cure for their existing disease . The project is to provide a very easy solution for the patients to get recommendation to what doctor or hospital they need to go after diagnosed with a severe disease.

This web application can find the solution to that, no need of thinking about what should be done after diagnosed with a severe disease. This web application handles reports to make predictions and give results accordingly to that, a best hospitals can be selected for their treatment and more lives can be saved.

Screenshot 2: Application About Us page (To know about the application what is the purpose and what it does).



Register

Arnab

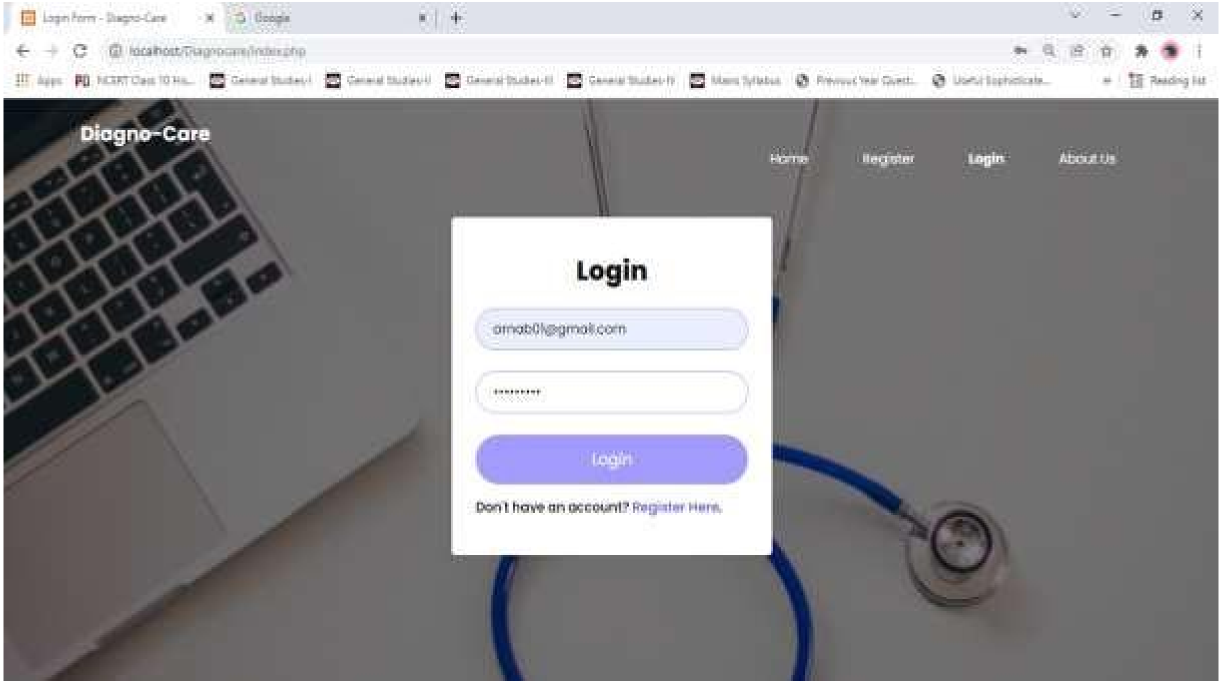
arnab01@gmail.com

••••••••••

••••••••••

Have an account? Login Here.

Screenshot 3: Application Register page (new users can register here into the application with id and password to safeguard their reports and reviews)

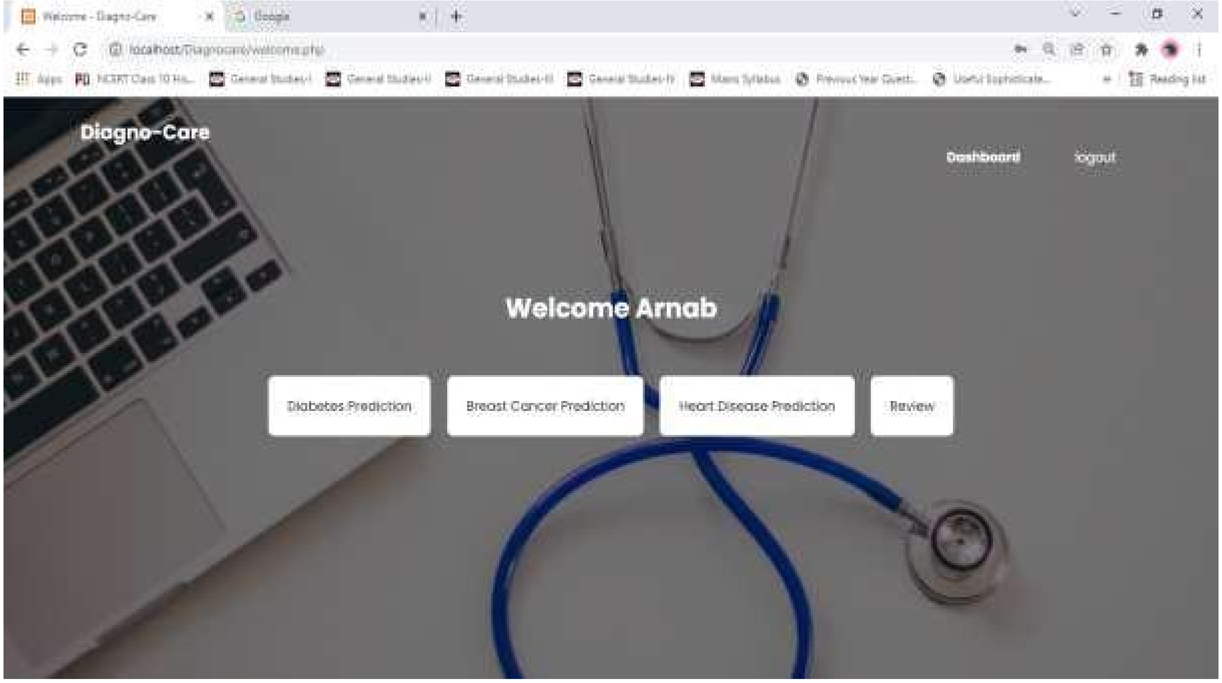


arnab0l@gmail.com

•••••••••

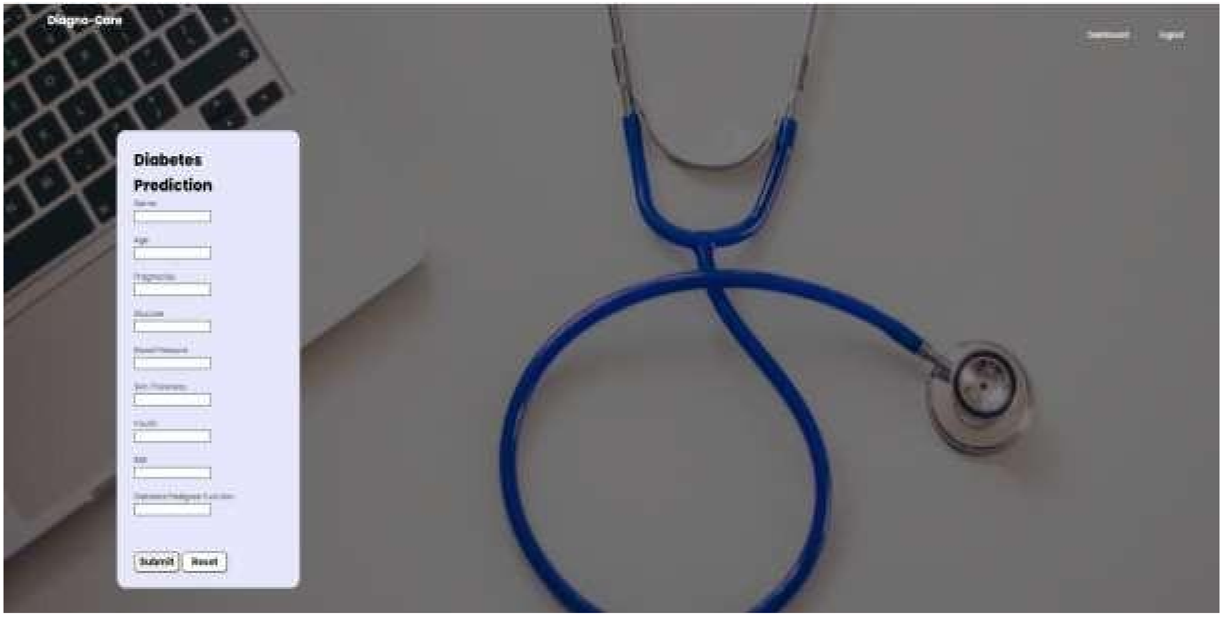
Don't have an account? Register Here.

Screenshot 4: Application Login page (Already registered users can use user id and password to login to the application and use the application for their benefits which is very user friendly to use.)



Screenshot 5: Application Dashboard page (where users are given different options for their disease prediction and to get recommendation and give

reviews)



Diagno-Care

Diabetes Prediction

Name

Age

Pregnancies

Glucose

Blood Pressure

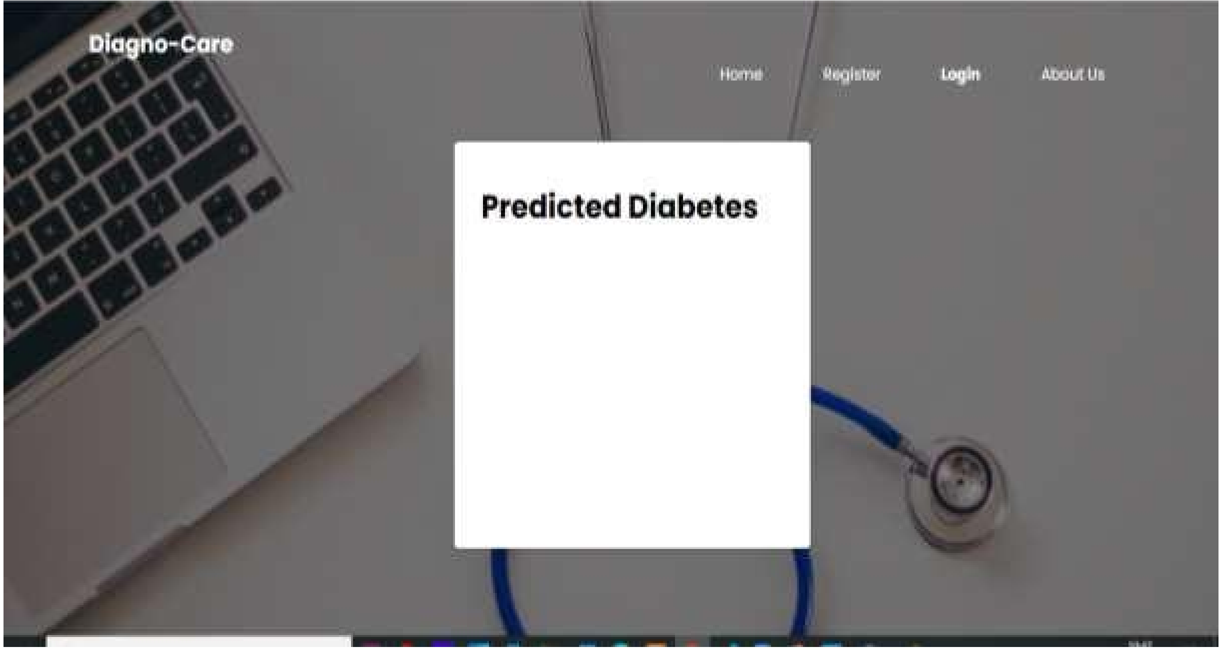
Skin Thickness

Insulin

BMI

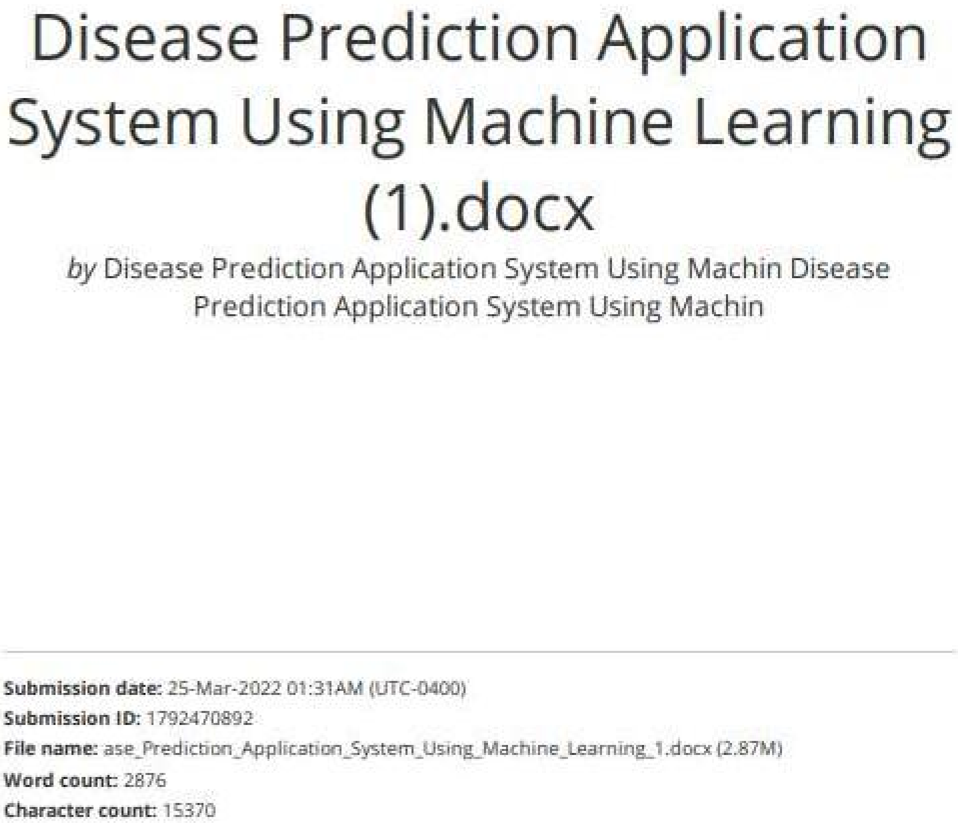
Diabetes Pedigree Function

Screenshot 6: Report entry page (where users can give enter according to their diagonised report)



Predicted Diabetes

Screenshot 7: Result page (where User get their report after the prediction whether he is suffering from the disease or no

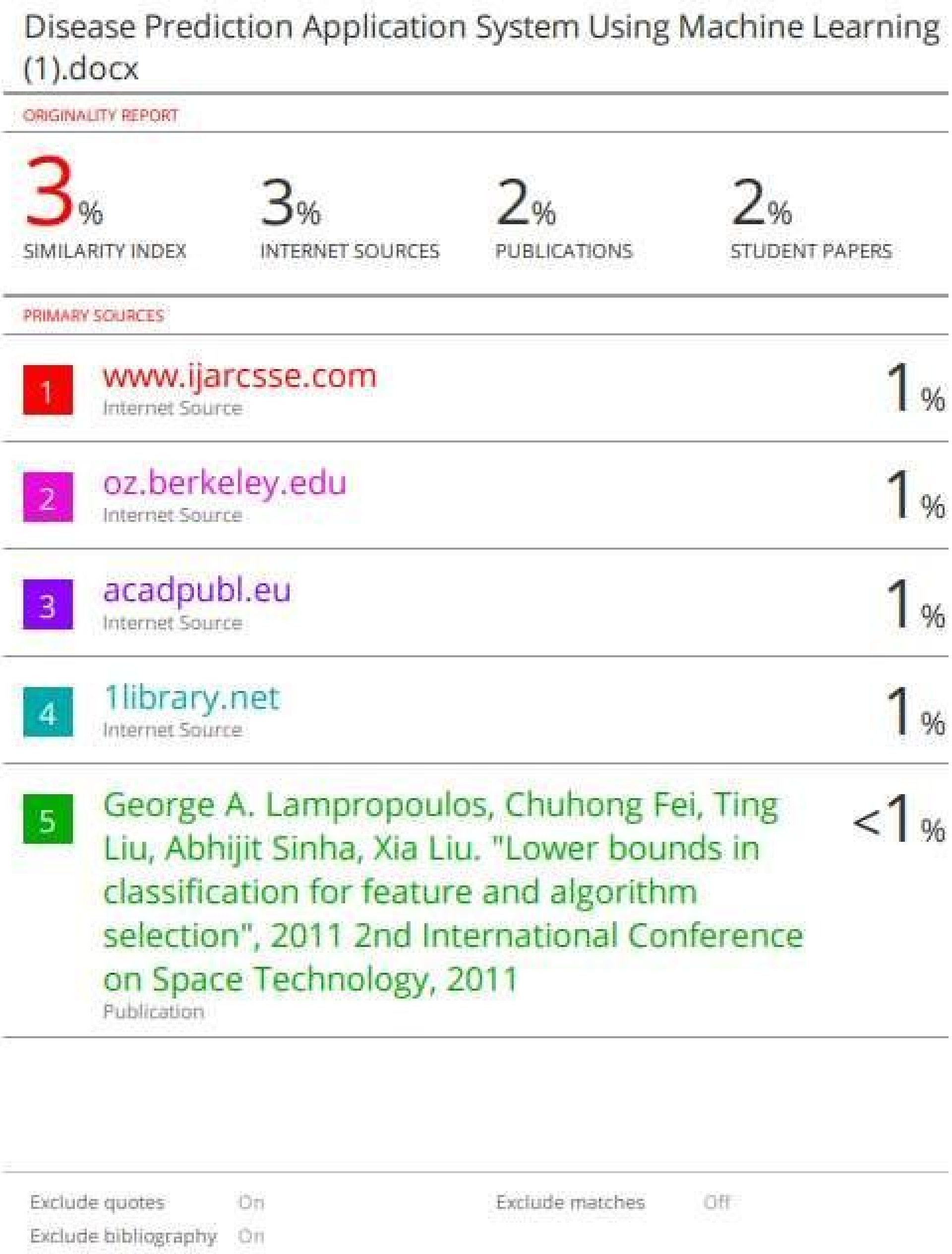


Disease Prediction Application System Using Machine Learning (1).docx

by Disease Prediction Application System Using Machin Disease Prediction Application System Using Machin

Submission date: 25-Mar-2022 01:31AM (UTC-0400)
Submission ID: 1792470892
File name: ase_Prediction_Application_System_Using_Machine_Learning_1.docx (2.87M)
Word count: 2876
Character count: 15370

Fig 7.1: Plagiarism Report 1



Disease Prediction Application System Using Machine Learning (1).docx

ORIGINALITY REPORT

3%	3%	2%	2%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.ijarcsse.com Internet Source	1%
2	oz.berkeley.edu Internet Source	1%
3	acadpubl.eu Internet Source	1%
4	1library.net Internet Source	1%
5	George A. Lampropoulos, Chuhong Fei, Ting Liu, Abhijit Sinha, Xia Liu. "Lower bounds in classification for feature and algorithm selection", 2011 2nd International Conference on Space Technology, 2011 Publication	<1%

Exclude quotes On

Exclude bibliography On

Exclude matches Off

Fig 7.2: Plagiarism Report 2