

# Axis Business School Kanpur



**Master Of Computer Application**

**Project**

**Of**

**Diabetes Prediction and Doctor Recommendation**

**Submitted by:**

Aman

Anuj Yadav

Anupriya Kumari

Hariom

Prakhar Tiwari

Shilpi Shukla

**Submitted to:**

**HOD:** Dr. Subha Jain

**Co-ordinator:** Mrs. Sadhna Yadav

**Guided By:** Mr Chandan Verma

**Session: 2020-22**

## **Diabetes Prediction**

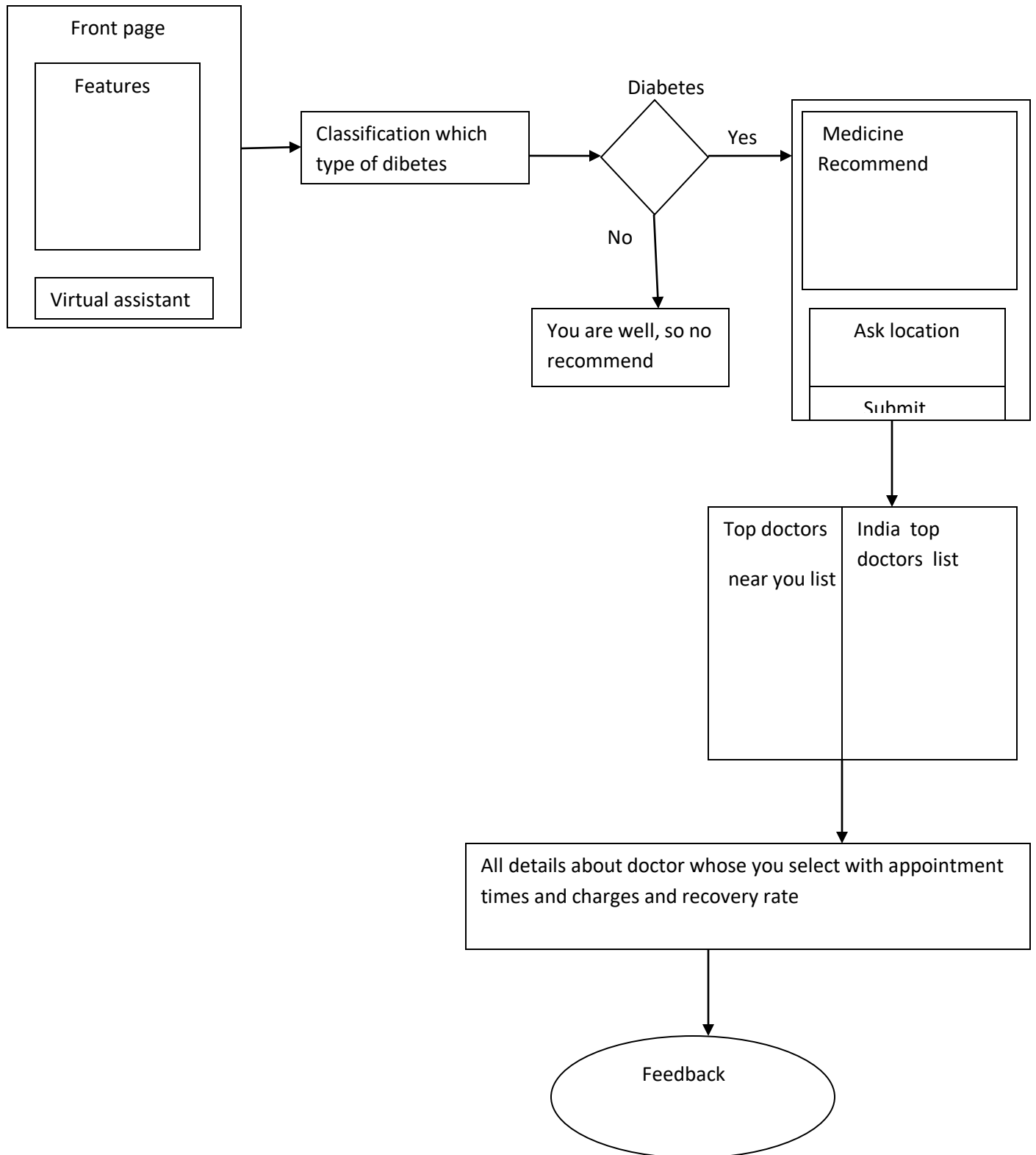
**Abstract:-** - Diabetes is an illness caused because of high glucose level in a human body. Diabetes should not be ignored if it is untreated then Diabetes may cause some major issues in a person like: heart related problems, kidney problem, blood pressure, eye damage and it can also affects other organs of human body. Diabetes can be controlled if it is predicted earlier. To achieve this goal this project work we will do early prediction of Diabetes in a human body or a patient for a higher accuracy through applying, Various Machine Learning Techniques. Machine learning techniques Provide better result for prediction by constructing models from datasets collected from patients. In this work we will use Machine Learning Classification and ensemble techniques on a dataset to predict diabetes. Which are K-Nearest Neighbor (KNN), Logistic Regression (LR) and Random Forest (RF). The accuracy is different for every model when compared to other models. The Project work gives the accurate or higher accuracy model shows that the model is capable of predicting diabetes effectively. Our Result shows that Random Forest achieved higher accuracy compared to other machine learning techniques.

**Introduction:-** Diabetes is noxious diseases in the world. Diabetes caused because of obesity or high blood glucose level, and so forth. It affects the hormone insulin, resulting in abnormal metabolism of carbs and improves level of sugar in the blood. Diabetes occurs when body does not make enough insulin. According to (WHO) World Health

Organization about 422 million people suffering from diabetes particularly from low or idle income countries. And this could

Be increased to 490 billion up to the year of 2030. However prevalence of diabetes is found among various Countries like Canada, China, and India etc. Population of India is now more than 100 million so the actual number of diabetics in India is 40 million. Diabetes is major cause of death in the world. Early prediction of disease like diabetes can be controlled and save the human life. To accomplish this, this work explores prediction of diabetes by taking various attributes related to diabetes disease. For this purpose we use the Pima Indian Diabetes Dataset, we apply various Machine Learning classification and ensemble Techniques to predict diabetes. Machine Learning Is a method that is used to train computers or machines explicitly. Various Machine Learning Techniques provide efficient result to collect Knowledge by building various classification and ensemble models from collected dataset. Such collected data can be useful to predict diabetes. Various techniques of Machine Learning can capable to do prediction, however it's tough to choose best technique. Thus for this purpose we apply popular classification and ensemble methods on dataset for prediction.

## Architecture of diabetes Prediction:



**Proposed Methodology:-** Goal of the paper is to investigate for model to predict diabetes with better accuracy. We experimented with different classification and ensemble algorithms to predict diabetes. In the following, we briefly discuss the phase.

**A. Dataset Description**-the data is gathered from UCI repository which is named as Pima Indian Diabetes Dataset. The dataset have many attributes of 768 patients.

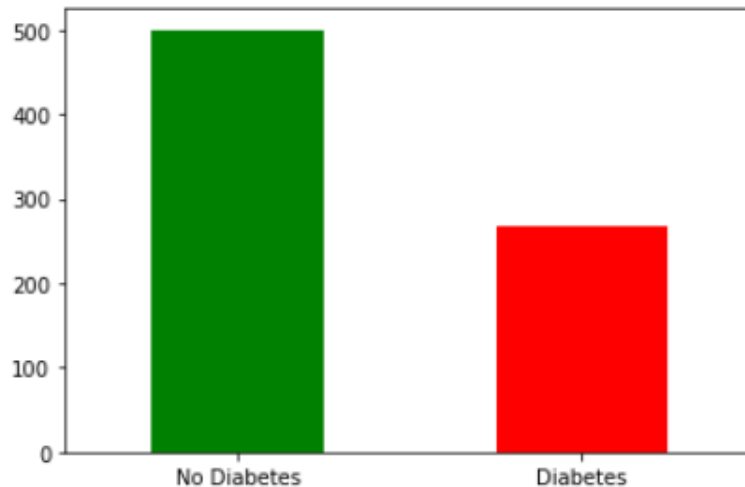
Table 1: **Dataset Description**

S No.	Attributes
1	Pregnancy
2	Glucose
3	Blood Pressure
4	Skin thickness
5	Insulin
6	BMI(Body Mass Index)
7	Diabetes Pedigree Function
8	Age

The 9<sup>th</sup> attribute is class variable of each data points. This class variable shows the outcome 0 and 1 for diabetics which indicates positive or negative for diabetics.

**Distribution of Diabetic patient-** We made a model to predict diabetes however the dataset was slightly imbalanced having around 500 classes labeled as 0 means negative means no diabetes and 268 labeled as 1 means positive means diabetic.

```
data["Outcome"].value_counts().plot(kind="bar",color=["green","red"])  
plt.xticks(np.arange(2), ('No Diabetes', 'Diabetes'),rotation=0);
```



**B. Data Preprocessing :-** Data preprocessing is most important process. Mostly healthcare related data contains missing value and other impurities that can cause effectiveness of x data. To improve quality and effectiveness obtained after mining process, Data preprocessing is done. To use Machine Learning Techniques on the dataset effectively this process is essential for accurate result and successful prediction. For Pima Indian diabetes dataset we need to perform pre processing in two steps.

**C. Apply Machine Learning :-** - When data has been ready we apply Machine Learning Technique. We use different classification and ensemble techniques, to predict diabetes. The methods applied on Pima Indians diabetes dataset. Main objective to apply Machine Learning Techniques to analyze the performance of these methods and find accuracy of them, and also been able to figure out the responsible/important feature which play a major role in prediction. The Techniques are follows -

```
In [2]: #models
from sklearn.ensemble import RandomForestClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import SVC
```

1) **K-Nearest Neighbor** - KNN is also a supervised machine learning algorithm. KNN helps to solve both the classification and regression problems. KNN is lazy prediction technique. KNN assumes that similar things are near to each other. Many times data points which are similar are very near to each other. KNN helps to group new work based on similarity measure. KNN algorithm record all the records and classify them according to their similarity measure. For finding the distance between the points uses tree like structure. To make a prediction for a new data point, the algorithm finds the closest data points in the training data set — it's nearest neighbors. Here K= Number of nearby neighbors, it's always a positive integer. Neighbor's value is chosen from set of class. Closeness is mainly defined in terms of Euclidean distance. The Euclidean distance between two points P and Q i.e. P (p1,p2, .... Pn) and Q (q1, q2,..qn) is defined by the following equation:-

$$d(P, Q) = \sum_{i=1}^n (P_i - Q_i)^2$$

### **Algorithm-**

- Take a sample dataset of columns and rows named as Pima indian Diabetes I data set.
- Take a test dataset of attributes and rows.
- Find the Euclidean distance by the help of formula-

$$EuclideanDistance = \sqrt{\sum_{i=1}^y \sum_{j=1}^m \sum_{l=1}^{n-1} (R_{(j,l)} - P_{(i,l)})^2}$$

- Then, Decide a random value of K. is the no. of nearest neighbors.
- Then with the help of these minimum distance and Euclidean d distance find out the nth column of each.
- Find out the same output values.

“ If the values are same, then the patient is diabetic, otherwise not. ”

## 2) Logistic Regression :-

Logistic regression is also a supervised learning classification algorithm. It is used to estimate the probability of a binary response based on one or more predictors. They can be continuous or discrete. Logistic regression used when we want to classify or distinguish some data items into categories. It classify the data in binary form means only in 0 and 1 which refer case to classify patient that is positive or negative for diabetes. Main aim of logistic regression is to best fit which is responsible for describing the relationship between target and predictor variable. Logistic regression is a based on Linear regression model. Logistic regression model uses sigmoid function to predict probability of positive and negative class. Sigmoid function  $P = \frac{1}{1+e^{-(a+bx)}}$  Here P = probability, a and b = parameter of Model.

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

## Ensembling :-

is a machine learning technique Ensemble means using multiple learning algorithms together for some task. It provides better prediction than any other individual model that's why it is used. The main cause of error is noise bias and variance, ensemble methods help to reduce or minimize

these errors. There are two popular ensemble methods such as – Bagging, Boosting, ada-boosting, Gradient boosting, voting, averaging etc. Here In these work we have used Bagging (Random forest) and Gradient boosting ensemble methods for predicting diabetes.

5) **Random Forest** – – It is type of ensemble learning method and also used for classification and regression tasks. The accuracy it gives is grater then compared to other models. This method can easily handle large datasets. Random Forest is developed by Leo Bremen. It is popular ensemble Learning Method. Random Forest Improve Performance of Decision Tree by reducing variance. It operates by constructing a multitude of decision trees at training time and outputs the class that is the mode of the classes or classification or mean prediction (regression) of the individual trees.

### **Algorithm-**

- The first step is to select the “R” features from the total features “m” where  $R \ll M$ .
- Among the “R” features, the node using the best split point.
- Split the node into sub nodes using the best split.
- Repeat a to c steps until ”l” number of nodes has been reached.
- Built forest by repeating steps a to d for “a” number of times to create “n” number of trees.

The random forest finds the best split using the Gin-Index Cost Function which is given by:

$$Gini = \sum_{k=1}^n p_k * (1 - p_k) \text{ Where } k = \text{Each class and } p = \text{proption of training instances}$$

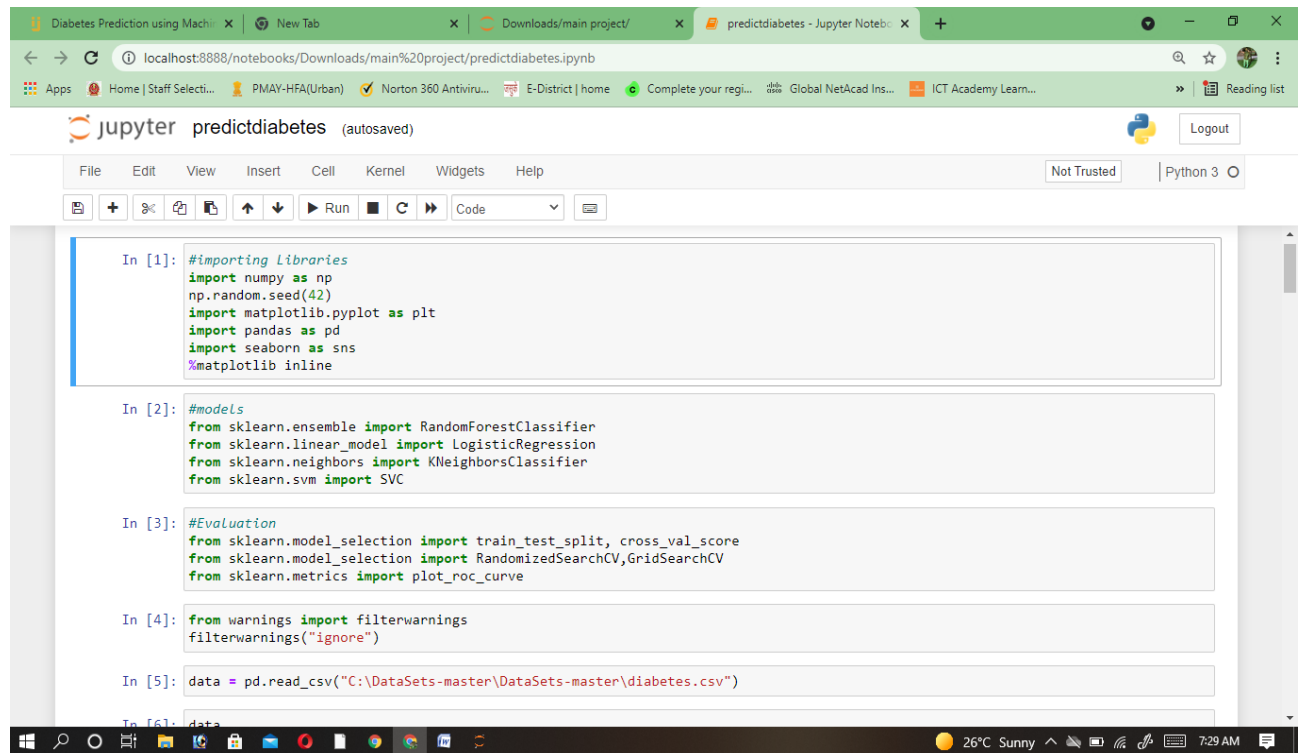
The first step is to need the take a glance at choices and use the foundations of each indiscriminately created decision tree to predict the result and stores the anticipated outcome at intervals the target place. Secondly, calculate the votes for each predicted target and ultimately, admit the high voted predicted target as a result of the ultimate prediction from the random forest formula. Some of the options of Random Forest does correct predictions result for a spread of applications are offered.



## Accuracy Table

<u>Algorithms</u>	<u>Accuracy</u>
KNN	77%
Logistic Regression	83%

## Screen Shot of code :-



```
In [1]: #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

In [2]: #models
from sklearn.ensemble import RandomForestClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import SVC

In [3]: #Evaluation
from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.model_selection import RandomizedSearchCV, GridSearchCV
from sklearn.metrics import plot_roc_curve

In [4]: from warnings import filterwarnings
filterwarnings("ignore")

In [5]: data = pd.read_csv("C:\DataSets-master\DataSets-master\diabetes.csv")

In [6]: data
```

Diabetes Prediction using Machi x New Tab x Downloads/main project/ x predictdiabetes - Jupyter Notebo x +

localhost:8888/notebooks/Downloads/main%20project/predictdiabetes.ipynb

Apps Home | Staff Selecti... PMAY-HFA(Urban) Norton 360 Antiviru... E-District | home Complete your regi... Global NetAcad Ins... ICT Academy Learn... Reading list

jupyter predictdiabetes (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3

In [6]: data

Out[6]:

	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 x 9 columns

In [7]: data.isna().sum()

Out[7]: Pregnancies 0

Glucose

Diabetes Prediction using Machi x New Tab x Downloads/main project/ x predictdiabetes - Jupyter Notebo x +

localhost:8888/notebooks/Downloads/main%20project/predictdiabetes.ipynb

Apps Home | Staff Selecti... PMAY-HFA(Urban) Norton 360 Antiviru... E-District | home Complete your regi... Global NetAcad Ins... ICT Academy Learn... Reading list

jupyter predictdiabetes (autosaved) Logout

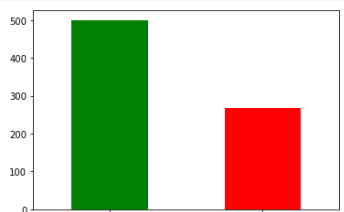
File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3

In [7]: data.isna().sum()

Out[7]:

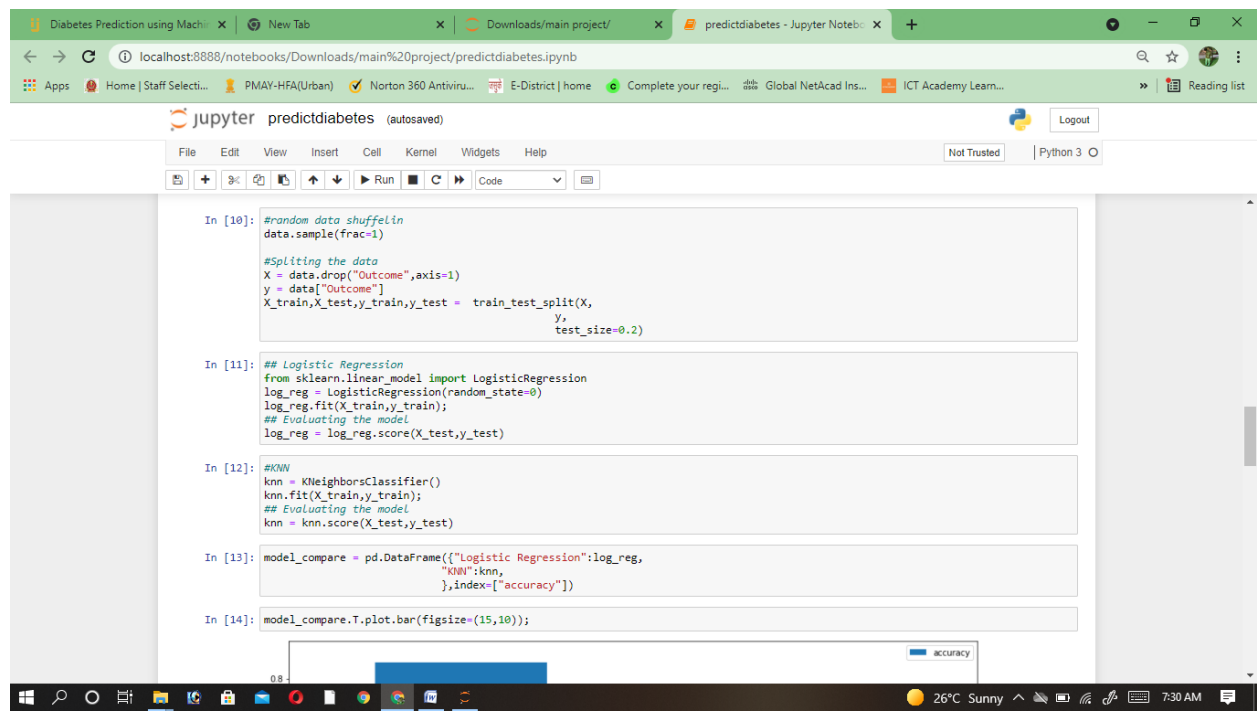
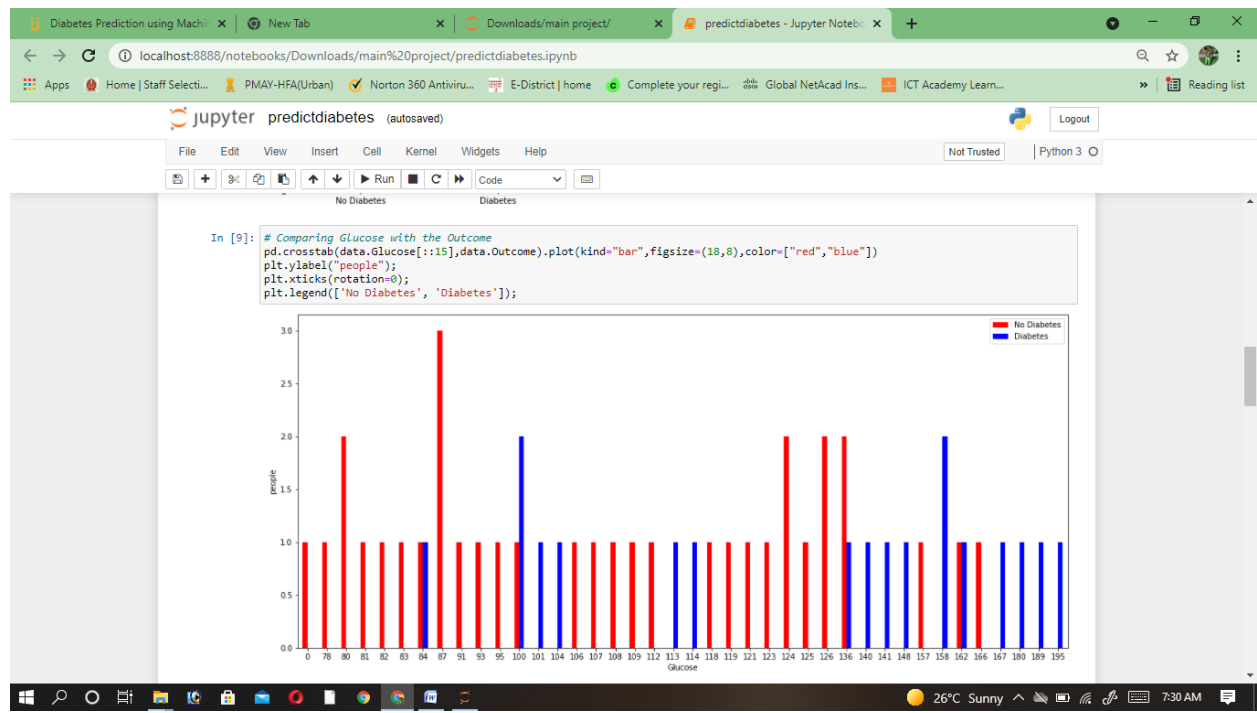
```
Pregnancies      0
Glucose          0
BloodPressure    0
SkinThickness    0
Insulin          0
BMI              0
DiabetesPedigreeFunction  0
Age              0
Outcome          0
dtype: int64
```

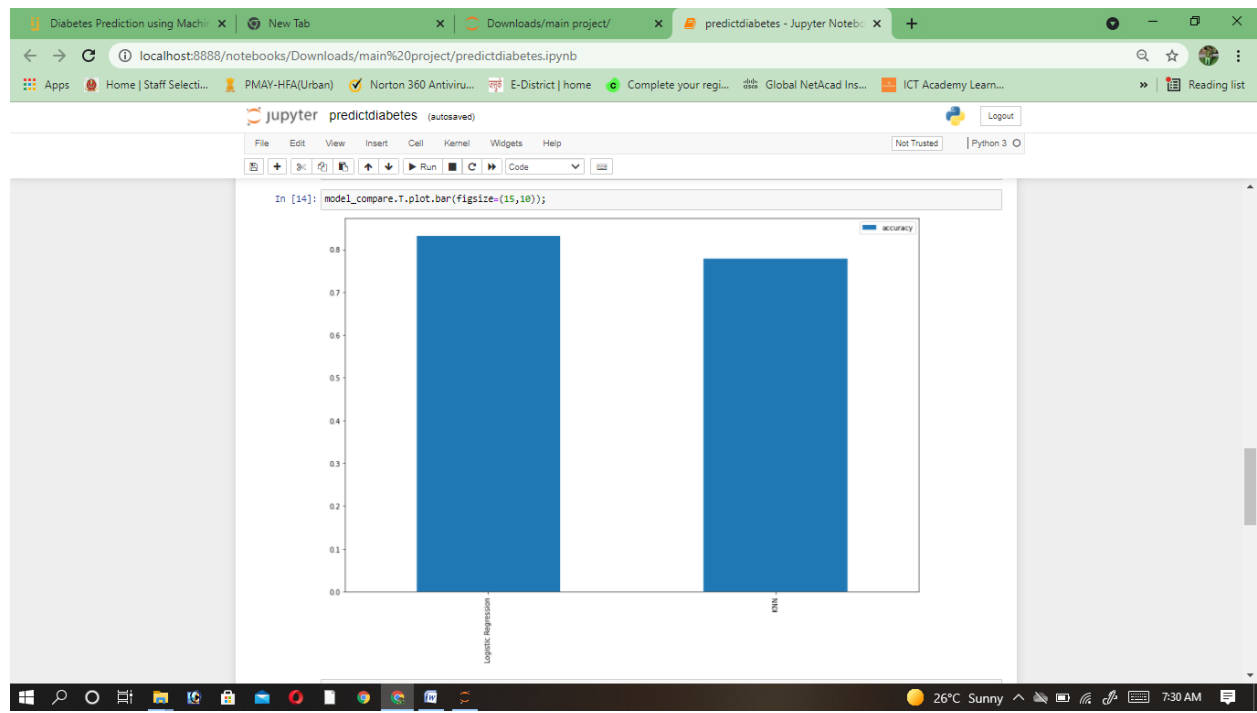
In [8]: data["Outcome"].value\_counts().plot(kind="bar",color=["green","red"])
plt.xticks(np.arange(2), ('No Diabetes', 'Diabetes'),rotation=0);



Outcome	Count
No Diabetes	500
Diabetes	270

26°C Sunny 7:29 AM





Diabetes Prediction using Machi x New Tab x Downloads/main project/ x predictdiabetes - Jupyter Notebo x +

localhost:8888/notebooks/Downloads/main%20project/predictdiabetes.ipynb

Apps Home | Staff Selecti... PMAY-HFA(Urban) Norton 360 Antiviru... E-District | home Complete your regi... Global NetAcad Ins... ICT Academy Learn... Reading list

jupyter predictdiabetes (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3

In [15]: `log_reg_grid = {'C': np.logspace(-4,4,30),  
 'solver':['liblinear']}  
  
#setup the grid cv  
gs_log_reg = GridSearchCV(LogisticRegression(),  
 param_grid=log_reg_grid,  
 cv=5,  
 verbose=True)  
  
#fit grid search cv  
gs_log_reg.fit(X_train,y_train)`

Fitting 5 folds for each of 30 candidates, totalling 150 fits

Out[15]: `GridSearchCV(cv=5, estimator=LogisticRegression(),  
 param_grid={'C': array([1.00000000e-04, 1.88739182e-04, 3.56224789e-04, 6.72335754e-04,  
1.26896100e-03, 2.39502662e-03, 4.52035360e-03, 8.53167852e-03, 1.61026203e-02, 3.03919538e-02, 5.73615251e-02, 1.08263673e-01,  
2.04335972e-01, 3.85662042e-01, 7.27895384e-01, 1.37382380e+00, 2.59294380e+00, 4.89390092e+00, 9.23670857e+00, 1.74332882e+01, 3.29034450e+01, 6.21016942e+01, 1.17210230e+02, 2.21221629e+02, 4.17531894e+02, 7.88046282e+02, 1.48735211e+03, 2.80721620e+03, 5.29831691e+03, 1.00000000e+04]),  
 solver='liblinear'},  
 verbose=True)`

In [16]: `gs_log_reg.score(X_test,y_test)`

Out[16]: 0.8376623376623377

Windows taskbar: 26°C Sunny 7:31 AM

Diabetes Prediction using Machi x New Tab x Downloads/main project/ x predictdiabetes - Jupyter Notebo x +

localhost:8888/notebooks/Downloads/main%20project/predictdiabetes.ipynb

Apps Home | Staff Selecti... PMAY-HFA(Urban) Norton 360 Antiviru... E-District | home Complete your regi... Global NetAcad Ins... ICT Academy Learn... Reading list

jupyter predictdiabetes (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3

In [17]:

```
import pickle

# Save trained model to file
pickle.dump(gs_log_reg, open("Diabetespre.pkl", "wb"))
```

In [18]:

```
loaded_model = pickle.load(open("Diabetespre.pkl", "rb"))
loaded_model.predict(X_test)
loaded_model.score(X_test,y_test)
```

Out[18]: 0.8376623376623377

In [19]:

```
Pregnancies = input()
Glucose = input()
BloodPressure = input()
SkinThickness = input()
Insulin = input()
BMI = input()
DiabetesPedigreeFunction = input()
Age = input()

75
45
45
45
45
45
45
45
45
```

26°C Sunny 7:31 AM

Diabetes Prediction using Machi x New Tab x Downloads/main project/ x predictdiabetes - Jupyter Notebo x +

localhost:8888/notebooks/Downloads/main%20project/predictdiabetes.ipynb

Apps Home | Staff Selecti... PMAY-HFA(Urban) Norton 360 Antiviru... E-District | home Complete your regi... Global NetAcad Ins... ICT Academy Learn... Reading list

jupyter predictdiabetes (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3

75  
45  
45  
45  
45  
45  
45  
45  
45

In [20]:

```
row_df = pd.DataFrame([pd.Series([Pregnancies,Glucose,BloodPressure,SkinThickness,Insulin,BMI ,DiabetesPedigreeFunction ,Age])])
```

In [21]:

```
prob = loaded_model.predict_proba(row_df)[0][1]
print(f"The probability of you having Diabetes is {prob}")

The probability of you having Diabetes is 1.0
```

In [22]:

```
loaded_model.predict(row_df)[0]
```

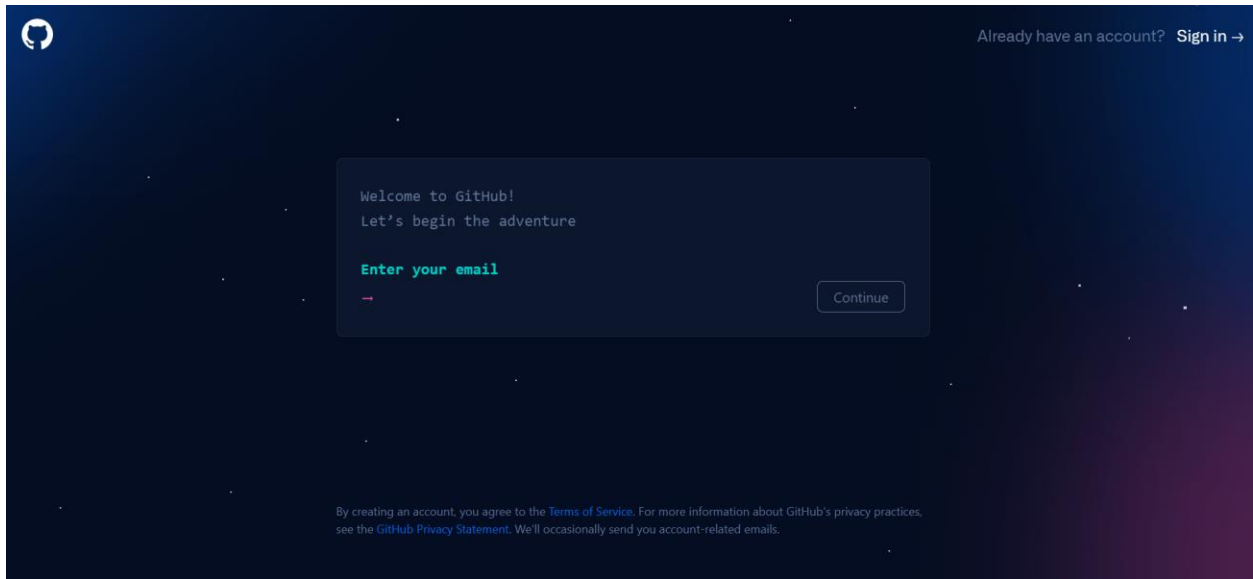
Out[22]: 1

In [ ]:

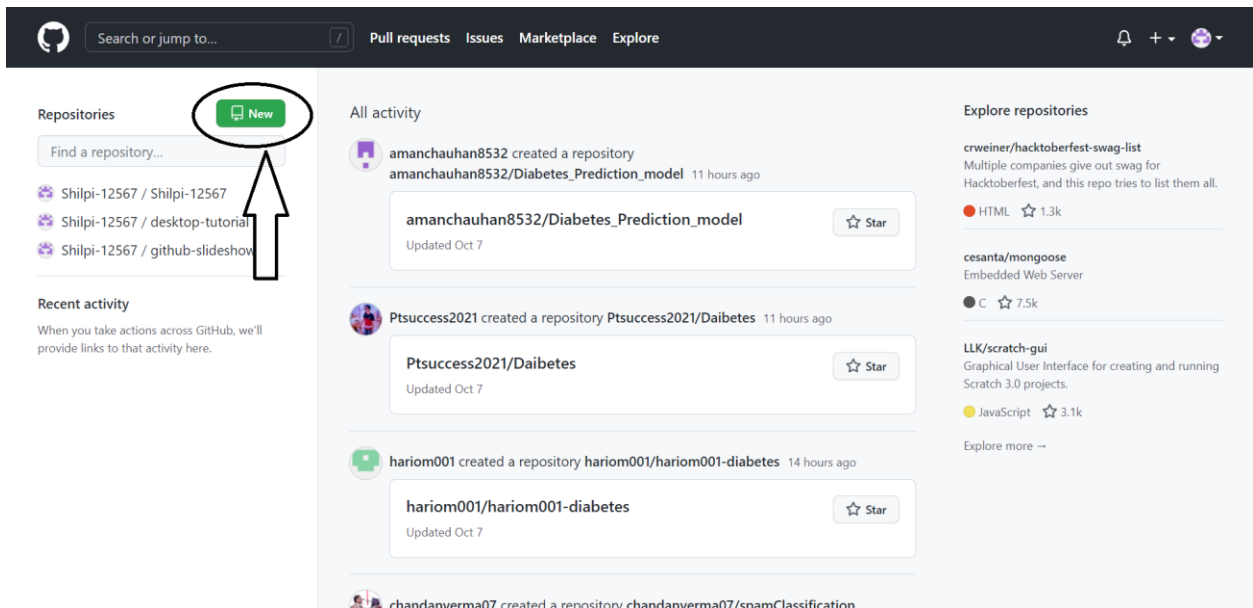
26°C Sunny 7:32 AM

## PROCESS OF MODEL DEPLOYING HEROKU:

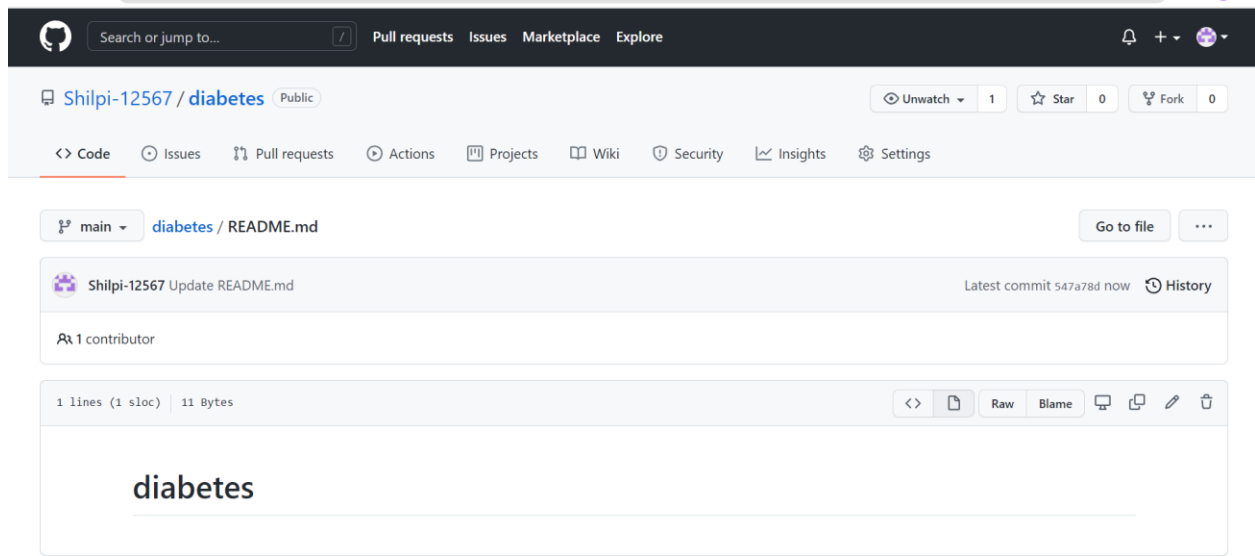
Step 1: Create the git hub id.



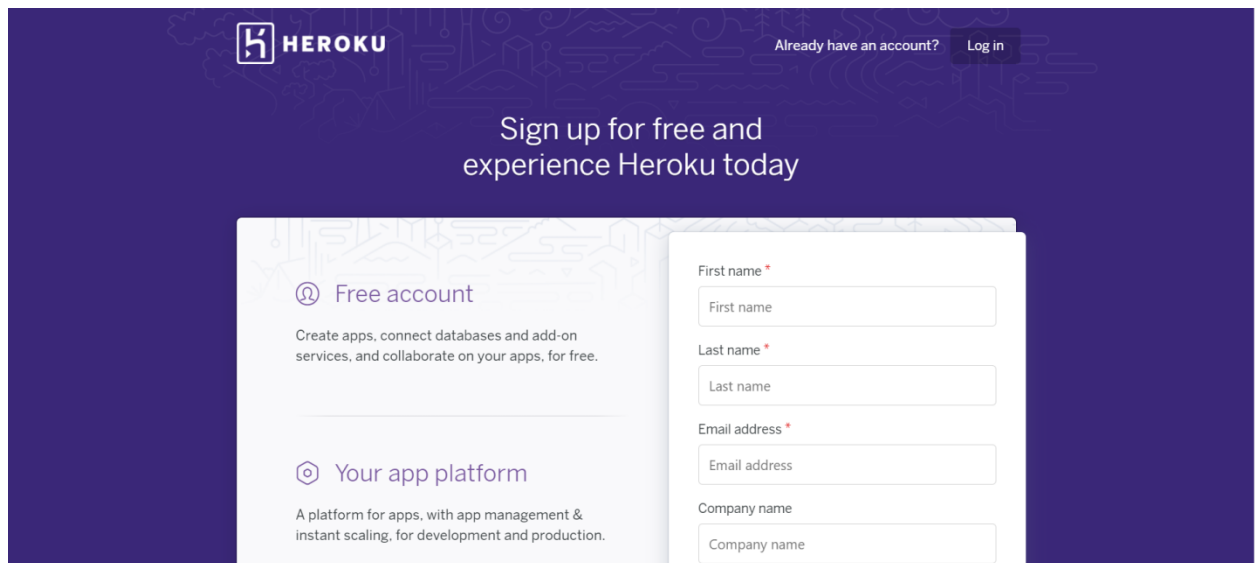
Step 2: Create repository.



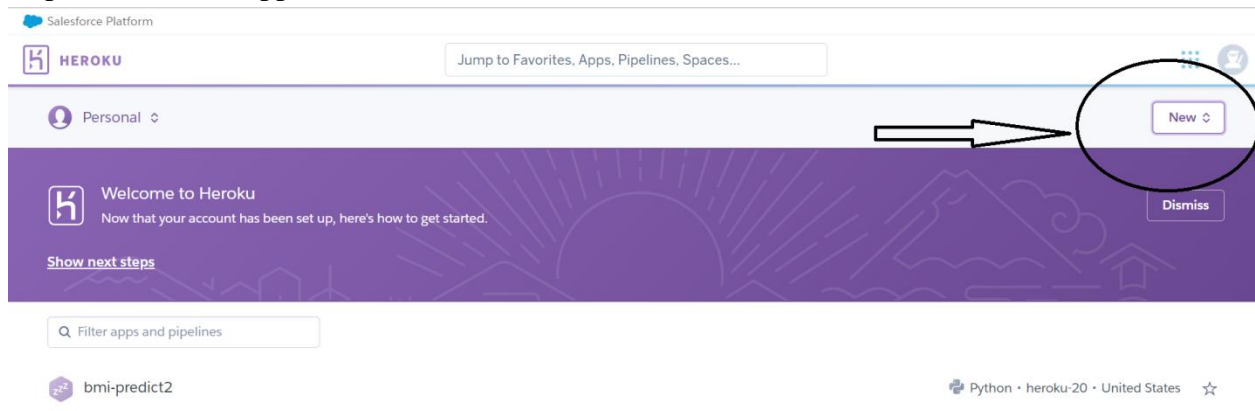
Step 3: Upload file on created.



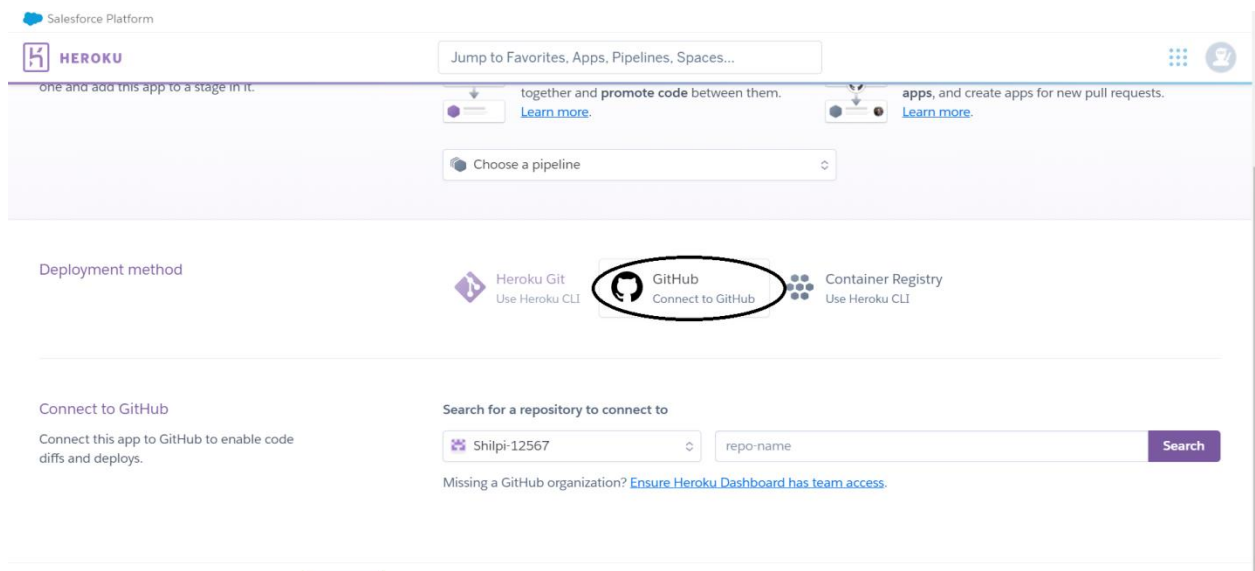
Step 4: Create id on heroku.



## Step 5: Create new app name.

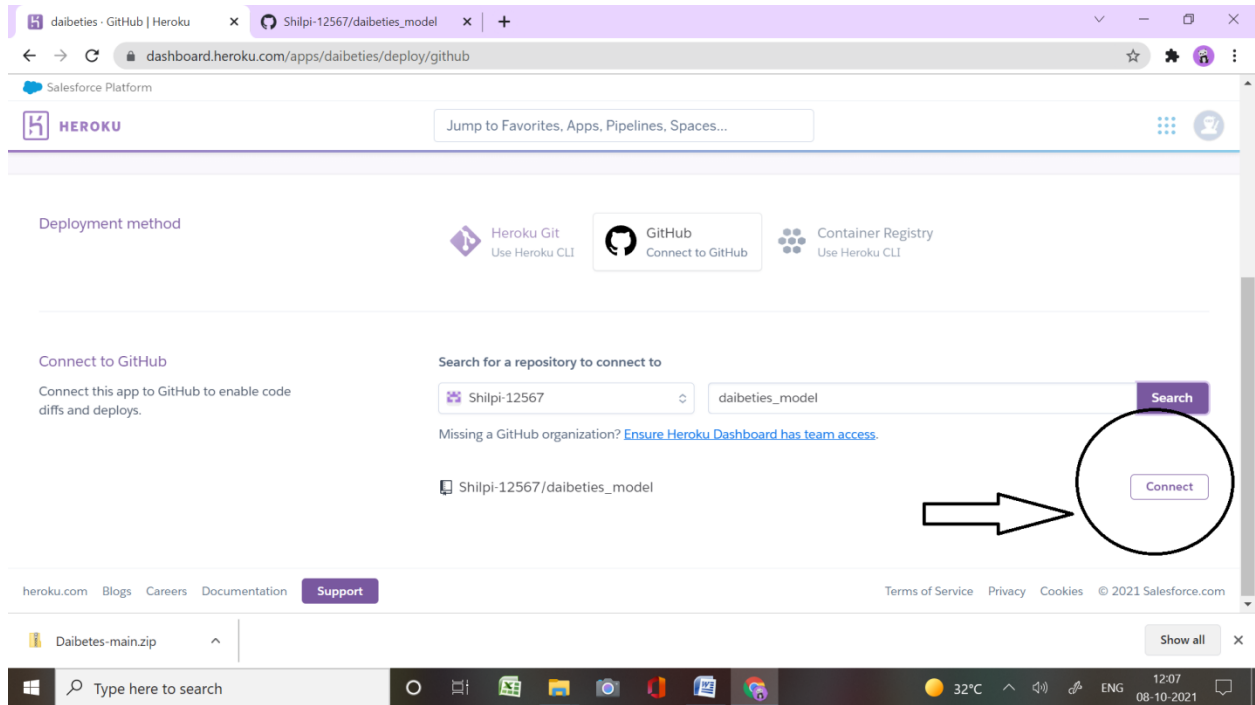


## Step 6: Connect github id.

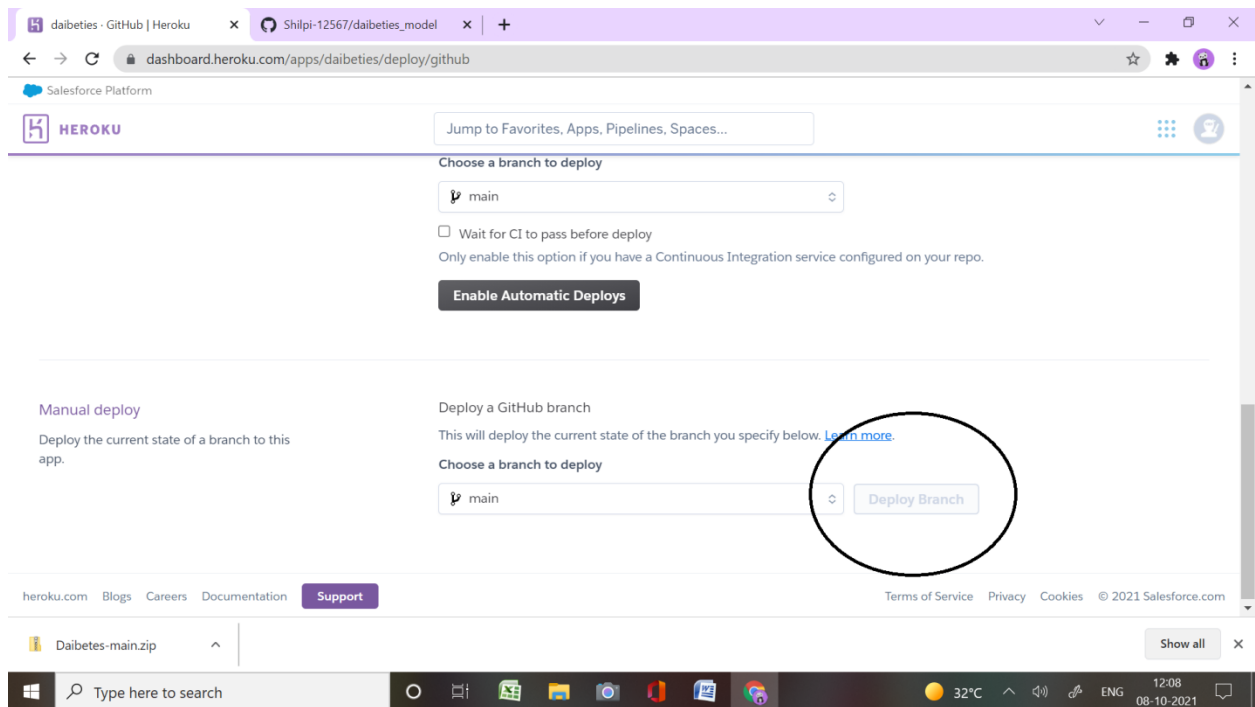


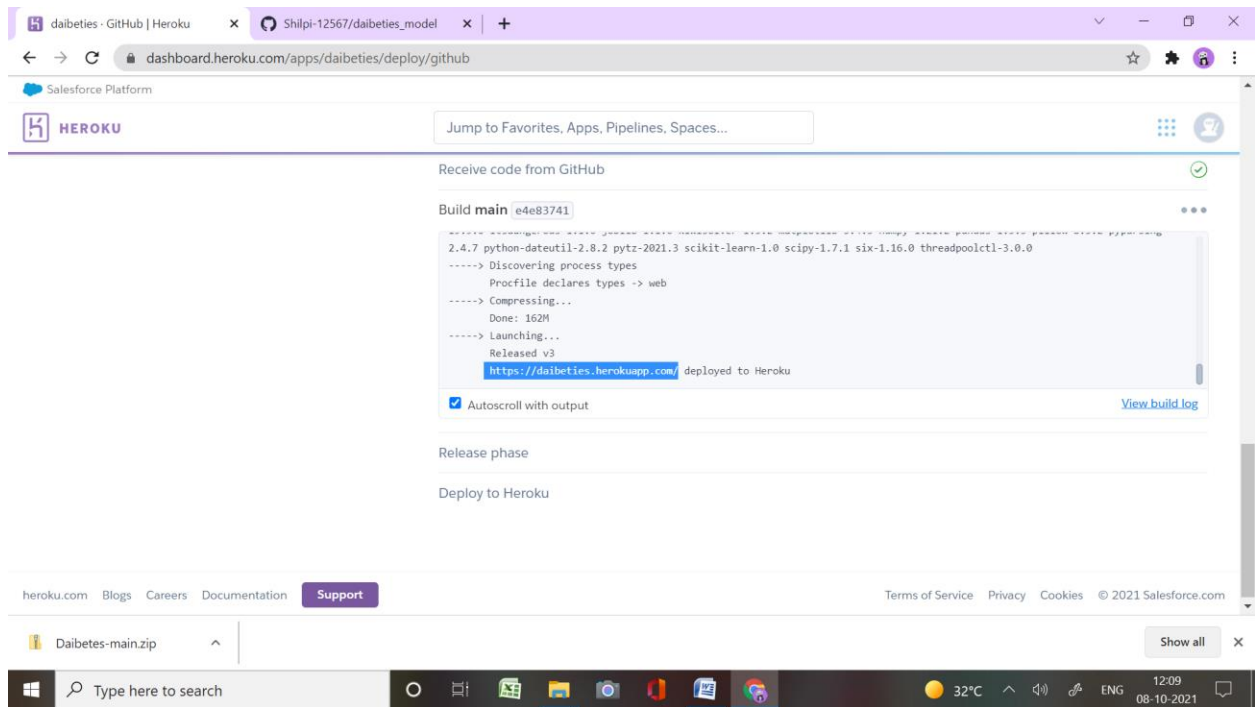
## Step 7: Connect repository.



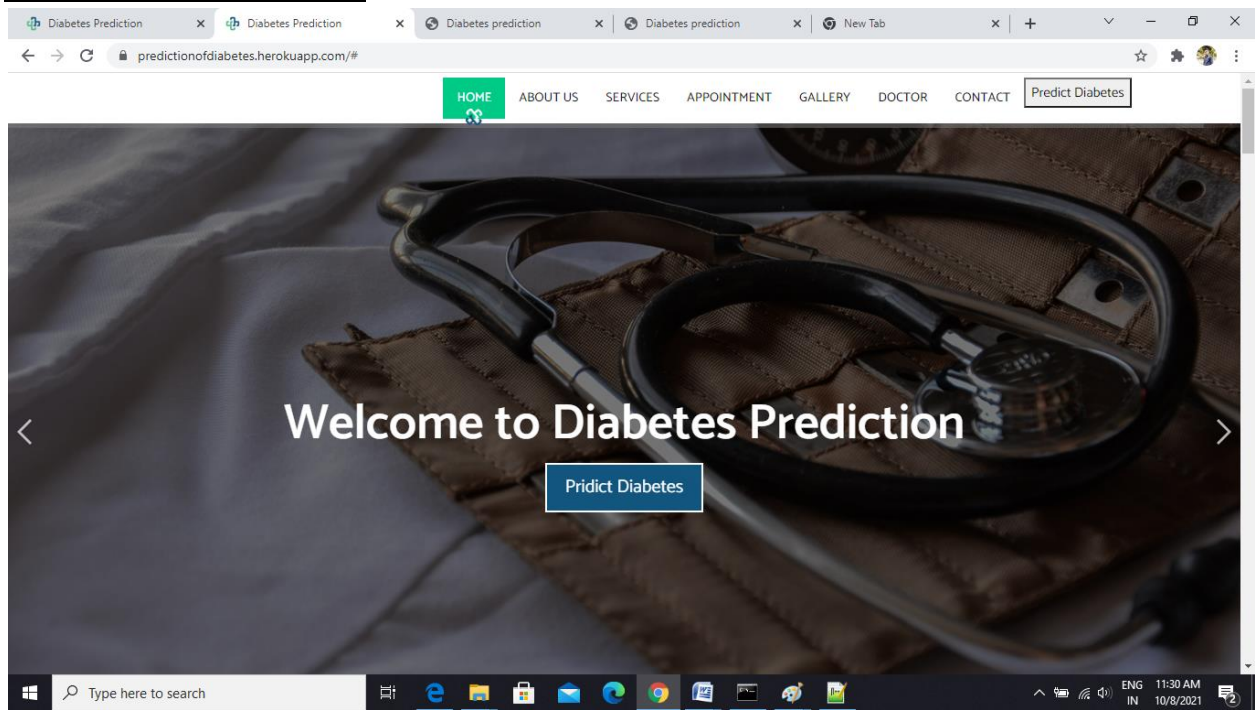


## Step 8: Deploy main and success deploy.





## Output Of The Model:



Diabetes Prediction

Home

# Diabetes Prediction

Predict the probability of having Diabetes

<b>Pregnancies</b> No. of Pregnancies	<b>Glucose</b> Glucose level in sugar	<b>BloodPressure</b> BloodPressure
<b>SkinThickness</b> SkinThickness	<b>Insulin</b> Insulin level	<b>BMI</b> Body Mass Index
<b>DiabetesPedigreeFunction</b> DiabetesPedigreeFunction	<b>Age</b> Age	

PREDICT PROBABILITY

Diabetes Prediction

Home Diabetes Prediction

You have chance of having diabetes. Probability of having Diabetes is 2.0%

>

Aman

amanchauhan8532@gmail.com

**Conclusion:** In this study, various machine learning algorithms are applied on the dataset and the classification has been done using various algorithms of which Logistic

Regression gives highest accuracy of 83%. It is clear that the model improves accuracy and precision of diabetes prediction with this dataset compared to existing dataset. Further this work can be extended to find how likely non-diabetic people can have diabetes in next few years .

**Link Of Our Deployed Model :** <https://predictionofdiabetes.herokuapp.com/>

**GitHub Id:**

- amanchauhan8532
- anuj0427
- anupriyaaxis
- hariom001
- shilpi-12567
- Ptsuccess2021

**Reference:**

- For collection of diabetes data we use –<https://www.kaggle.com/>
- Hands on machine learning with Scikit-learn, keras, and tensorflow: concepts, tools and technologies to build intelligent systems.