Project Report

On

DIABETES PREDICTION SYSTEM

Submitted by

C. Narasimha	**R170574**
T. SaiKumar Reddy	**R170575**
G. Prasad	**R170649**

Under the guidance of

P.Ravi Kumar
M.E, (Ph.D),Assistant Professor

Department of Computer Science and Engineering

Rajiv Gandhi University of Knowledge and Technologies(RGUKT),R.K.Valley, Kadapa,

Andhra Pradesh.

Rajiv Gandhi University of Knowledge TechnologiesRK
Valley, Kadapa (Dist), Andhra Pradesh, 516330

CERTIFICATE

This is to certify that the project work titled "**DIABETES PREDICTION SYSTEM** "
is a bonafied project work submitted by C Narasimhulu,G.Prasad,T.Sai kumar in the department of COMPUTER SCIENCE AND ENGINEERING in partial fulfillment of requirements for the award of degree of Bachelor of Technology in Computer science and engineering for the year 2020-2021 carried out the work under the supervision

GUIDE
P RAVI KUMAR

HEAD OF THE DEPARTMENT
P HARINADHA

PANEL MEMBERS :

1.

2.

3.

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible and whose constant guidance and encouragement crown all the efforts success.

I am extremely grateful to our respected Director,Prof. K. SANDHYA RANI for fostering an excellent academic climate in our institution.

I also express my sincere gratitude to our respected Head of the Department Mr.HARINADHA for his encouragement, overall guidance in viewing this project a good asset and effort in bringing out this project.

I would like to convey thanks to our guide at college Mr.P.RAVI KUMAR for his guidance, encouragement, co-operation and kindness during the entire duration of the course and academics.

My sincere thanks to all the members who helped me directly and indirectly in the completion of project work. I express my profound gratitude to all our friends and family members for their encouragement.

INDEX

Abstract

Diabetes mellitus is the most common disease worldwide and keeps increasing everyday due to changing lifestyles,unhealthy food habits and over weight problems. There were Studies handled in predicting diabetes through physical and Chemical tests are available for diagnostic diabetes.

The purpose of this project is to develop a system to predict the probablity of having diabetes,if the probability is more than 50% we can take treatment before it becomes critical. Diabetes mellitus is caused because of high level of glucose in human body ,it may cause some major issues like heart related problems ,eye damage and blood pressure.

Diabetes can be controlled if we Predict it earlier ,in this project we predict the probability of having diabetes based on the values of glucose,Pregnancies,BloodPressure(BP),SkinThickness,Insulin,BMI, Diabetes pedigree function and age by Applying varous Machine Learning Algorithms,like Support Vector Machine(SVM),Logistic Regression etc.The Experimental result shows that Prediction of diabetes done at high accuracy.

Introduction

Diabetes Mellitus (DM) is the most widely prevailing and rapidly spreading disease in all the countries all over the world. Early Prediction of diabetes helps us to prevent it from crictical and save human life.we Predict the probability of diabetes by taking various factors related to the diabetes mellitus.

For this prediction we used PIMA Indian diabetes dataset obtained from kaggle,we apply Various Machine Learning and Deep Learning Algorithms to predict probability of having diabetes.Machine Leaning Provides various method through which human efforts can be reduced in Detecting diabetes manually.

Purpose

The main purpose of Diabetes Prediction system is to predict the Probability of having Diabetes at earlier stages Preventing it from becoming too critical. Diabetes is one of the major cause of death in the world. Various Machine Learning techniques provide Efficient result by building various classification and ensemble models from Collected data set. Such collected can be used to predict diabetes .Various Machine learning and Deep Learning Algorithms can capable to predict the probability of having diabetes.

If the probability is more than 50% we can take treatment before it becomes critical. Diabetes mellitus is caused because of high level of glucose in human body ,it may cause some major issues like heart related problems ,eye damage and blood pressure,so we can save the human life by predicting it Earlier stage.

EXISTING SYSTEM AND ISSUES

1.Existing Many research handled for diabetes detection .

2.Data mining approach like clustering,classification were studied in existing System .

3.Diabetes prediction Using algorithms such as K-means clustering K-nearest Neighbour was Proposed

4.Using these algorithms the accuracy of prediction is less. And Also High False Prediction values. There is no Interactive tool for users to predict diabetes.

Proposed System

1.The Proposed System study is classification of Indian PIMA data set for diabetes as binary classification Problem .

2.This Proposed to achieve through Machine Learning And Deep Learning Algorithms.

3.For Machine Learning Logistic Regression and Support Vector Machines is proposed. Both Algorithms will give high accuracy .

4.For Deep learning Neural Network is Proposed. The Proposed system improves the accuracy of the prediction through deep learning techniques.

Advantages:

1. Interactive web Application ,in which users can give input values and will get the probability of having diabetes

2.Accuracy is High using Logistic Regression and Support Vector Machine.

3.Accuracy can be improved Using Deep Learning Techniques

Disadvantages:

- It reduces employment as the human efforts are being automated by this system.

Requirement Specification

Hardware Configuration:

Ram	At least 4 GB
Hard disk	10 GB
Processor	Intel i3 or more 2.00FHZ

Software Requirement:

Front end	HTML,CSS
Server side Language	FLASK
Web Browser	Firefox , Google Chrome or any compatible browser
Operating System	Ubuntu,Windows or any equivalent OS
Software	xampp

TOOLS AND TECHNOLOGIES

Numpy

NumPy is a library for the Python Programming *language,* adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays. In 2005,TravisOliphant created NumPy by incorporating features of the competing Numarray into Numeric, with extensive modifications. NumPy is open-source softwareand has many contributors.

Pandas

pandas is a software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series.

Matplotlib

Matplotlib is an amazing visualization library in Python for 2D plots of arrays. Matplotlib is a multi-platform data visualization library built on NumPy arrays and designed to work with the broader SciPy stack.Matplotlib consists of several plots like line, bar, scatter, histogram etc.

Scikit Learn

Scikit-learn (Sklearn) is the most useful and robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction via a consistence interface in Python. This library, which is largely written in Python, is built upon NumPy, SciPy and Matplotlib.

Seaborn

Seaborn is an amazing visualization library for statistical graphics plotting in Python. It provides beautiful default styles and color palettes to make statistical plots more attractive. It is built on the top of matplotlib library and also closely integrated to the data structures from pandas.

Flask

Flask is a micro web framework written in Python. It is classified as a micro-framework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions .However, Flask supports extensions that can add application features as if they were implemented in Flask itself. Extensions exist for object-relational mappers, form validation, upload handling, various open authentication technologies and several common framework related tools

MODEL DETAILS

1.Logistic Regression

Logistic regression is named for the function used at the core of the method, the logistic function .The logistic function, also called the sigmoid function was developed by statisticians to describe properties of population growth in ecology, rising quickly and maxing out at the carrying capacity of the environment. It's an S -shaped curve that can take any real-valued number and map it into a value between 0 and 1, but never exactly at those limits.

sigmoid Function

If z represents the output of the linear layer of a model trained with logistic regression, then $sigmoid(z)$ will yield a value (a probability) between 0 and 1. In mathematical terms:

$$y' = \frac{1}{1+e^{-z}}$$

where:

- y' is the output of the logistic regression model for a particular example.
- $z = b + w_1x_1 + w_2x_2 + \ldots + w_Nx_N$
  - The w values are the model's learned weights, and b is the bias.
  - The x values are the feature values for a particular example.

Note that z is also referred to as the *log-odds* because the inverse of the sigmoid states that z can be defined as the log of the probability of the 1 label (e.g., "dog barks") divided by the probability of the 0 label (e.g., "dog doesn't bark"):

$$z = \log\left(\frac{y}{1-y}\right)$$

Loss function for Logistic Regression

The loss function for linear regression is squared loss. The loss function for logistic regression is **Log Loss**, which is defined as follows:

$$\text{Log Loss} = \sum_{(x,y)\in D} -y \log(y') - (1-y) \log(1-y')$$

where:

- $(x, y) \in D$ is the data set containing many labeled examples, which are (x, y) pairs.
- y is the label in a labeled example. Since this is logistic regression, every value of y must either be 0 or 1.
- y' is the predicted value (somewhere between 0 and 1), given the set of features in x.

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

In the SVM algorithm, we are looking to maximize the margin between the data points and the hyperplane. The loss function that helps maximize the margin is hinge loss.The cost is 0 if the predicted value and the actual value are of the same sign. If they are not, we then calculate the loss value. We also add a regularization parameter the cost function. The objective of the regularization parameter is to balance the margin maximization and loss. After adding the regularization parameter, the cost functions looks as below.

$$c(x, y, f(x)) = \begin{cases} 0, & \text{if } y * f(x) \geq 1 \\ 1 - y * f(x), & \text{else} \end{cases}$$

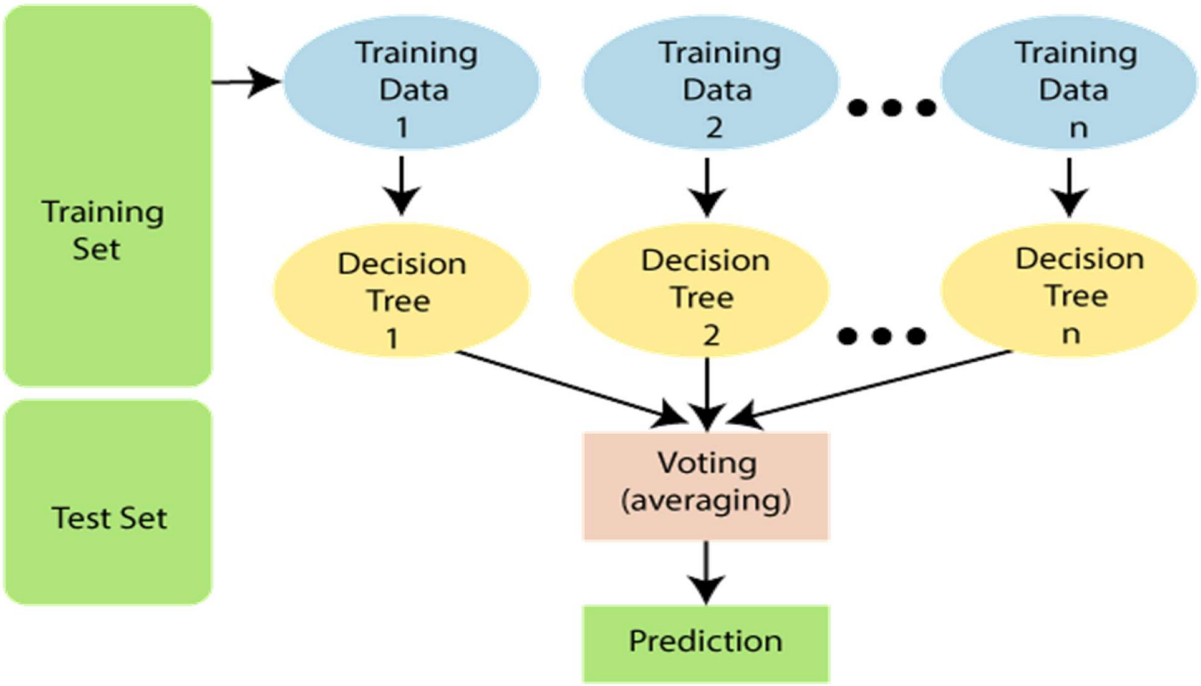
$$min_w \lambda \parallel w \parallel^2 + \sum_{i=1}^{n} (1 - y_i \langle x_i, w \rangle)_+$$

3.Random Forest Classifier

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

App.py

```
from flask import Flask,request, url_for, redirect, render_template
import pickle
import pandas as pd
app = Flask(__name__)
model = pickle.load(open("Diabetes.pkl", "rb"))
@app.route('/')
def hello_world():
    return render_template("index.html")
@app.route('/predict',methods=['POST','GET'])
def predict():
    text1 = request.form['1']
    text2 = request.form['2']
    text3 = request.form['3']
    text4 = request.form['4']
    text5 = request.form['5']
    text6 = request.form['6']
    text7 = request.form['7']
    text8 = request.form['8']

    row_df = pd.DataFrame([pd.Series([text1,text2,text3,text4,text5,text6,text7,text8])])
    print(row_df)
    prediction=model.predict_proba(row_df)
    output='{0:.{1}f}'.format(prediction[0][1], 2)
    output = str(float(output)*100)+'%'
    if output>str(0.5):
        return render_template('result.html',pred=f'You have chance of having diabetes.\nProbability of having
Diabetes is {output}')
    else:
        return render_template('result.html',pred=f'You are safe.\n Probability of having diabetes is {output}')

if __name__ == '__main__':
    app.run(debug=True)
```

Index.html

```
<!DOCTYPE html>
<html lang=”en”>
<head>
  <meta http-equiv=”Content-Type” content=”text/html; charset=UTF-8”/>
  <meta name=”viewport” content=”width=device-width, initial-scale=1, maximum-scale=1.0”/>
  <title>Diabetes prediction</title>

  <!—CSS →
  <link href=”https://fonts.googleapis.com/icon?family=Material+Icons” rel=”stylesheet”>
  <link href=”./static/css/materialize.css” type=”text/css” rel=”stylesheet” media=”screen,projection”/>
  <link href=”./static/css2/style.css” type=”text/css” rel=”stylesheet” media=”screen,projection”/>
</head>

<body>
  <nav class=”light-blue lighten-1” role=”navigation”>
    <div class=”nav-wrapper container”><a id=”logo-container” href=”#” class=”brand-logo”>Diabetes
Prediction</a>
      <ul class=”right hide-on-med-and-down”>
        <li><a href=”#”>Home</a></li>
      </ul>

  </nav>


  <div class=”section no-pad-bot” id=”index-banner”>
    <div class=”container”>
     <h5 style=”text-align: center; color: brown; font-weight: 600;”>{{pred}}</h5>
      <h1 class=”header center orange-text”>Diabetes Prediction</h1>
      <div class=”row center”>
        <h5 class=”header col s12 light”>Predict the probability of having Diabetes
        <br>
        </h5>
      </div>


      <div class=”row”>
        <form action=’/predict’ method=”post” class=”col s12”>
          <div class=”row”>
            <div class=”input-field col s4”>
              <label for=”first_name”><b>Pregnancies</b></label>
              <br>
```

```
            <input placeholder="No. of Pregnancies" name="1" id="first_name" type="text"
class="validate">
          </div>
          <div class="input-field col s4">
            <label for="last_name"><b>Glucose </b></label>
            <br>
            <input id="last_name" name="2" placeholder="Glucose level in sugar"  type="text"
class="validate">

          </div>

          <div class="input-field col s4">
            <label for="_name"><b>BloodPressure</b></label>
            <br>
            <input id="_name" name="3" placeholder="BloodPressure" type="text" class="validate">

          </div>
          <div class="input-field col s4">
            <label for="first_name"><b>SkinThickness</b></label>
            <br>
            <input placeholder="SkinThickness" name="4" id="first_name" type="text" class="validate">
          </div>
          <div class="input-field col s4">
            <label for="last_name"><b>Insulin</b></label>
            <br>
            <input id="last_name" name="5" placeholder="Insulin level"  type="text" class="validate">
          </div>
          <div class="input-field col s4">
           <label ><b>BMI</b></label>
           <br>
           <input name = '6' placeholder="Body Mass Index">

          </div>
          <div class="input-field col s4">
            <label for="last_name"><b>DiabetesPedigreeFunction</b></label>
            <br>
            <input id="last_name" name="7" placeholder="DiabetesPedigreeFunction"  type="text"
class="validate">
          </div>
          <div class="input-field col s4">
            <label for="_name"><b>Age</b></label>
            <br>
            <input id="_name" name="8" placeholder="Age" type="text" class="validate">
          </div>
```

```
        </div>

        <div class="row center">

          <button type="submit" class="btn-large waves-effect waves-light orange">Predict
Probability</button>

        </div>
      </form>
 </div>

    <br>


   <br><br>
 </div>
 </div>>

 <footer class="page-footer orange">
   <div class="footer-copyright">
     <div class="container" style="padding-bottom: 1rem; position: fixed; font-size: 1.2rem; margin-left:
5rem;">
       <p>&copy; Abhay</p>
       <p>Parasharabhay13@gmail.com</p>
     </div>
   </div>
 </footer>


 <!--  Scripts→
 <script src="https://code.jquery.com/jquery-2.1.1.min.js"></script>
 <script src=".js/materialize.js"></script>
 <script src="js/init.js"></script>

 </body>
</html>
```

result.html

```
<html>
<head>
   <title>diabates prediction</title>
   <Link rel="stylesheet"type="text/css" href="/static/css/styles.css">
</head>
<body>
   <header>
      <h1>DAIBATES PREDICTION</h1>
   </header>

   <div class="nav">
      <ul>
         <li><a href="/">HOME</a></li>
      </ul>
   </div>
   <div class="row" style="margin:15% 0% 0% 10%">
      <h3 >{{pred}}</h3>
   </div>
</body>
</html>
```

Styles.css

```
body
{
   font-family:  Arial, Helvetica, sans-serif,Times, serif;
}

header
{
   background-color: #59bfeb;
   padding:10px;
   margin:0;
   overflow:hidden;
   text-align: center;
   font-family: Verdana, Geneva, Tahoma, sans-serif;
}
```

```
header h1
{
   color:white;
   font-size: 1.8em;
}

.predict
{
   text-align: center;
   font-family:Haettenschweiler, 'Arial Narrow Bold', sans-serif;
   font-size: 1.5em;
   margin-top: 20px;
}

.nav ul
{

   list-style-type:none ;
   background-color: #b1bec4;
   margin:0;
   padding:0;
   overflow:hidden;
}

.nav li
{
   float:center;
   font-family:'Times New Roman', Times, serif;
   font-size: 0.9em;

}

.nav li a
{
   text-decoration: none;
   padding:16px 79px ;
   display:block;
   color:black;
   text-align: center;
}

.nav li a:hover
{
```

```
    background-color:#0b253f ;
    color:white;
}

.nav li a.active
{
    background-color: #659dbd;
    color:black;
}

form{
    background-color: rgb(255, 255, 255);
    margin-left: auto;
    margin-right: auto;
    width: 700px;
    text-align: center;
    padding-bottom: 24px;
    border: 1px solid #ccc;
}

#input-wrapper{
    padding-top: 80px;
    padding-bottom: 40px;
    padding-right: 80px;
    padding-left: 80px;
}

.input-box{
    font-size: 20px;
    padding-top: 20px;
    padding-bottom: 12px;
    margin-bottom: 36px;
    border: 0;
    border-bottom: 1px solid #ccc;
    width: 100%;
}

#submit-button{
    background-color: #59bfeb;
    padding-top: 12px;
    padding-bottom: 12px;
    padding-right: 24px;
    padding-left: 24px;
    color: white;
```

```
    font-size: 18px;
    border: none;
}

.footer{
    padding:0 10px;
    color:#0b253f;
    background-color:#59bfeb ;
    padding: 8px;
}
```

PIMA_INDIAN_Diabetes.ipynb

```
#importing Libraries
import numpy as np
np.random.seed(42)
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
%matplotlib inline


#models
from sklearn.ensemble import RandomForestClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import SVC

#Evaluation
from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.model_selection import RandomizedSearchCV,GridSearchCV
from sklearn.metrics import confusion_matrix, classification_report
from sklearn.metrics import precision_score, recall_score, f1_score
from sklearn.metrics import plot_roc_curve

#for warning
from warnings import filterwarnings
filterwarnings("ignore")
#Load the dataset
data = pd.read_csv("diabetes.csv")## Build an model (Logistic Regression)
from sklearn.linear_model import LogisticRegression
```

```
log_reg = LogisticRegression(random_state=0)
log_reg.fit(X_train,y_train);
## Evaluating the model
log_reg = log_reg.score(X_test,y_test)
## Build an model (Random forest classifier)
clf= RandomForestClassifier()
clf.fit(X_train,y_train);
## Evaluating the model
clf = clf.score(X_test,y_test)
## Build an model (Support Vector Machine)
svm = SVC()
svm.fit(X_train,y_train)
svm = svm.score(X_test,y_test)
model_compare = pd.DataFrame({"Logistic Regression":log_reg,
                      "Random Forest Classifier":clf,
                     "Support Vector Machine":svm,
                      },index=["accuracy"])
import pickle

# Save trained model to file
pickle.dump(gs_log_reg, open("Diabetes.pkl", "wb"))
loaded_model = pickle.load(open("Diabetes.pkl", "rb"))
loaded_model.predict(X_test)
loaded_model.score(X_test,y_test)
## Enter the new data
X_test.head(1)
Pregnancies = input()
Glucose = input()
BloodPressure = input()
SkinThickness = input()
Insulin = input()
BMI = input()
DiabetesPedigreeFunction = input()
Age = input()
row_df = pd.DataFrame([pd.Series([Pregnancies,Glucose,BloodPressure,SkinThickness,Insulin,BMI
,DiabetesPedigreeFunction ,Age])])
row_df
prob = loaded_model.predict_proba(row_df)[0][1]
print(f"The probability of you having Diabetes is {prob}")
```

IMPLEMENTATION DIAGRAM FOR DIABETIES PREDICTION

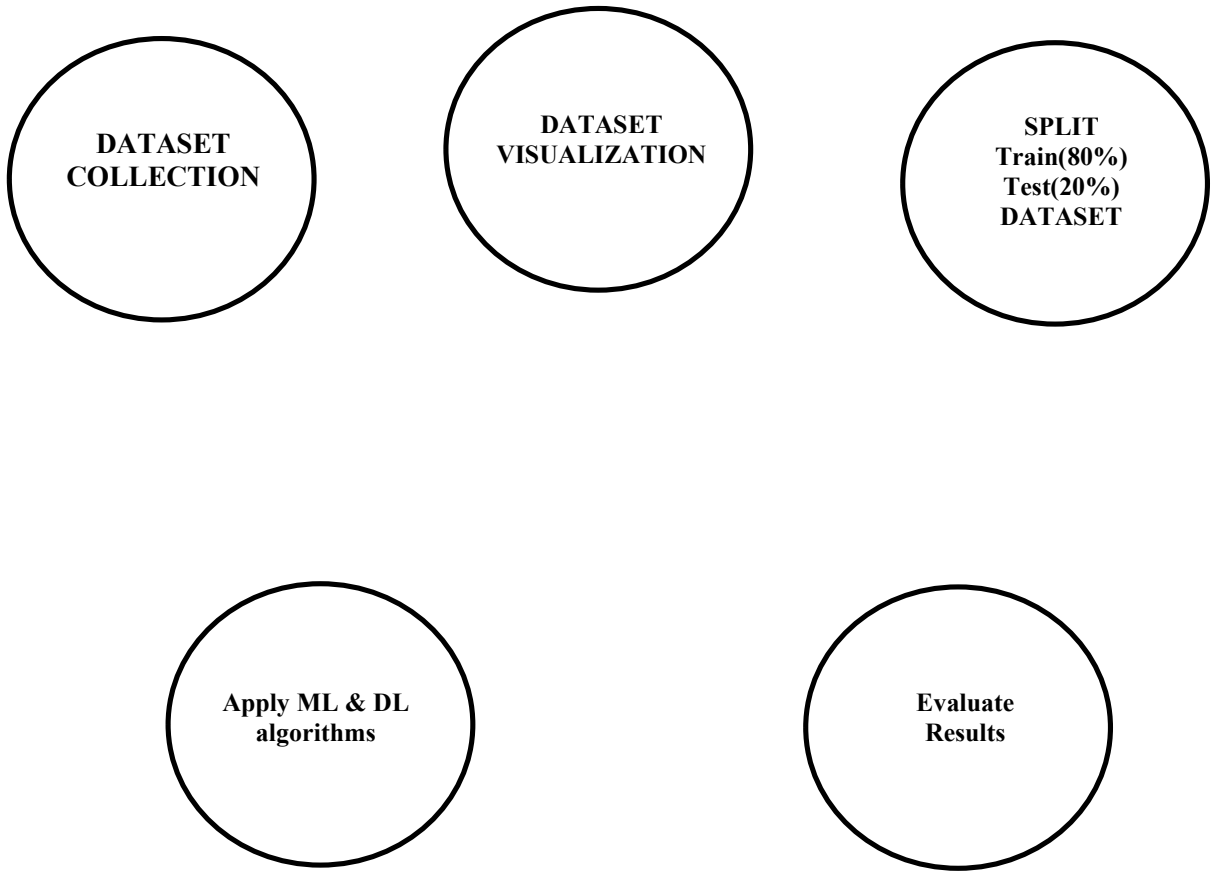


Fig.1

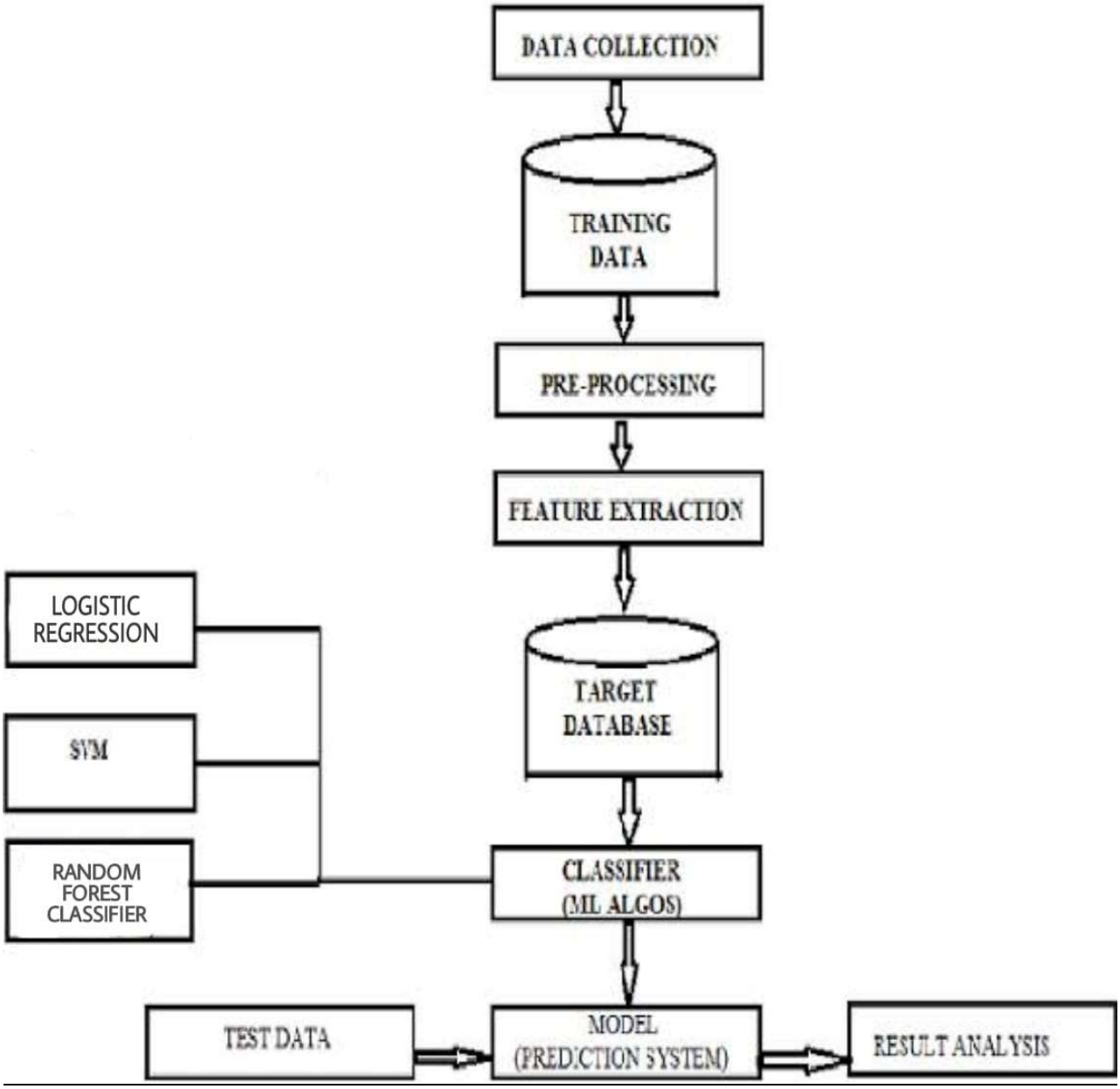

DATA COLLECTION
TRAINING DATA
PRE-PROCESSING
FEATURE EXTRACTION
LOGISTIC REGRESSION
SVM
RANDOM FOREST CLASSIFIER
TARGET DATABASE
CLASSIFIER (ML ALGOS)
TEST DATA
MODEL (PREDICTION SYSTEM)
RESULT ANALYSIS

Fig.2

Implementation and System Testing

After all phase have been perfectly done, the system will be implemented to the server and the system can be used.

System Testing

The goal of the system testing process was to determine all faults in our project .The program was subjected to a set of test inputs and many explanations were made and based on these explanations it will be decided whether the program behaves as expected or not. Our Project went through two levels of testing

1. Unit testing

2 .Integration testing

Unit Testing

Unit testing is commenced when a unit has been created and effectively reviewed .In order to test a single module we need to provide a complete environment i.e. besides the section we would require The procedures belonging to other units that the unit under test calls Non local data structures that module accesses .A procedure to call the functions of the unit under test with appropriate parameters

Integration Testing

In the Integration testing we test various combination of the project module by providing the input.

The primary objective is to test the module interfaces in order to confirm that no errors are occurring when one module invokes the other module.

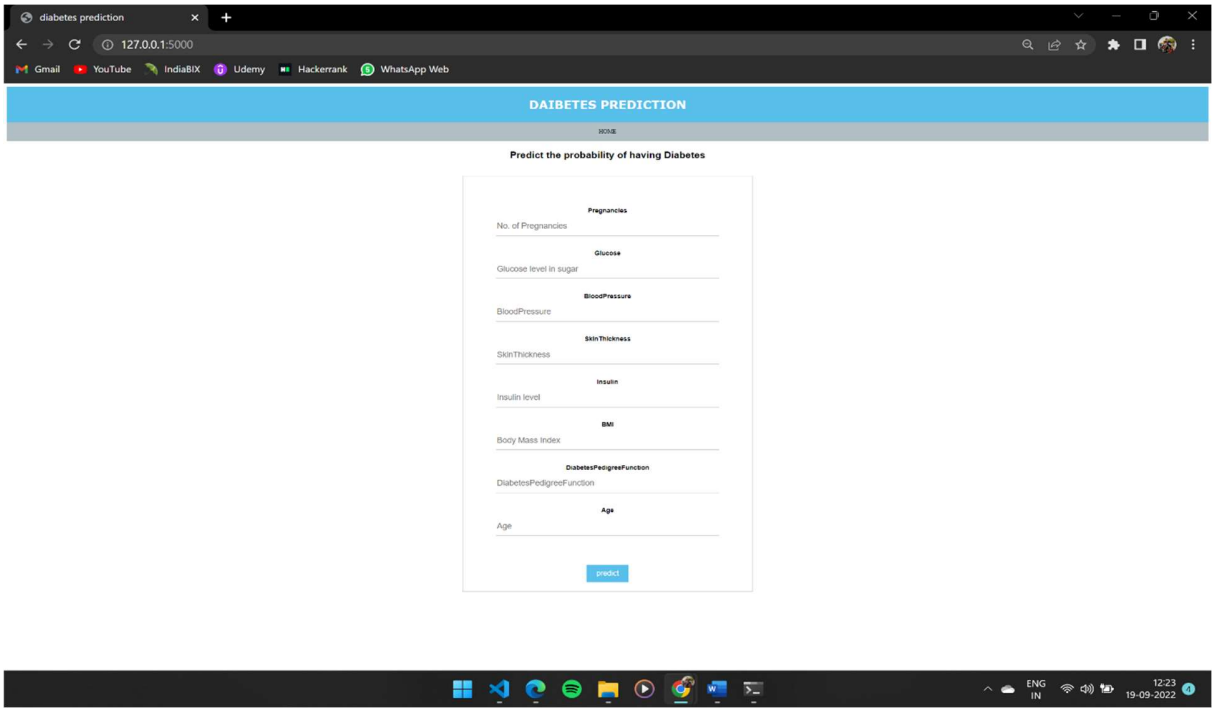
Project Output

Project URL: http://127.0.0.1:5000/

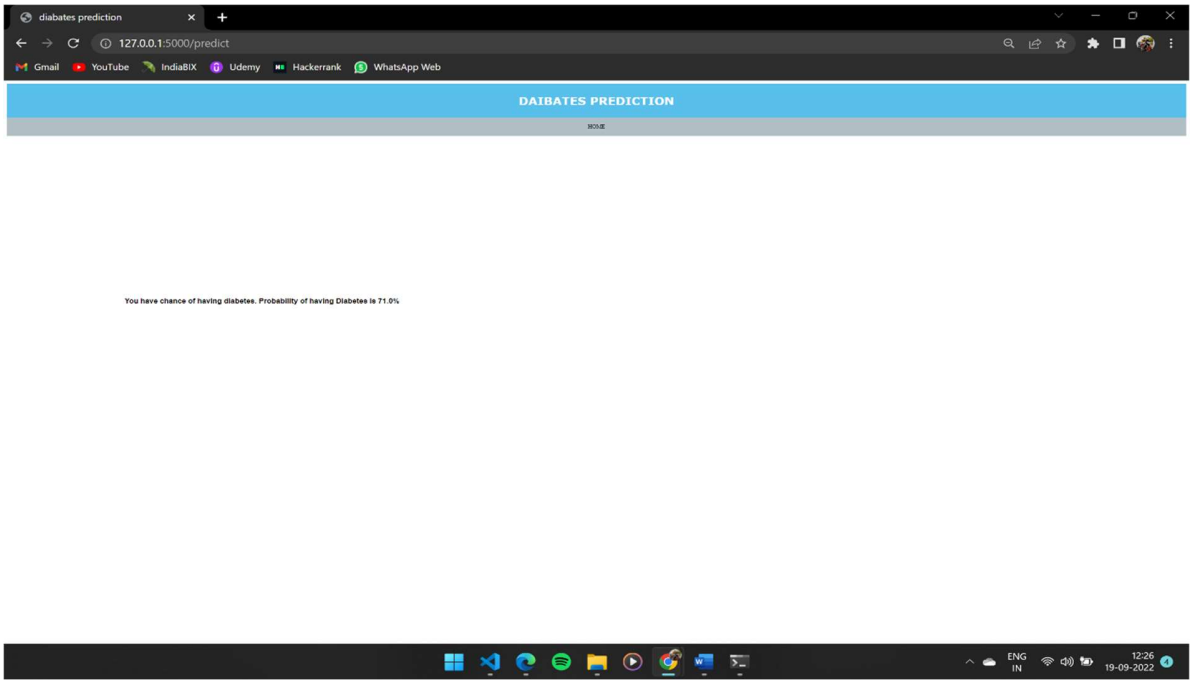
Dataset.csv

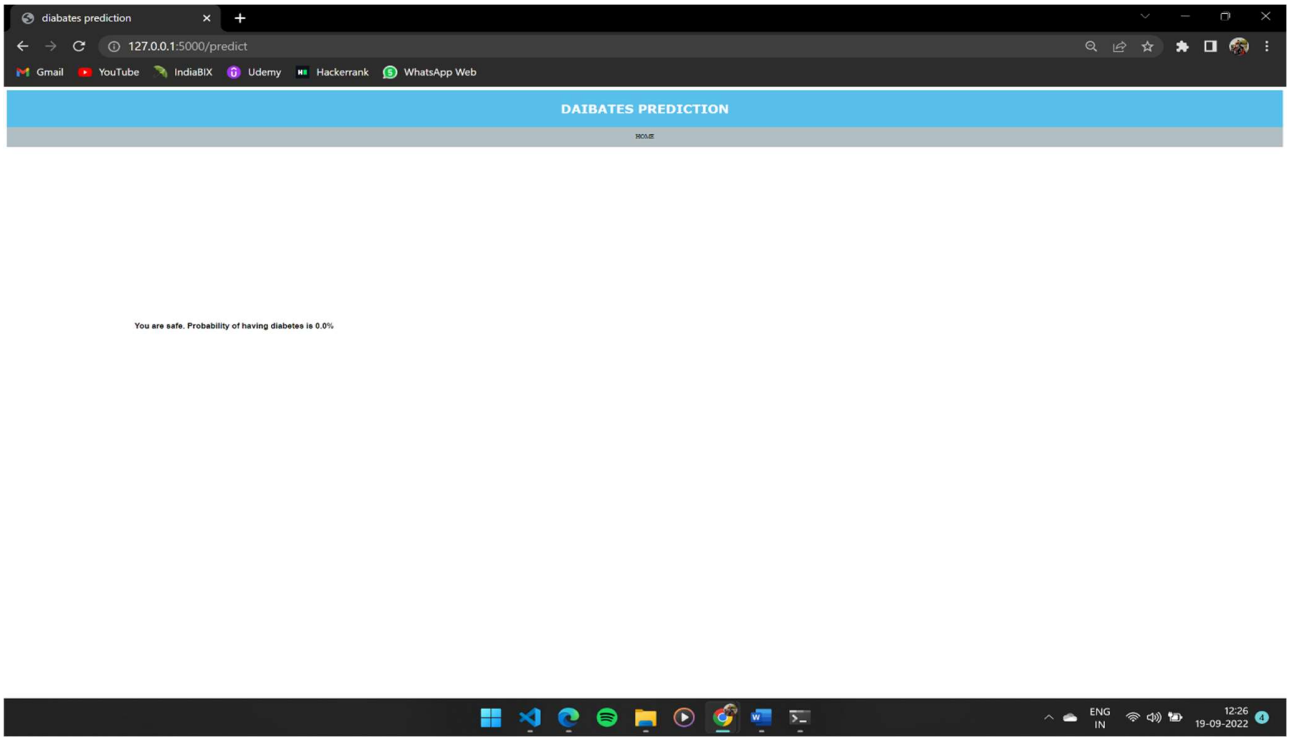
	A	B	C	D	E	F	G	H	I
1	Pregnancie	Glucose	BloodPress	SkinThickn	Insulin	BMI	DiabetesPe	Age	Outcome
2	6	148	72	35	0	33.6	0.627	50	1
3	1	85	66	29	0	26.6	0.351	31	0
4	8	183	64	0	0	23.3	0.672	32	1
5	1	89	66	23	94	28.1	0.167	21	0
6	0	137	40	35	168	43.1	2.288	33	1
7	5	116	74	0	0	25.6	0.201	30	0
8	3	78	50	32	88	31	0.248	26	1
9	10	115	0	0	0	35.3	0.134	29	0
10	2	197	70	45	543	30.5	0.158	53	1
11	8	125	96	0	0	0	0.232	54	1
12	4	110	92	0	0	37.6	0.191	30	0
13	10	168	74	0	0	38	0.537	34	1
14	10	139	80	0	0	27.1	1.441	57	0
15	1	189	60	23	846	30.1	0.398	59	1
16	5	166	72	19	175	25.8	0.587	51	1
17	7	100	0	0	0	30	0.484	32	1
18	0	118	84	47	230	45.8	0.551	31	1
19	7	107	74	0	0	29.6	0.254	31	1
20	1	103	30	38	83	43.3	0.183	33	0
21	1	115	70	30	96	34.6	0.529	32	1
22	3	126	88	41	235	39.3	0.704	27	0
23	8	99	84	0	0	35.4	0.388	50	0
24	7	196	90	0	0	39.8	0.451	41	1
25	9	119	80	35	0	29	0.263	29	1

Home page



Result page




diabates prediction
127.0.0.1:5000/predict
Gmail
YouTube
IndiaBIX
Udemy
Hackerrank
WhatsApp Web
DAIBATES PREDICTION
HOME
You are safe. Probability of having diabetes is 0.0%
ENG
IN
12:26
19-09-2022

Conclusion

DIABETES Prediction System is very much graceful and lively. Patients need to enter some of their details to check the diabetes percentage.By that information patients can get the information that will say whether thet get the diabetes or not by the percentage. Automation of the entire system improves the productivity.

- It gives appropriate access to the authorized users depending on their permissions.
- The System has adequate scope for modification in future if it is necessary.
- Updating of information becomes so easier.
- System security, data security and reliability are the striking features.
- It effectively overcomes the delay in communications

In future we implement diabetes prediction system using deep learning algorithms in cloud to get more accurate results.

References

For HTML

- https://www.w3schools.com/html/
- https://www.tutorialspoint.com/html/index.htm

For CSS

- https://www.w3schools.com/css/

For FLASK

- https://flask.palletsprojects.com/en/2.2.x/
- https://www.tutorialspoint.com/flask/index.htm

For MACHINE LEARNING ALGORITHM

- https://scikit-learn.org/stable/
- https://www.javatpoint.com/numpy-tutorial
- https://www.javatpoint.com/python-pandas

**********THANKYOU**********