Diabetes Prediction Application

Report on the project done as part of Gurucool Program by IBM in association with Smart Internz from 28th Sep, 2020 to 6th Oct, 2020.

1. Introduction

AI (Artificial Intelligence) is a wide domain which contains wide spectrum of methods for modelling variety of solutions to problems. It has humongous applications in healthcare domain, mainly in disease prediction. Diabetes is a metabolic disorder which is the result of defects in insulin secretion and insulin action. Various life style and other parameters have an impact on the onset and progression of this disease. These parameters and the outcome for number of patients already available in the domain can be used to make the machine learn from and predict with good amount of accuracy when new/unseen data points are posed to it. This would open new avenues for machine learning in the area of healthcare that would assist the healthcare stakeholders to make informed decisions by studying large number of patient data. This would also enable them to strategize customized treatment methods to better meet the varying disease symptoms in case of chronic diseases like diabetes.

1.1. Overview

This Diabetes prediction application is built by using IBM Watson Studio associated with Machine Learning service to perform the prediction task and Node-Red is used to build the web application.

1.2. Purpose

The purpose of building this application is to predict whether or not a patient acquires diabetes based on some diagnostic features. Though this application would not substitute a doctor or professional medical advice, it serves as a preliminary investigator inot the symptoms a person has w.r.t. diabetes.

2. Literature Survey

2.1 Existing Problem

The existing systems like glucose level test that can be done at home, though almost accurate, can monitor only one parameter at two levels namely fasting sugar and post-prandial, for prediction of the diabetes disease. The test results obtained from labs have to be further taken to doctor for medical advice. But, the patient will be curious to quickly know about his/her health status before even consulting a doctor. They can use existing apps to check the status but still the existing ones use limited diagnostic parameters and the prediction is not very much reliable.

2.2 Proposed System

The proposed system uses 8 vital parameters that can predict the chances of acquiring the diabetes which uses machine learning techniques to learn from hundreds of patient data, fit the best performing model consistent with the training examples given and it is tested/validated with unseen data also. It is found to perform with about 78% accuracy in its predictions. So, an application built on this will be highly reliable with good prediction accuracy.

3. Theoretical Analysis

3.1 Block Diagram

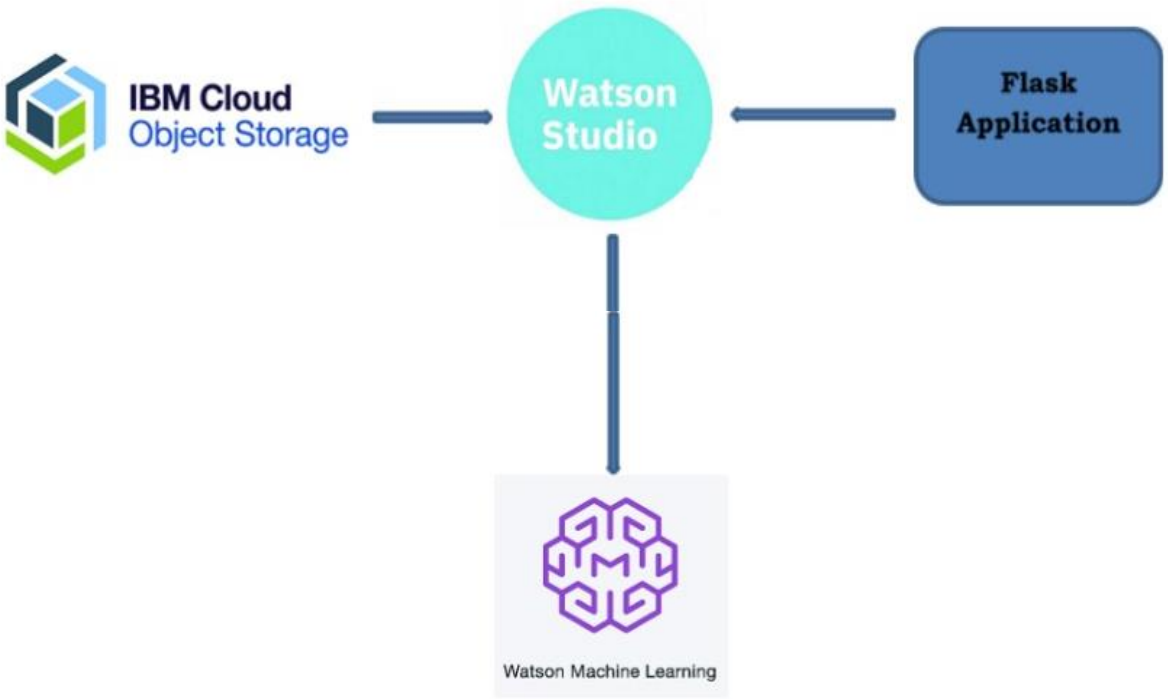


Fig. 3.1 Proposed Technical Architecture

3.2 Hardware/ Software designing

- IBM Watson Studio
- AI/Machine Learning Service
- IBM Cloud Object Storage
- Node Red flow-based development tool

4. Dataset collection

Pima Indians Diabetes dataset was collected from Kaggle. All data is from females atkeast 21 years old of Pima Indian heritage.

- **Dataset description**

1.	No. of Pregnancies	-	Number of times pregnant
2.	Random Glucose	-	Plasma glucose concentration a 2 hours in an oral glucose tolerance test
3.	Blood Pressure	-	Diastolic blood pressure (mm Hg)
4.	Skin Thickness	-	Triceps skin fold thickness (mm)
5.	Insulin	-	2-Hour serum insulin (mu U/ml)
6.	Body Mass Index	-	Body mass index (weight in kg/(height in m)^2)
7.	Diabetes Pedigree Function	-	A function which scores the likelihood of diabetes based on family history. It provided some data on diabetes mellitus history in relatives and the genetic relationship of those relatives to the patient
8.	Age	-	Age (in years)
9.	Outcome	-	The outcome label 1 for Yes (for chances of acquiring diabetes and 0 for No (for no chances of acquiring diabetes)

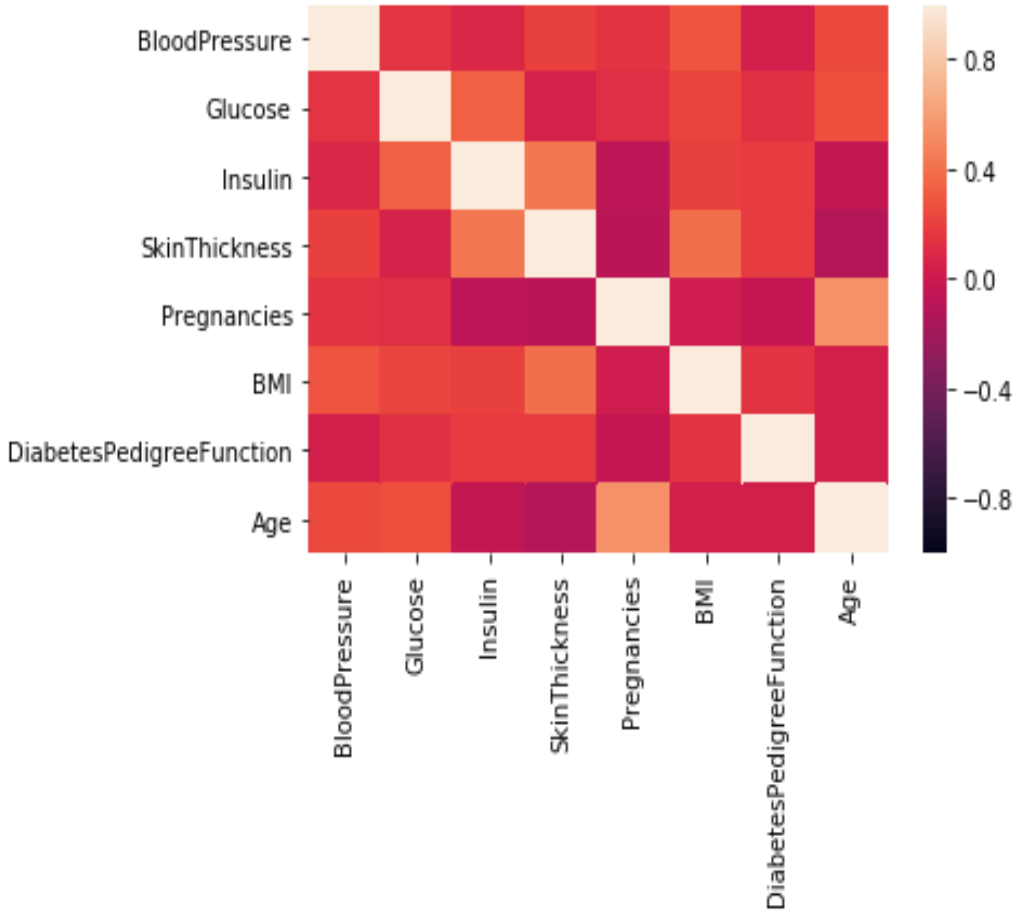
5. Data Visualization

Exploratory data analysis and visualization helps us to understand the characteristics and distribution of the data that enable us to take informed decisions.

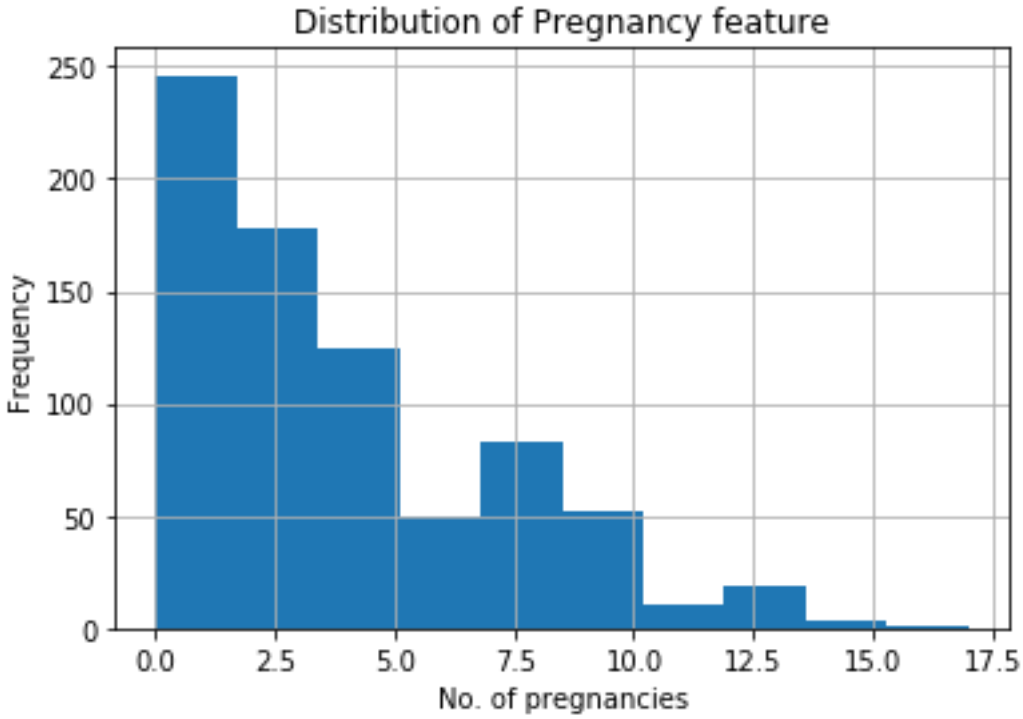
i. Five-point summary

	BloodPressure	Glucose	Insulin	SkinThickness	Pregnancies	BMI	DiabetesPedigreeFunction	Age
count	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000
mean	69.105469	120.894531	79.799479	20.536458	3.845052	31.992578	0.471876	33.240885
std	19.355807	31.972618	115.244002	15.952218	3.369578	7.884160	0.331329	11.760232
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.078000	21.000000
25%	62.000000	99.000000	0.000000	0.000000	1.000000	27.300000	0.243750	24.000000
50%	72.000000	117.000000	30.500000	23.000000	3.000000	32.000000	0.372500	29.000000
75%	80.000000	140.250000	127.250000	32.000000	6.000000	36.600000	0.626250	41.000000
max	122.000000	199.000000	846.000000	99.000000	17.000000	67.100000	2.420000	81.000000

ii. Correlation map

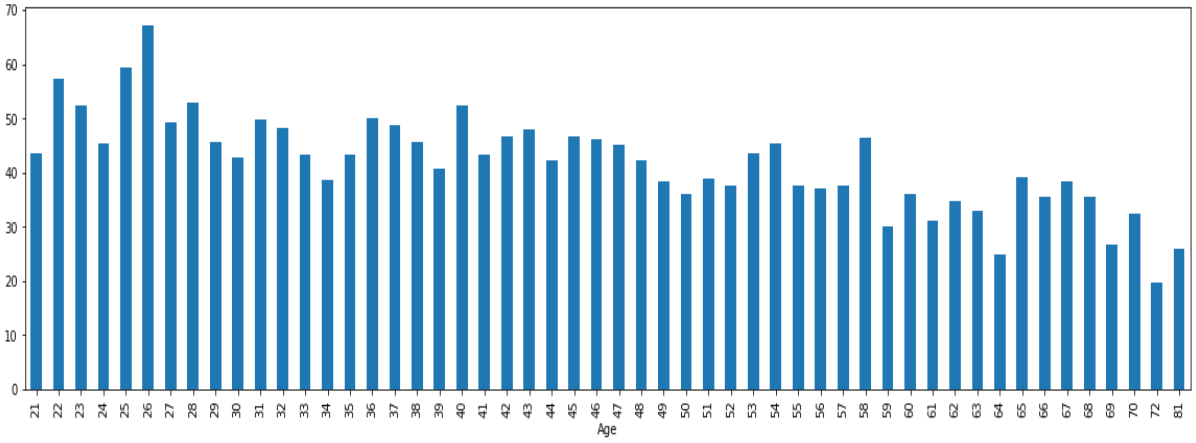


iii. Histogram to depict frequency of number of pregnancies



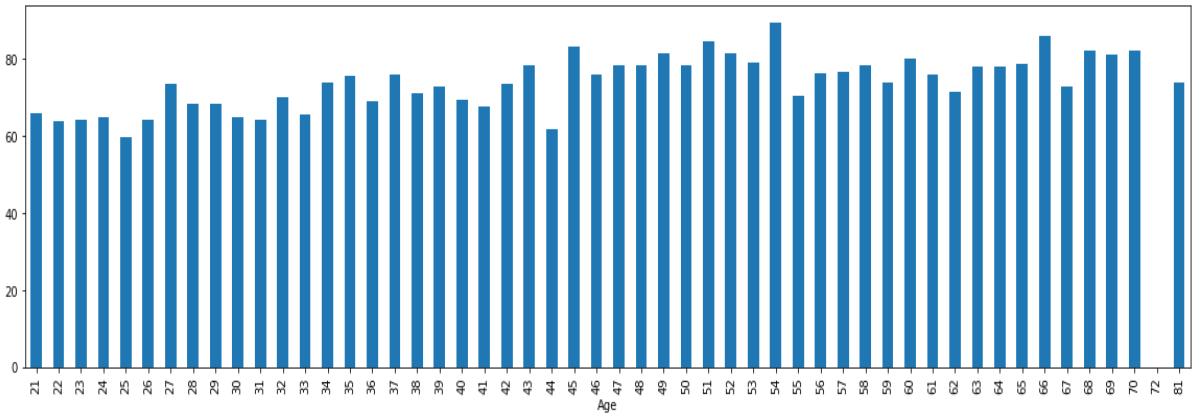
iv. Age – BMI plot

BMI in the younger age women is more than in the older age



v. Age – Blood Pressure plot

Average blood pressure is slightly higher in the older age women compared to the younger age



vi. Check for missing values, maximum and minimum values for each feature

```
1 #To check presence of null values
2 features_data.isna().sum()
```

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

```
#To check max value of the features
features_data.max()
```

```
BloodPressure              122.00
Glucose                    199.00
Insulin                    846.00
SkinThickness               99.00
Pregnancies                 17.00
BMI                         67.10
DiabetesPedigreeFunction     2.42
Age                         81.00
dtype: float64
```

```
#To check min value of the features
features_data.min()
```

```
BloodPressure               0.000
Glucose                     0.000
Insulin                     0.000
SkinThickness               0.000
Pregnancies                 0.000
BMI                         0.000
DiabetesPedigreeFunction    0.078
Age                        21.000
dtype: float64
```

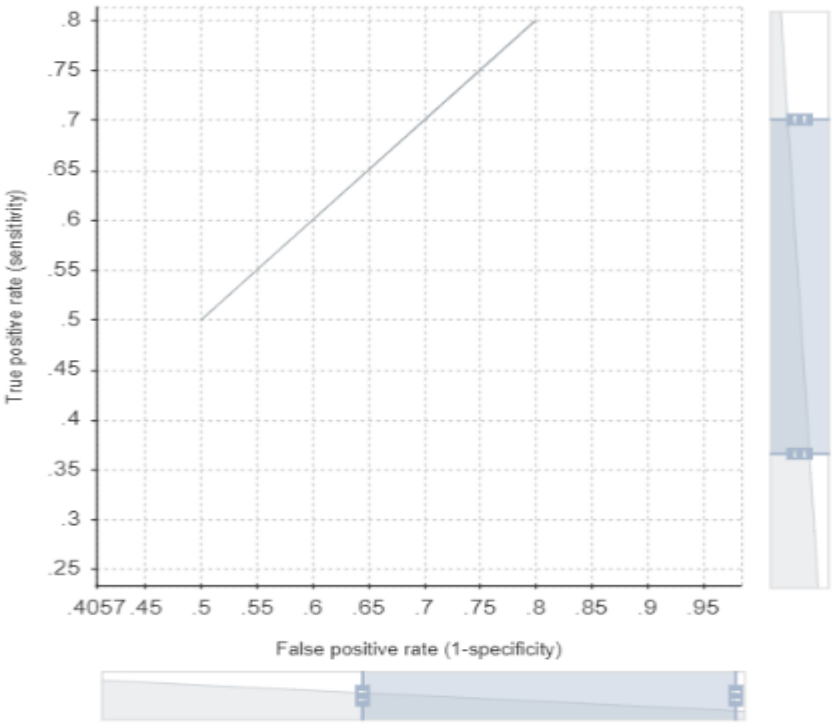
6. Steps followed to build the project

a) Create a project in Watson Studio – DiabetesPrediction
b) Add Auto AI experiment
c) Create a Machine Learning instance
d) Associate ML instance to the project
e) Load the dataset to cloud object storage
f) Select the target variable (prediction parameter) in the dataset
g) Train the model
h) Deploy
i) Build web application using Node-Red

7. Auto AI Experiment Results

XGBoost Classifier is selected by the Auto AI experiment as the best performing model after fine tuning all the hyper-parameters. It is found to give about 78% accuracy with 90% training set size and 10% test set size. The Area Under the Curve (AUC) is also satisfactory which depicts the TPR (sensitivity) and FPR (specificity). The models having higher AUC are said to perform better.

i. ROC Curve:



ii. Model Evaluation Measures

Model Evaluation Measures

	Holdout Score	Cross Validation Score
Accuracy	0.779	0.770
Area Under ROC Curve	0.836	0.811
Precision	0.708	0.665
Recall	0.630	0.681
F_1 Measure	0.667	0.673
Average Precision	0.789	0.695
Log Loss	0.478	0.523

8. Node Red Flow

Node-RED is a flow-based development tool for visual programming developed originally by IBM for wiring together hardware devices, APIs and online services. The flows created in Node-RED are stored using JSON.

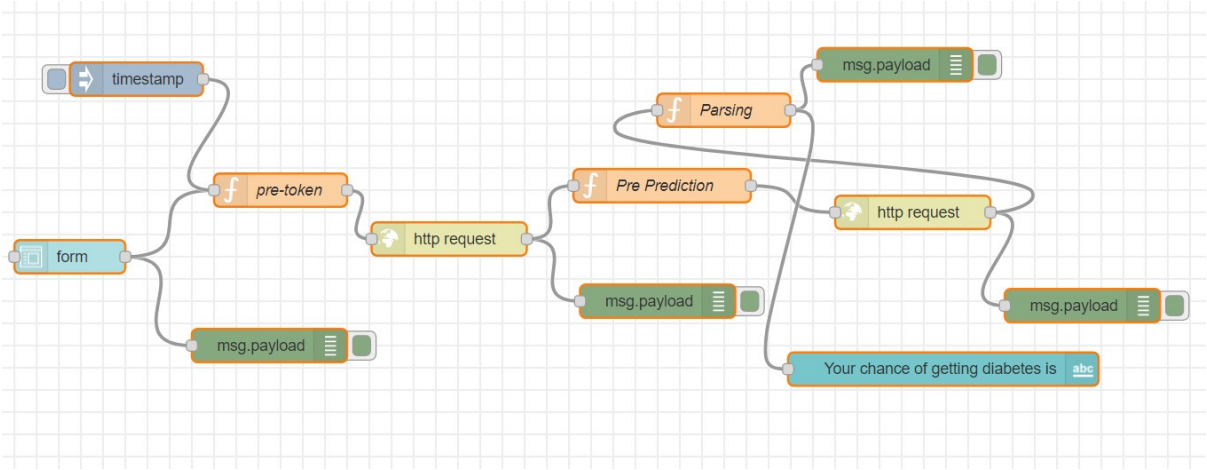


Fig. 8.1 Node-Red Flow

9. Demonstration of the application with the screenshots

This application predicts the chance of acquiring diabetes based on the features/information the user enters through the user interface.

i. Home page of the application

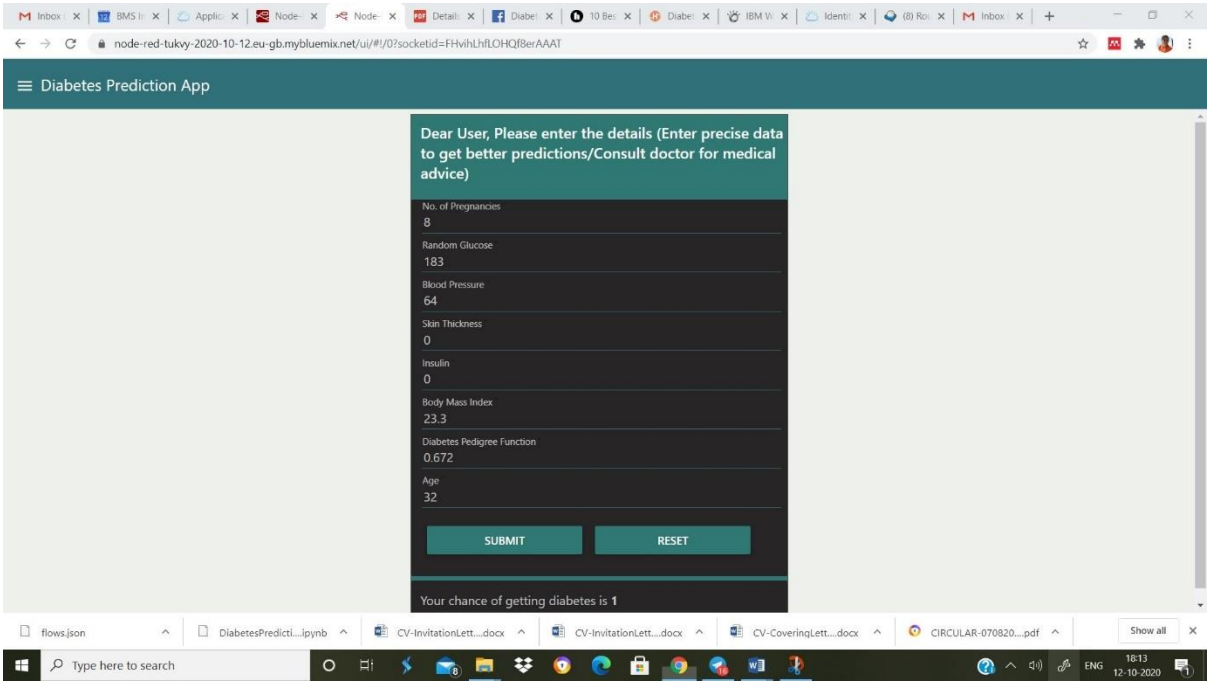


Fig. 9.1 Home page

ii. Reset button can be pressed to clear the previous input and enter fresh details

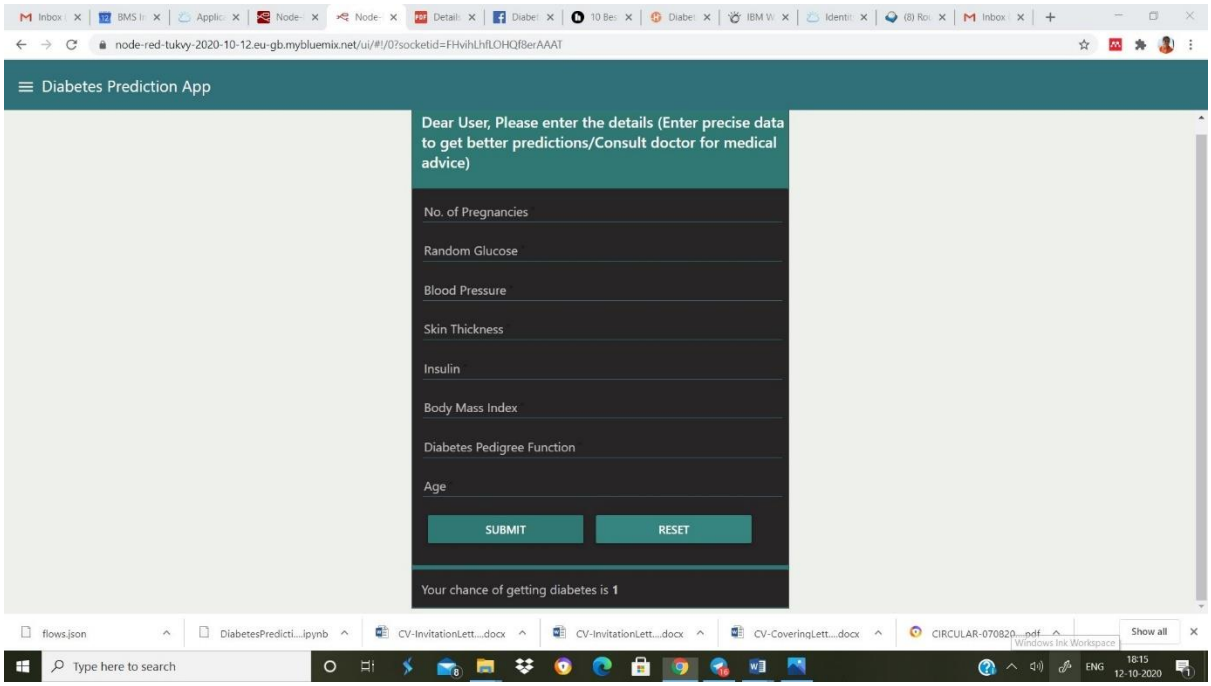


Fig. 9.2 Reset button usage

iii. Fresh details entry and submit button

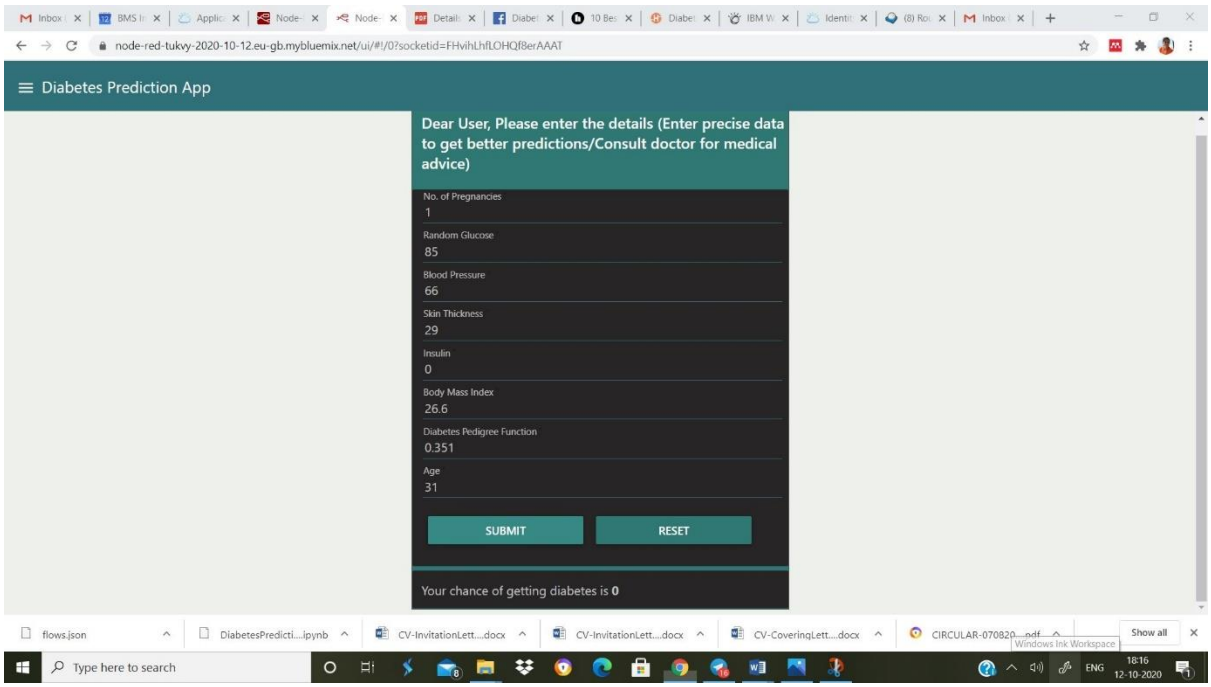


Fig. 9.3 Submit button usage

iv. Prediction display

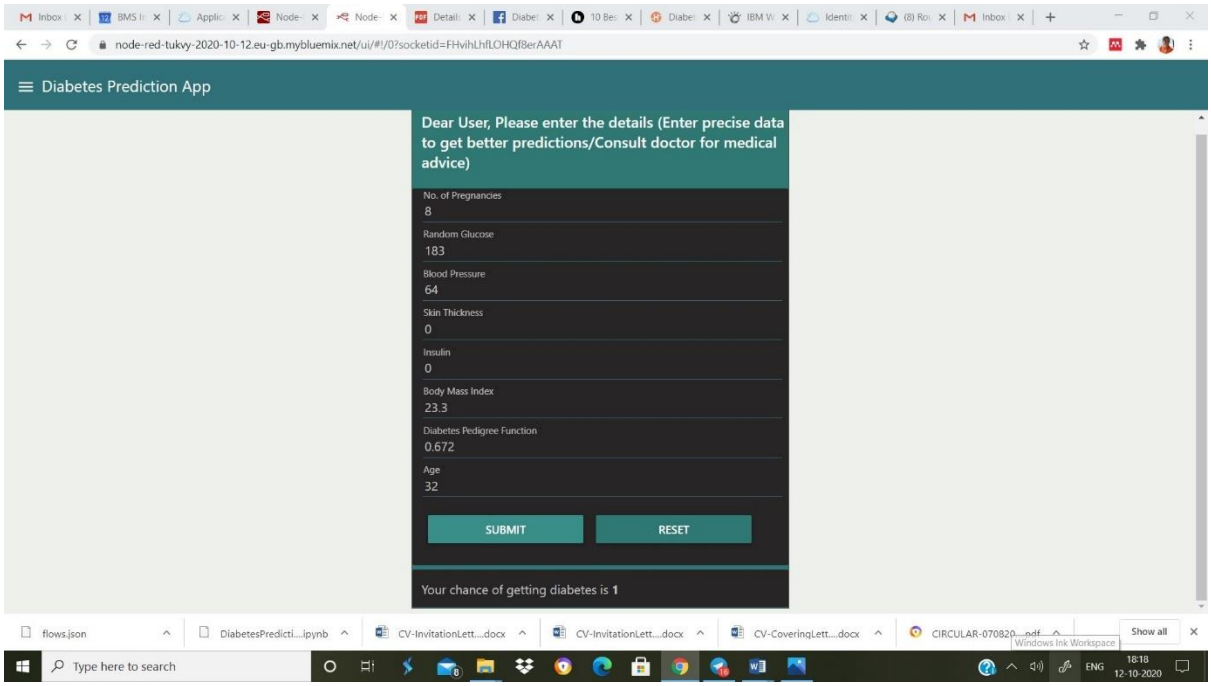


Fig. 9.4 Prediction display as 1

v. Additional features available with the application

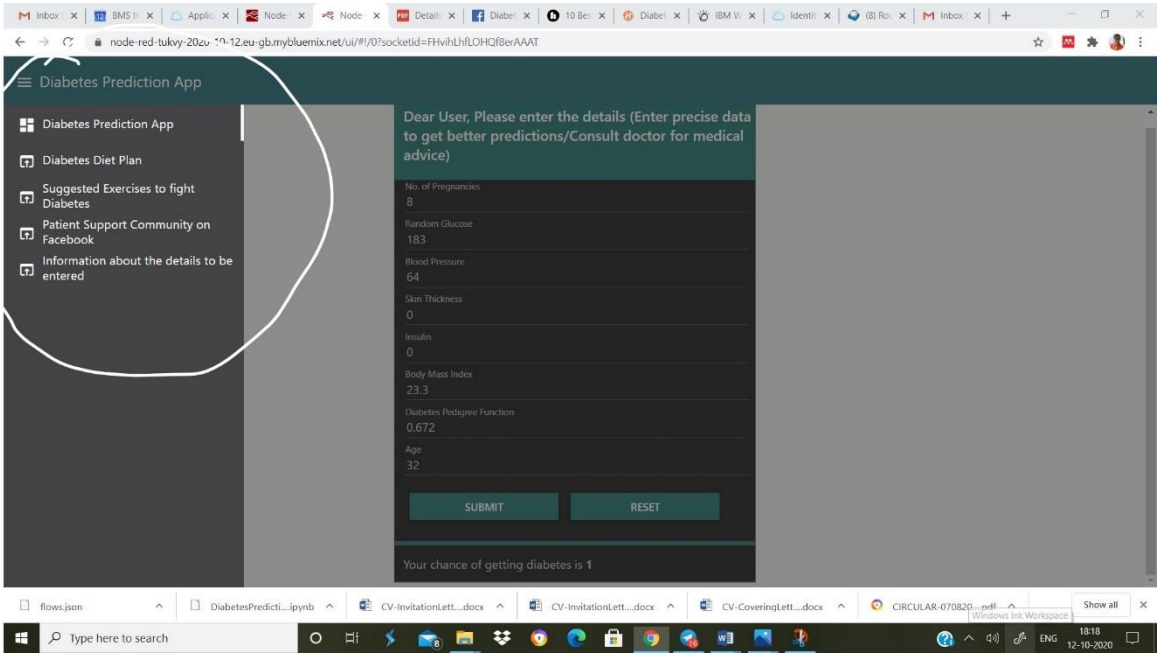


Fig. 9.5 Additional Features

- **Diabetes diet plan link (Source: Healthifyme.com)**

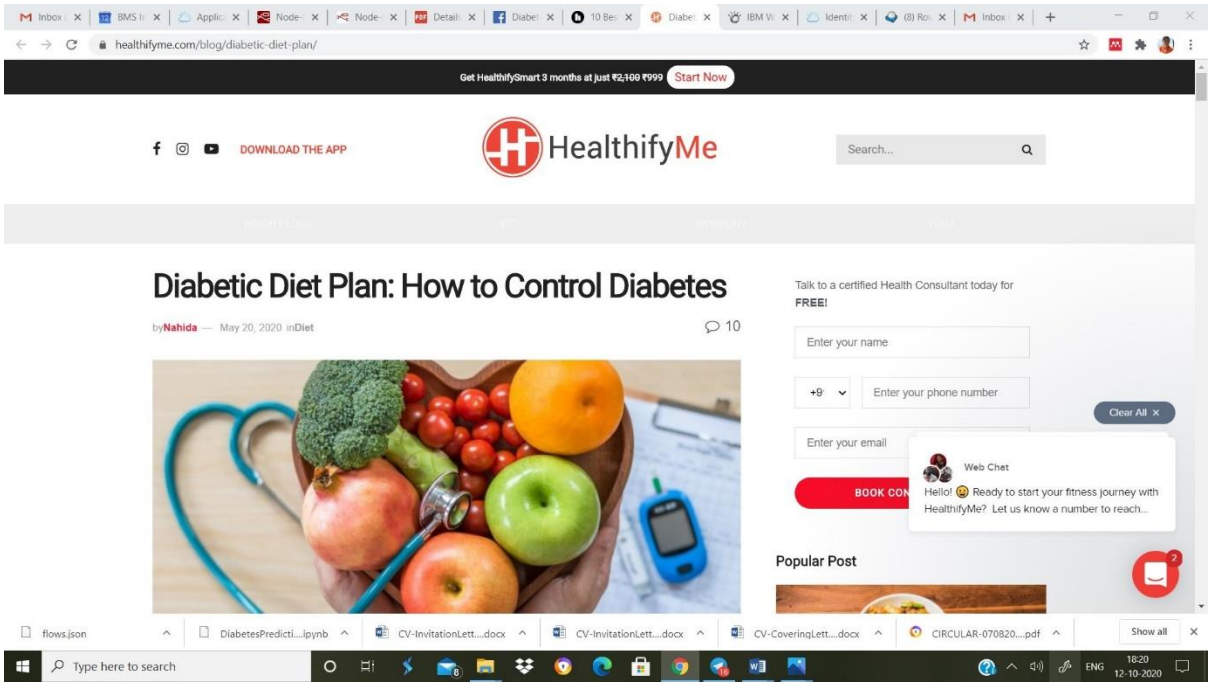


Fig. 9.6 Diabetes diet plan

- **Suggested exercises to fight diabetes link (Source: Healthline.com)**

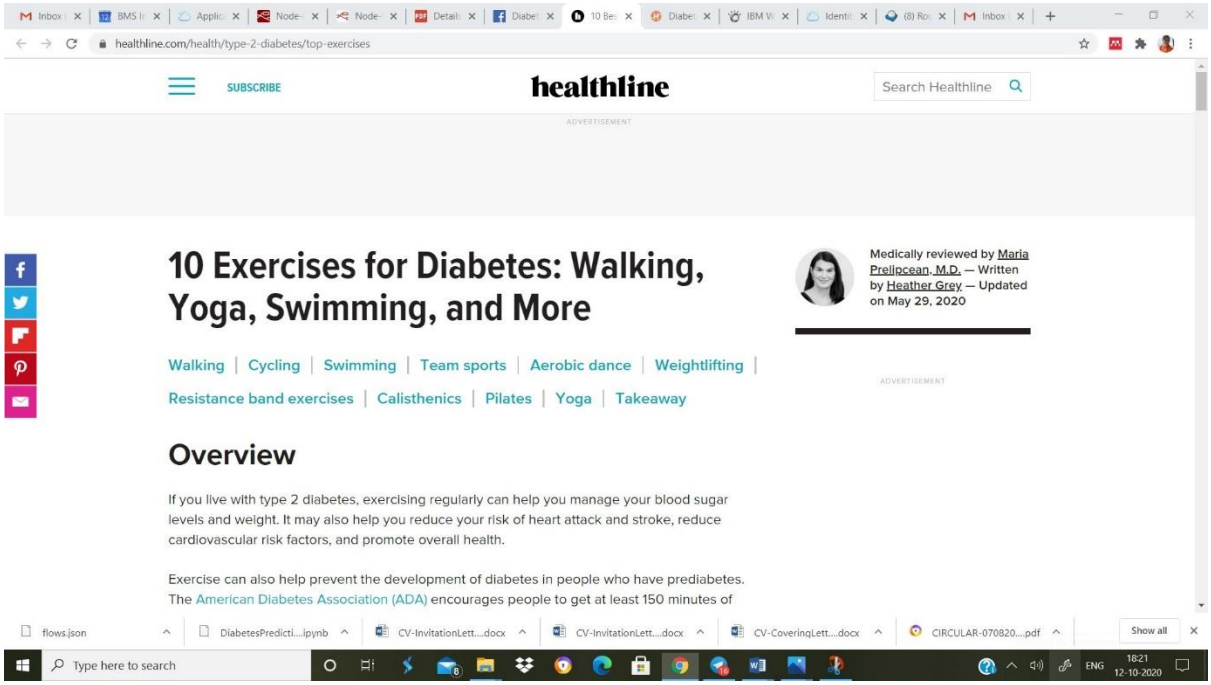


Fig. 9.7 Exercises to fight diabetes

- **Patient support community on Facebook link (Source: Facebook.com)**

Fig. 9.8 Diabetes community on Facebook

- **More information about the details to be entered**

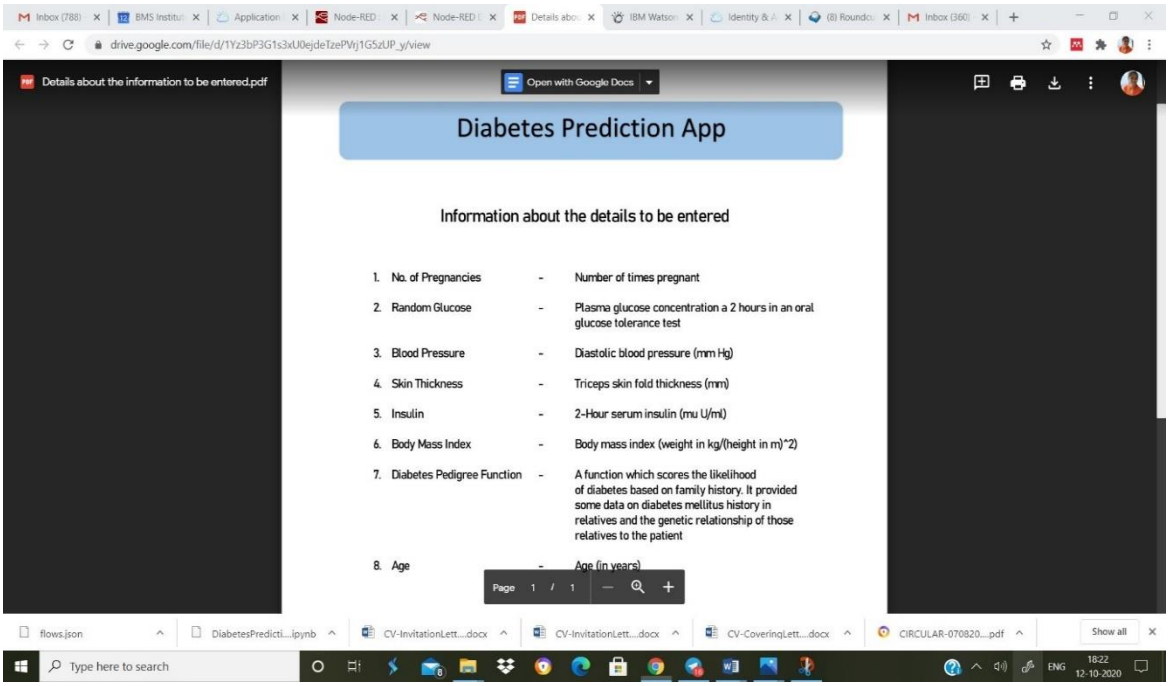


Fig. 9.9 More information on the details to be entered

10. Conclusion and Future Work

The application developed is useful to get predictions about chances of acquiring diabetes based on certain features. Further enhancements can be done w.r.t. user interface to make it more attractive. The auto AI model can be trained on both male and female diabetes patients' data to make the application unbiased.

Acknowledgement

I thank all the faculty members, IBM Team and mentor Mr. Hemanth Kumar who provided me with the knowledge to complete the project. I once again quote the continuous and constructive support given by the mentors in completing the project. Thank you everyone.

Change is the end result of all true learning.

— Leo Buscaglia

Learning is not attained by chance, it must be sought for with ardour and attended to with diligence.

— Abigail Adams