AI Algorithm System for Prediction of Diabetes Using Progressive Web Appand IBM Cloud

Dr. Mohammed Abdul Raheem[1], Shaik Ehetesham[2], Mohammad Faiz Ahmed Subhani[3], Sayed Abdul Zakir[4]

[1] Professor, Department of Electronics and Communication Engineering, Muffakham Jah College of Engineering and Technology, Osmania University, Hyderabad, India

[2,3,4] Student of B.E.(ECE), Department of Electronics and Communication Engineering, Muffakham Jah College of Engineering and Technology, Osmania University, Hyderabad, India

[1]abdulraheem[at]mjcollege.ac.in
[2]160417735098[at]mjcollege.ac.in,
[3]160417735092[at]mjcollege.ac.in
[4]160417735090[at]mjcollege.ac.in

Abstract: ***In the recent times, the most common disease that the world is confronting is Diabetes. The insulin is a hormone which produces glucose in the body. When this insulin is not produced or utilized efficiently, it's resulted into Diabetes. It is estimated that more than 77 million people in India have diabetes as of 2019, and it is predicted that this number can cross 125 million mark by 2040. However, 1 in 5 of those affected doesn't know that they have it. Although there are many ways to detect this disease, using the powerful Artificial Intelligence algorithms can revolutionize the way this disease is detected in early stages. Further, this helps in avoiding associated health issues like heart diseases, neuropathy, nephropathy etc., The research is aimed at designing a model that can predict and detect diabetes in the early stages, and help in general well-being of the subjects. The researchers have employed four Machine learning algorithms like Logistic Regression, KNN Classification, Random Forest Classification and Support Vector machine algorithms to detect Diabetes. In order to maintain the record of the diabetic patients and to notify them their diabetic levels, Artificial Neural Networks algorithms is used. To reach out to common people and to detect diabetes in the early stage, a PWA (Progressive Web App) is built using IBM Cloud and Flask microframework. The researchers believe that this method of diagnosing and informing vulnerable subjects could minimize the diabetic incidence around the world.***

Keywords: Diabetes, Machine Learning, IBM Cloud, Artificial Intelligence algorithms, Artificial Neural Networks, Progressive Web App, Flask, Heroku

1. Introduction

The healthcare sector has long been a quick adopter of cutting-edge technologies like Machine learning and Artificial Intelligence. "Machine learning is the science (and art) of programming computers so they can learn from data," writes Aurelian Geron [6].

The main objective of this research is to develop a Machine Learning models on the historical data and then build a Progressive Web app which can be used by anyone in this world. This app can help the subjects to detect the Diabetes in the early stage. This is achieved through a pipeline in which firstly we collected the real time data from different age groups.

2. Literature Review

Parvin Soleymani [1] from Ryerson University, Canada, "conducted an experimental study using three Machine Learning Classifiers namely Naive Bayes, Logistic Regression and Decision tree to predict the likeliness of diabetes. These models were used to compare their performance in terms of accuracy, precision, recall and ROC Score. The final result of this experiment showed that the Logistic Regression classifier plays the best performance in this prediction of diabetes with a highest accuracy of 78% in comparison to the other models".

Abdullah A.et al [2], of King Saud University (KSA); used data mining and machine learning tools to analyse the trend of Diabetes among various age groups. "Their aim was to predict the type of diabetes and also suggest the appropriate treatment relevant to the case based more significantly on the age of the patient. They were focused on six types of treatments that were identified in the 2005 World Health Organization's NCD report of Ministry of Health, Saudi Arabia. The treatments included: Drug, Diet, Weight reduction, Smoke cessation, Exercise, Insulin".

Mukesh Kumari et al [3] from P.D.M College of Engineering, Bahadurgarh,worked with the "WEKA tool for the prediction of diabetes and their research was reviewed by the International Conference on Computational Intelligence and Data Science (ICCIDS 2018). WEKA is a software which is designed in the country New Zealand by University of Waikato, which includes a collection of various machine learning methods for data classification, clustering, regression, visualization etc. One of the biggest advantages of using WEKA is that it can be personalized according to the requirements. This proposed project was evaluated on Diabetes Dataset namely (PIDD), which is taken from UCI Repository".

Gauri D. Kalyankar et al[4]used HADOOP along with other machine learning algorithms to predict diabetes. "They implemented Hadoop MapReduce based machine learning algorithms for Pima Indian diabetes data set to find out missing values in it and to discover patterns from it. Their work suggested that implemented algorithms were able to impute missing values and to recognize patterns from the data set. Further pattern matching was employed by applying the discovered patterns on testing data set to predict diabetic prevalent and risk levels associated with it".

S.R. Surya [5] a professor from SRM University, performed a research on Predicting diabetes at an early stage. He used predictive analytic methods of Big data to predict the diabetes. "Due to the unstructured nature of Big Data form health industry, it is necessary to structure and emphasis their size into nominal value with possible solution. Healthcare industry faces many challenges that make us to know the importance to develop the data analytics of the diabetes mellitus. So he implemented SVM and KNN algorithms and has obtained an accuracy of 92.34% and 86.5% respectively".

3. Methodology

The main objective of this research is to predict diabetes and provide assistance to the patients, firstly, the researchers identified the problem statement and converted it into a business model so that it can provide a possible solution (Fig-1). Secondly, they conducted the field work relating to Diabetes and built an appropriate model. After building the model, the researchers integrated the Machine Learning models with the IBMCloud in the deployment/QA stage using Flask and Docker. Then the deployment package was tested in the dry runphase. Finally, the Progressive Web App was created as the end product.

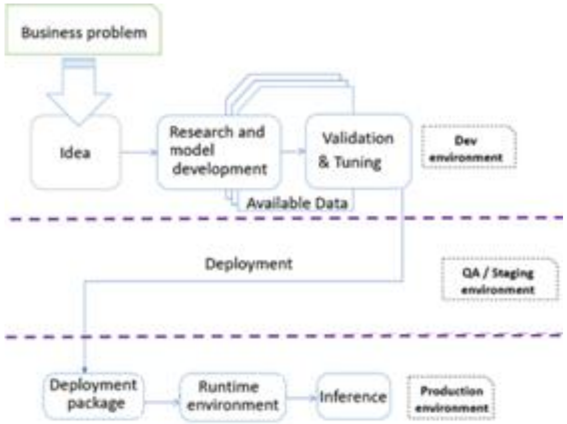


Figure 1

This project can be divided into two scenarios:
1) A normal person
2) A Diabetic person

In scenario 1(Normal person) we aim to determine if the person is diabetic or not. So, by the term 'Normal Person' it is intended to convey that the person in this scenario is unaware if he has the disease or not. Further if the person has Diabetes, then our model can also predict whether it's Type 1/ Type 2 diabetes based on the input data provided by the user. Since it is a "yes/no" or "true/false" question that we are trying get an answer to, we will use Classification algorithms.

In scenario 2 (Diabetic person); the person is already aware that he is diabetic, here we are aiming to identify what kind of diabetes that person is suffering from and we also intend to create a sort of a reminder system, which is efficient enough to predict any sort of adverse effect of Diabetes in advance using the historical data of the patient.

The above two scenarios are illustrated in the below fig-2. As said above, for a normal person, to detect the Diabetes, the classification algorithms are used and to maintain the track record and then to send the notification, the Neural Networks are used.

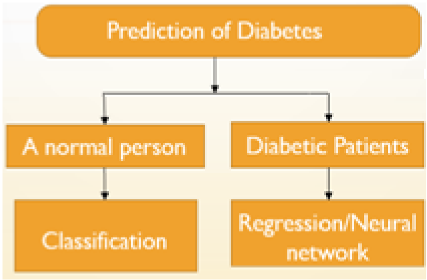


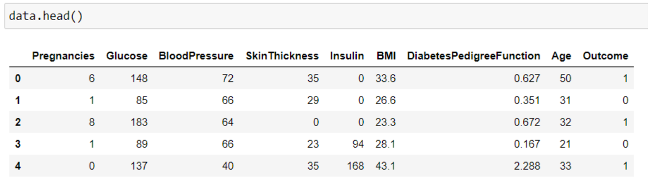
Figure 2

4. Algorithms Used

a) Dataset

The real time data has been collected from different age groups and then the data points are colligated into the dataset. The dimensions used in our dataset are:

- Gender
- Pregnancies
- BloodPressure
- Urination_frequency
- BMI
- Hereditary
- Age
- Outcome
- To fetch the dataset, we need to use: Index(['Gender','Pregnancies','BloodPressure','Urination_frequency','BMI','Hereditary''Age' 'Outcome'], dtype='object') as shown in Fig-3.



```
data.head()
```

	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

Figure 3

The diabetes data set consists of 6903data points, with 9 features. The dimension of diabetes data: (767, 9). "Outcome" is the output feature. If it's 0, it means that 'No diabetes', and if it's 1, that means 'diabetes'. Then it's converted into the percentage. Of these 767 data points, 499 are labelled as 0 and 268 as 1.

b) Visualization of the dataset

The visualization of the data is an important factor in the process of selecting the accurate model for the dataset.

So, to understand the dataset, Matplotlib and Seaborn are used for low-level and the high-level visualization.

In fig-4, the graph is plotted between Age and Blood Pressure. From this seaborn plot, we can observe that in the age group of 40-60, Blood Pressure is high. This age group is more prone to be affected by Diabetes.

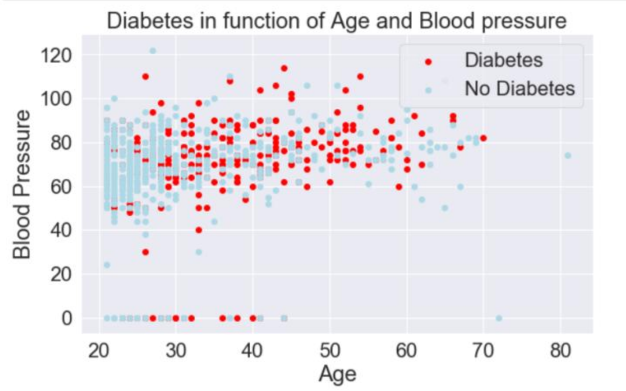


Figure 4

By using the Matplotlib library, bar graph is plotted for all the dimensions in the dataset. From this plot, we can understand the data points clearly so that we can implement various algorithms (fig-5).

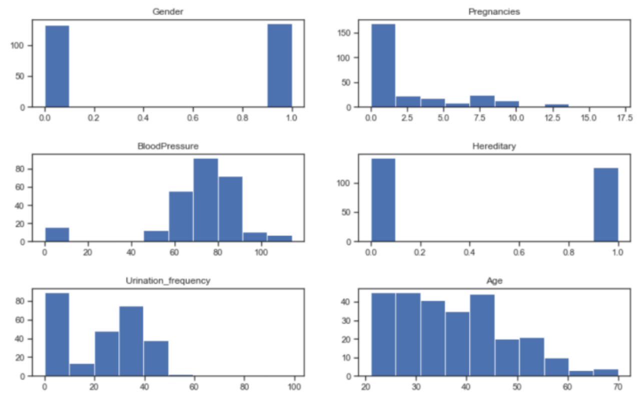


Figure 5

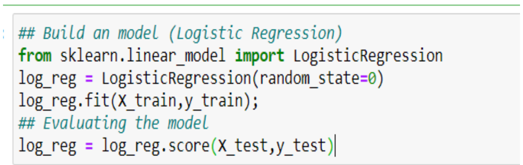
c) Algorithms Used in this project

Supervised Machine Learning: The Supervised Machine is a part of the Machine learning, where in the model is trained by the different features in the dataset [9]. Supervised Machine learning is divided into Classification and Regression and algorithms.

Classification
A classification algorithm is used when the output variable is binary, such as “diabetes” or “no diabetes”. We’ve used numerical data to classify and predict whether the person has diabetes or not[8]. Classification algorithms used:

d) Logistic Regression

Logistic regression [7] is a powerful classification algorithm which is well known for classifying the categorical or the numerical data using the function ‘ϕ’.



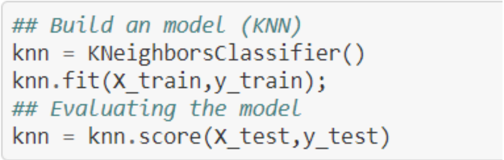
```
## Build an model (Logistic Regression)
from sklearn.linear_model import LogisticRegression
log_reg = LogisticRegression(random_state=0)
log_reg.fit(X_train,y_train);
## Evaluating the model
log_reg = log_reg.score(X_test,y_test)
```

Figure 6

As shown in the above figure (Figure-) From ‘sklearn.linear_model’, importingLogisticRegression and then fitting the dataset into the model and finally evaluating the model.

e) KNN Classification

K- Nearest Neighbour [8] classification is used to classify the dataset based on the nearest neighbours. The test data point is classified into ‘Diabetes’ or ‘No Diabetes’ based on the distance calculated between this test data point and the other two sets.

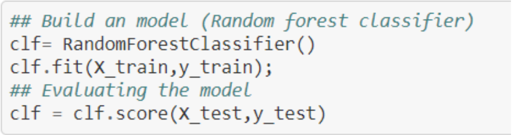


```
## Build an model (KNN)
knn = KNeighborsClassifier()
knn.fit(X_train,y_train);
## Evaluating the model
knn = knn.score(X_test,y_test)
```

Figure 7

f) Random Forest Classification

Random forest classification is an algorithm which uses various decision trees to classify the given problem statement. The RandomForestClassifier is imported from the scikit learn library as shown in below fig-8.

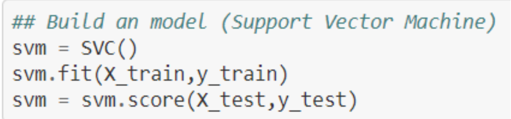


```
## Build an model (Random forest classifier)
clf= RandomForestClassifier()
clf.fit(X_train,y_train);
## Evaluating the model
clf = clf.score(X_test,y_test)
```

Figure 8

g) SVM Classification

The Support Vector machine is a classification algorithm which classifies the data based on the hyperplanes in the dataset. As shown in the below snapshot(fig-9), the SVC() is implemented.

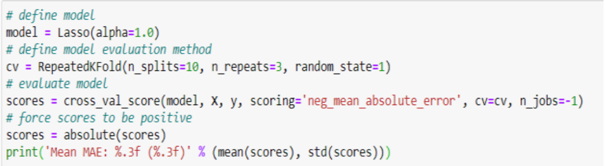


```
## Build an model (Support Vector Machine)
svm = SVC()
svm.fit(X_train,y_train)
svm = svm.score(X_test,y_test)
```

Figure 9

h) Lasso Regression

L1 Regularization is used in the Lasso regression algorithm [10]. In this regularization, the absolute value is added to the coefficients so that the error rate is decreased as shown below in Fig-10

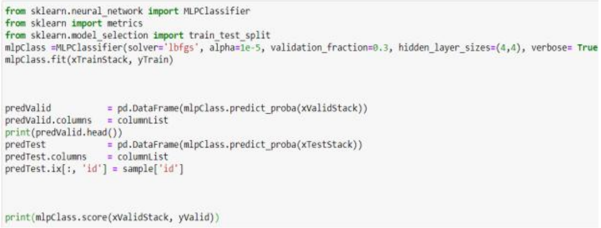


```
# define model
model = Lasso(alpha=1.0)
# define model evaluation method
cv = RepeatedKFold(n_splits=10, n_repeats=3, random_state=1)
# evaluate model
scores = cross_val_score(model, X, y, scoring='neg_mean_absolute_error', cv=cv, n_jobs=-1)
# force scores to be positive
scores = absolute(scores)
print('Mean MAE: %.3f (%.3f)' % (mean(scores), std(scores)))
```

Figure 10

i) Multi-Layer Perceptron:

Multi-Layer Perceptron can be used for univariate time series prediction [11]. For instance, if the diabetic levels of a patient go high, then the Multi Linear perceptron algorithm can be used to detect the anomaly and then send the notification.



```
from sklearn.neural_network import MLPClassifier
from sklearn import metrics
from sklearn.model_selection import train_test_split
mlpClass =MLPClassifier(solver='lbfgs', alpha=1e-5, validation_fraction=0.3, hidden_layer_sizes=(4,4), verbose= True
mlpClass.fit(xTrainStack, yTrain)


predValid          = pd.DataFrame(mlpClass.predict_proba(xValidStack))
predValid.columns  = columnList
print(predValid.head())
predTest           = pd.DataFrame(mlpClass.predict_proba(xTestStack))
predTest.columns   = columnList
predTest.ix[:, 'id'] = sample['id']


print(mlpClass.score(xValidStack, yValid))
```

Figure 11

j) Deployment of ML models on IBM cloud

After building the Machine Learning models, the next phase ids to deploy these models on the IBM cloud. In the IBM Cloud, the IBM Watson Studio is launched and then a Machine Learning instance is created. Then the dataset is uploaded in the assets section. After uploading the dataset, the pre-built models are deployed by choosing the Auto AI feature. This will select the models which has highest accuracy automatically, from the models file(.ipynb) which is given as an input with the dataset.

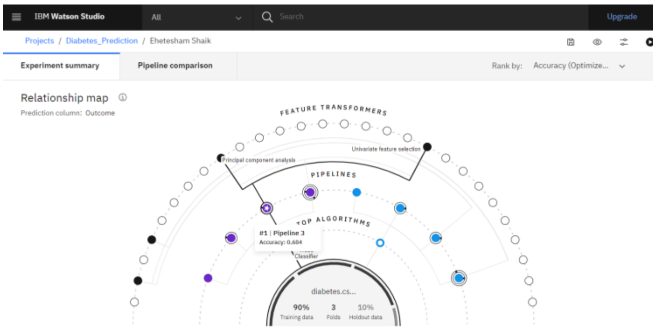
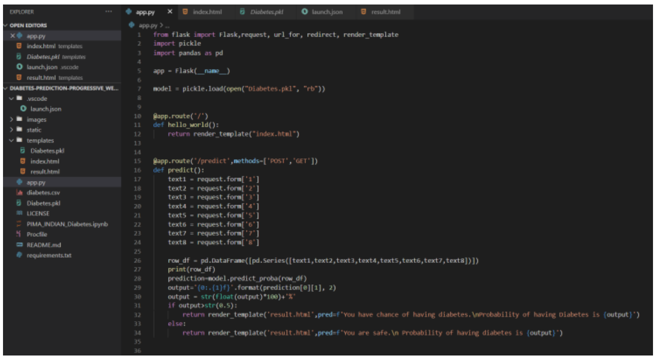


Figure 12

k) Building the Progressive Web App

Finally, the website is built using HTML, CSS and JavaScript. Flask is a microframework which has been used for the backend development. After deploying the models on IBM Cloud, a JavaScript code is generated which is embedded into the Website code. Then the code is tested on Dry run which is on local host after creating the virtual environment.



```
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}')
```

Figure 13

As shown in Fig-13, the app.py is a Flask app which is a common ground between the frontend and backend files. Here the pickle file is used which is a compressed file of the machine learning models which have been deployed in this project.

The chatbot is also integrated in the index.html page, which has been created using 'Google Dialogflow' for the assistance of the user to calculate BMI, or to give them a few tips to reduce the Diabetes.

5. Results and Discussion

Based on this dataset, the Machine Learning algorithms used and the accuracy of these models are shown in Table-1. Since the accuracy of Logistic Regression is high in the classification algorithms, we've implemented this algorithm in our project for detecting Diabetes.

If we consider the second scenario, to maintain the record of diabetes patients, we've implemented the Multi-Layer Perceptron as it has the highest accuracy.

Table 1: Accuracies obtained by algorithms

Type of Algorithm	Algorithm	Accuracy
Classification Algorithms	Logistic Regression	82%
	KNN Classification	75%
	Random Forest Classification	80%
	SVM Classification	78%
Artificial Neural Networks	Lasso Regression	80%
	Multi Layer Perceptron	86%

After implementing the models, they are deployed on IBMCloud with the help of Flask and Docker. As shown in below fig-14, all the models ran through different accuracy metrics like AUC, F1, Log loss and Precision which can determine the accuracy.

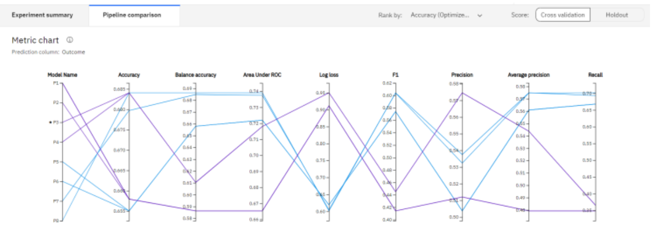


Figure 14

Finally, the Progressive Web app is created. The user needs to open the app "DIABETICO, Your Diabetes monitor". Then after signing up in the app, the user can enter the values in the respective fields as shown in Fig-15.

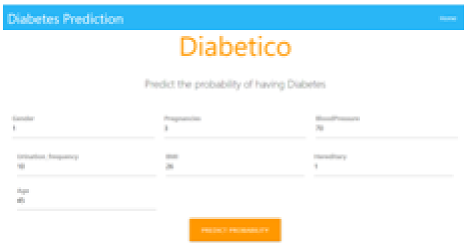
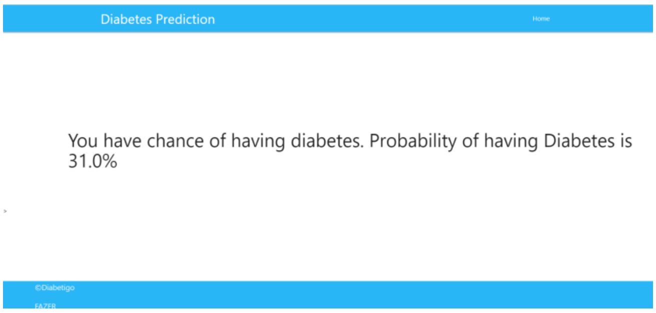


Figure 15

After clicking on the ‘Predict Probability’, it will be redirected to the results page and the % of probability of having diabetes is displayed in the screen as shown in Fig-16.

Basically, whenever the user enters the values, they will be appended in the IBM Cloud, and then if the Diabetic levels increase, then the user gets a push notification in the mobile.



You have chance of having diabetes. Probability of having Diabetes is 31.0%

Figure 16

6. Conclusion & Future Scope

It’s very important to detect Diabetes in the early stage. Although the accuracy achieved by these Machine Learning models are high, there are a few limitations in this project.

Firstly, the number of parameters can be increased, considering that Diabetes is a very complex disease and a limited number of parameters might not be sufficient enough to predict the disease accurately. For example, a few features like hereditary, Gestational diabetes etc. are not available in the dataset, so in future, these features can be added to predict the Diabetes more precisely.

Secondly, the app which was built in this study can be improved further by adding new features like automatic location detection of the user to conveniently suggest the patient to the nearest diagnostic centers.

References

[1] Parvin Soleymani,. Prediction of Diabetes, Ryerson University- Computer Science Department, Canada., 2020.

[2] Al Juma, A.L., Ahmad, M.G., Siddiqui, M.K., 2013. Application of data mining: Diabetes health care in young and old patients. Journal of King Saud University - Computer and Information Sciences 25, 127–136. doi:10.1016/j.jksuci.2012.10.003.

[3] Mukesh kumari et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (4) , 2014, 5174-5178, “Prediction of Diabetes Using Bayesian Network”.

[4] G. D. Kalyankar, S. R. Poojara and N. V. Dharwadkar, "Predictive analysis of diabetic patient data using machine learning and Hadoop," 2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), Palladam, 2017, pp. 619-624, doi: 10.1109/I-SMAC.2017.8058253.

[5] S.R.Surya, Assistant Professor, Faculty of science and Humanities SRM Institute of Science and Technology, “LITERATURE SURVEY ON DIABETES MELLITUS USING PREDICTIVE ANALYTICS OF BIG DATA”, *International Journal of Advance Engineering and Research Development Volume 5, Issue 02, February -2018.*

[6] Deepti Sisodia , Dilip Singh Sisodia, “Prediction of Diabetes using Classification Algorithms”, *International Conference on Computational Intelligence and Data Science (ICCIDS 2018)*.

[7] Beyer, W. H. CRC Standard Mathematical Tables, 31st ed. Boca Raton, FL: CRC Press, pp. 536 and 571, 2002.

[8] Agresti A. (1990) Categorical Data Analysis. John Wiley and Sons, New York.

[9] Kotz, S.; et al., eds. (2006), Encyclopedia of Statistical Sciences, Wiley.

[10] Wheelan, C. (2014). Naked Statistics. W. W. Norton & Company.

[11] Sheetal Sharma(2017) Artificial Neural Network (ANN) in Machine Learning, Data Science Central.