**SAVITRIBAI PHULE PUNE UNIVERSITY**

**A PRELIMINARY PROJECT REPORT ON**

**“Diabetes Prediction System”**

**SUBMITTED TOWARDS THE PARTIAL FULFILMENT OF THE REQUIREMENTS OF**

**BACHELOR OF ENGINEERING (TE COMPUTERENGINEERING)**

**Academic Year: 2019-20**

**By:**

**Gaurav Borse (TECOC310)**

**Manish Choudhary (TECOC313)**

**Under The Guidance of**

**Prof . Rajesh Lomte**



**DEPARTMENT OFCOMPUTER ENGINEERING,**

**PIMPRI CHINCHWAD COLLEGE OF ENGINEERING**

**SECTOR 26, NIGDI, PRADHIKARAN**

***Introduction***

*The first year after diagnosis is a crucial time for patients with Type 2 diabetes. While it’s always important to maintain healthy blood sugar levels, new research shows that better control during the first year can reduce the future risk for complications, including kidney disease, eye disease, stroke, heart failure and poor circulation to the limbs.*

*Diabetes, often referred to by doctors as diabetes mellitus, describes a group of metabolic diseases in which the person has high blood glucose (blood sugar), either because insulin production is insufficient, or because the body's cells do not respond properly to insulin, or both. This project helps in identifying whether a person has diabetes or not, if predicted diabetic the project suggests measures for maintaining normal health and if not, diabetic it predicts the risk of getting diabetic. In this project Classification algorithm was used to classify the Pima Indian diabetes dataset. Results have been obtained using Web Application.*

**Treating diabetes early is important because it puts patients’ health on a trajectory for the rest of their lives.**

***Scope***

*It is composed for all application which deals with the health care system is one of the most common and needed test. patient from all over the world are insert their health required data into application to identify the diabetes or not in future. The work of model is to identify the patient will be diabetic in future.*

***Advantages***

* *Less time consuming.*
* *Easy to use.*
* *Simple UI.*
* *Gives details to diagnose the Diabetes.*

***Predictable System Evolution***

* *The system is portable and scalable so that it supports n number of users and can adapt to any type of establishment e.g. neighbourhood or organizations etc.*

***Product Functionality***

* *file input*
* *Predicted output*

***Dependencies***

* *Depends on the parameters such as number of different kinds of attribute to predict the patient will have diabetes in future.*

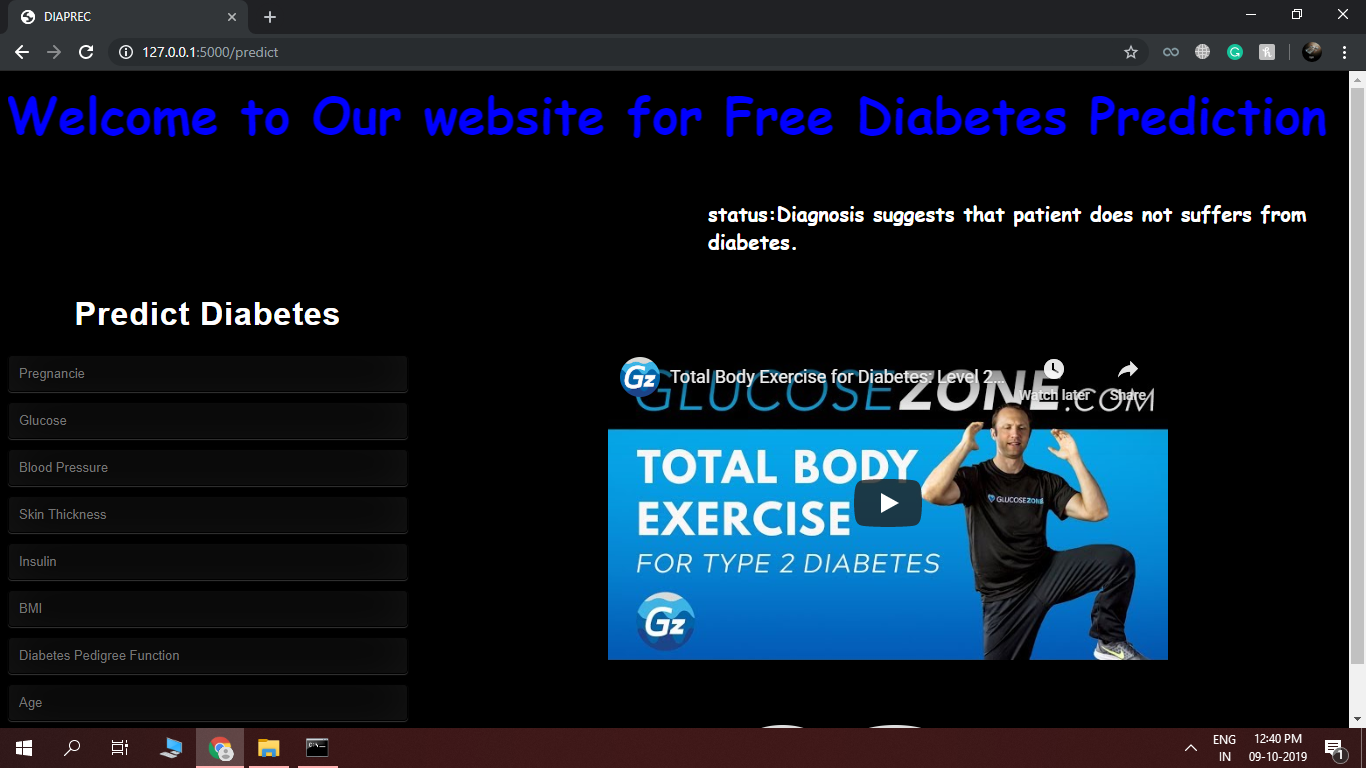
***Functional Requirements***

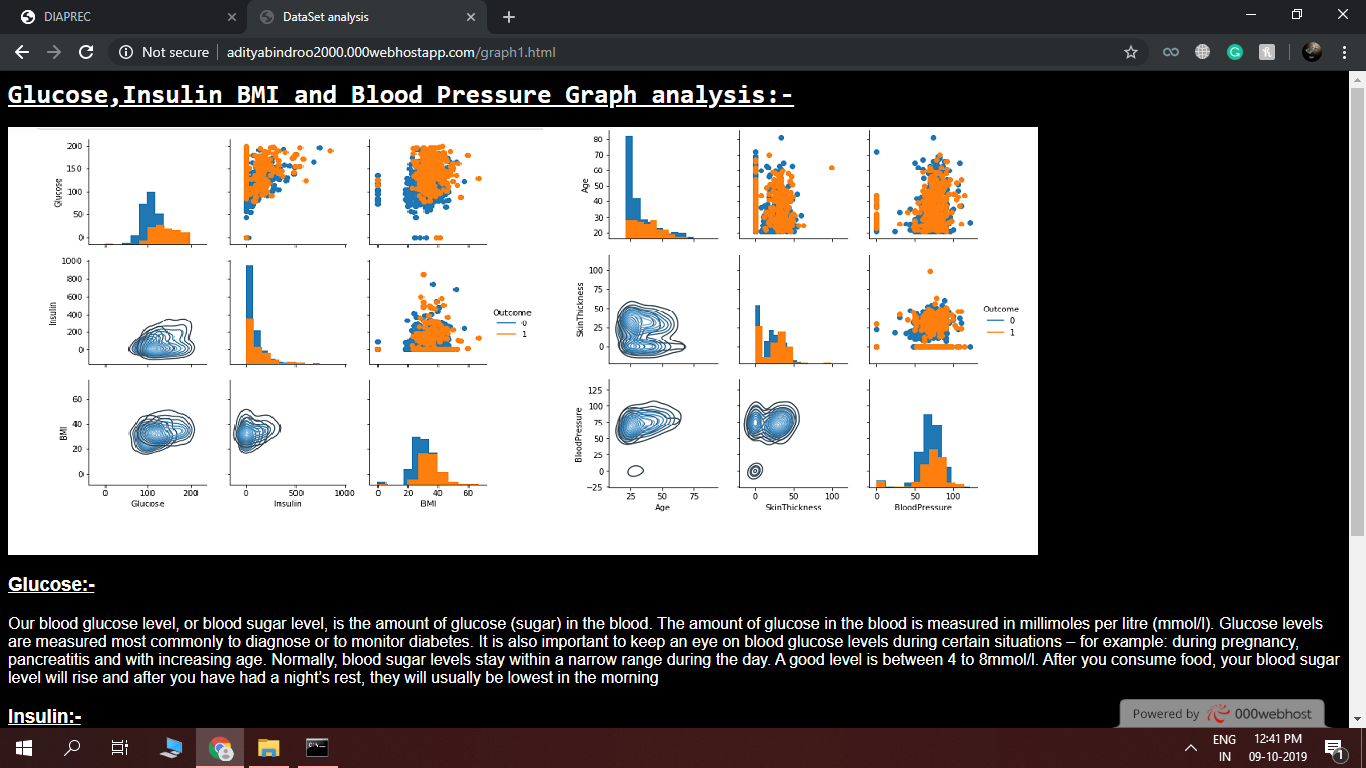
* *Csv file*
  + *The input data is taken in the form of csv file with rows and columns. The greater number of entries the clearer it becomes the machine learning module to predict it.*
* *Predicted Output*
  + *The predicted output will be in the format of binary value Simply ‘1’ as diabetic or ‘0’ as non-diabetic. There are total 9 columns in which first eight are independent data*
  + *Attribute column and the ninth column is the dependent data column.*

***Non-Functional Requirements***

* *Performance*
  + *The more you train the machine, the better accuracy it develops and thus increases performance substantially.*
* *Reliability*
  + *Machine makes fewer or no errors as compared to human calculation, so it’s quite reliable.*
* *Maintainability*
  + *Requires less or nearly no maintenance and very easy to use.*

***GUI***





***Code***

*Diabetes.py*

import numpy as np # linear algebra

import pandas as pd # data processing, CSV file I/O (e.g. pd.read\_csv)

import matplotlib.pyplot as plt

import seaborn as sns

import warnings

warnings.filterwarnings('ignore')

data = pd.read\_csv('diabetes.csv')

data.head()

# In[ ]:

import pickle

# In[4]:

data.describe()

# In[5]:

g = sns.PairGrid(data, vars=['Glucose', 'Insulin', 'BMI'], hue="Outcome", size=2.4)

g.map\_diag(plt.hist)

g.map\_upper(plt.scatter)

g.map\_lower(sns.kdeplot, cmap="Blues\_d")

g.add\_legend()

plt.show()

# In[6]:

g = sns.PairGrid(data, vars=['Age', 'SkinThickness', 'BloodPressure'], hue="Outcome", size=2.4)

g.map\_diag(plt.hist)

g.map\_upper(plt.scatter)

g.map\_lower(sns.kdeplot, cmap="Blues\_d")

g.add\_legend()

plt.show()

# In[17]:

columns = ['Glucose', 'Age', 'BloodPressure', 'Insulin','BMI','SkinThickness' ,'Pregnancies', 'DiabetesPedigreeFunction']

n\_cols = 2

n\_rows = 4

idx = 0

for i in range(n\_rows):

fg,ax = plt.subplots(nrows=1,ncols=n\_cols,sharey=True,figsize=(8, 2.4))

for j in range(n\_cols):

sns.violinplot(x = data.Outcome, y=data[columns[idx]], ax=ax[j])

idx += 1

if idx >= 8:

break

# In[7]:

max\_skinthickness = data.SkinThickness.max()

data = data[data.SkinThickness!=max\_skinthickness]

# In[8]:

def replace\_zero(df, field, target):

mean\_by\_target = df.loc[df[field] != 0, [field, target]].groupby(target).mean()

data.loc[(df[field] == 0)&(df[target] == 0), field] = mean\_by\_target.iloc[0][0]

data.loc[(df[field] == 0)&(df[target] == 1), field] = mean\_by\_target.iloc[1][0]

# run the function

for col in ['Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI']:

replace\_zero(data, col, 'Outcome')

# In[9]:

data.describe()

# In[14]:

X = data.iloc[:,:-1]

y = data.iloc[:, -1]

from sklearn.model\_selection import train\_test\_split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size = 0.3, random\_state=100)

print(X\_train.shape)

print(X\_test.shape)

print(y\_train.size)

print(y\_test.size)

# In[27]:

y\_test

# In[29]:

from sklearn.metrics import accuracy\_score

from sklearn.neighbors import KNeighborsClassifier

knn = KNeighborsClassifier(n\_neighbors=8)

clf\_ = knn.fit(X\_train, y\_train)

y\_pred = clf\_.predict(X\_test)

print('Accuracy is {}'.format(accuracy\_score(y\_test,y\_pred )))

y\_pred

# In[32]:

pickle.dump(knn,open('model.pkl','wb'))

model=pickle.load(open('model.pkl','rb'))

# In[33]:

print(model.predict([[0,190,89,35,168,74.6,0.8927,53]]))

*App.py*

import numpy as np

from flask import Flask, request, jsonify, render\_template

import pickle

app = Flask(\_\_name\_\_)

model = pickle.load(open('model.pkl', 'rb'))

@app.route('/')

def home():

return render\_template('new22.html')

@app.route('/predict',methods=['POST'])

def predict():

'''

For rendering results on HTML GUI

'''

int\_features = [float(x) for x in request.form.values()]

final\_features = [np.array(int\_features)]

prediction = model.predict(final\_features)

output = prediction

if(output==1):

out\_str="Our diagnosis suggests patient does suffer from diabetes.Please get checked soon!"

else:

out\_str="Diagnosis suggests that patient does not suffers from diabetes."

return render\_template('new22.html', prediction\_text=' status:{} '.format(out\_str))

#return render\_template('index.html', prediction\_text='Employee Salary should be $ {}'.format(output))

@app.route('/predict\_api',methods=['POST'])

def predict\_api():

'''

For direct API calls trought request

'''

data = request.get\_json(force=True)

prediction = model.predict([np.array(list(data.values()))])

output = prediction[0]

return jsonify(output)

if \_\_name\_\_ == "\_\_main\_\_":

app.run(debug=True)

***FUTURE SCOPE*:**

*In future this system can designed for any prediction of any other disease such as cancer, thyroid, lung diseases etc., if these a web application of such disease prediction would be of great use in the near future. Another future enhancement would be to reduce the no of attributes considered for the prediction purpose. Considering less no of attributes and produce more accurate results is needed as an enhancement for the existing.*

***Conclusion:***

*The discovery of knowledge from datasets is important in order to make effective diagnosis. The aim of machine is to extract information stored in dataset and generate output. This study aims at the discovery of a KNN model for the prediction of diabetes. Pre-processing is used to improve the quality of data. While pre-processing, the significant attributes of the dataset are considered for prediction of diabetes. This is an important factor for consideration. The KNN algorithm used for classification also produces maximum accuracy when compared to other algorithms of classification. Finally, the results of the system are obtained in a Web application which is very useful for the present generation.*



PIMPRI CHINCHWAD COLLEGE OF ENGINEERING

DEPARTMENT OF COMPUTER ENGINEERING

**CERTIFICATE**

This is to certify that, the project entitled

**“**Diabetes Prediction System”

is successfully carried out as a mini project successfully submitted by following students of “PCET's Pimpri Chinchwad College of Engineering, Nigdi, Pune-44**”.**

**Under the guidance of Prof. Rajesh Lomte**

In the partial fulfillment of the requirements for the T.E. (Computer Engineering)

**Gaurav Borse (TECOC310)**

**Manish Choudhary (TECOC313)**

**Mr. Rajesh Lomte**

**Project Guide**