Volume 10, Issue 3, July 2023 ISSN: 2348-8565 (Online)
International Journal of Modern Engineering and Research Technology
Website: http://www.ijmert.org Email: editor.ijmert@gmail.com

Diabetes Mellitus Prediction Using Machine Learning Technique

Shekhar Singh Baghel
Research Scholar M.Tech.
Computer Science and Engineering
Shri Ram Group of Institutions,
Jabalpur, (M.P.) India
Email: shekharsingh2602@gmail.com

Sapna Choudhary
Associate Professor
Department of Computer Science and Engineering
Shri Ram Group of Institutions,
Jabalpur, (M.P.) India
Email: choudharysapnajain@gmail.com

Anupam Choudhary
Lecturer,
Government Kalaniketan Polytechnic College
Jabalpur (M.P.), India
Email:chowdharyanupam7@yahoo.com

ABSTRACT

According to the increasing morbidity of 2040, in the last few years, the world's diabetic patients will reach 642 million, which means one out of 10 adults will suffer from diabetes in the future. There is no doubt that great attention is needed in this alarming figure. Machine learning has been applied to many aspects of medical health with the rapid development of machine learning. In order to determine whether the subject is diagnosed with juvenile diabetes, a series of tests carried out immediately before diagnosis were used. A modified set of training settings consisting of differences between test results at various times was also used to establish classifiers to predict if juvenile diabetes was diagnosed. Supervised were compared to decision-making trees and both types of classifiers were not supervised. In this study, a diagnosis based on the pre-test probability calculated from patient information, including symptoms of previous tests, is most likely confirmed by the system and test. If the probability of the post-test disease of the patient is higher than the threshold, a diagnostic decision will be taken and vice versa. If not, the patient will need additional tests to make a decision. Then the system recommends the next optimal test and repeats the same process. In this thesis, find out what approach is better in the proposed framework for diabetes data. Use feature selection techniques to reduce process characteristics and complexities. The aim of this research is the development of a system that can predict diabetes early for a patient with greater accuracy by combining the findings of various methods of machine learning. This investigation aims to predict diabetes through three different methods, including: support of vector machines (SVM), logistic regression, random forest classification and selection of features. The objective of this project is also to propose an effective technique for early diabetes detection.

Keywords:— *Machine learning, Diabetes Prediction, probability, Feature Selection, Classification, SVM, Logistic regression, Random Forest Classifier*

I. INTRODUCTION

Diabetes is one of the world's deadliest diseases. This disease is not only a disease, it is also a source of various diseases such as heart attacks, blindness, kidney diseases etc.[1]. The normal process of identification

is for patients to visit a diagnostic centre, consult their doctors and sit down for one day or more to get their reports. Moreover, they must waste their money in vain every time they want their diagnosis report. Diabetes mellitus (DM) is defined as a group of metabolic disorders caused primarily by abnormal secretion of insulin and/or action. Insulin failure results in high blood glucose (hyperglycemia) and impaired carbohydrate, fat, and protein metabolism. DM is one of the most common endocrine disorders in the world that affects over 200 million people.

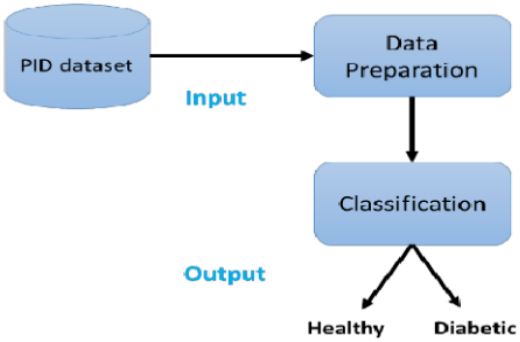

Figure 1: Diabetes prediction classification models [1]

II. Machine Learning

Machine learning is the scientific field that deals with how machines learn from experience. The term "machine learning" is identical for many researchers to that of "artificial intelligence" because learning is the main characteristic of an entity that is called intelligent in the broadest sense of the word. The aim of machine learning is to build computer systems capable of adapting and learning from their experience[8]. Mitchel gives a more detailed and formal definition of machine learning: a computer program should learn from experience E in relation to some class of Tasks and measure P if its performance in T tasks, measured by P, is improved with experience E. Through the development of machine teaching approaches, we have developed a system that uses data mining to predict whether or not the patient has diabetes. In addition, early prediction leads to treatment of patients before it becomes critical. Data mining is capable of extracting hidden knowledge from a large number of data relating to diabetes[14]. This is why it has an important role now more than ever in diabetes research. The aim of this research is to develop a system that can accurately predict a patient's diabetic risk level. This research focuses on the development of a system based on three methods of classification: support of vector machine, logistic regression and algorithms for Artificial Neural Networks.

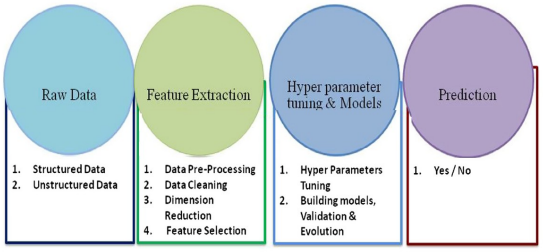

Figure 2. Essential Learning process to develop a predictive model [20]

2.1 Review of Prior Works

Md. In this [19] study, logical regression (LR) is used to identify risk factors for diabetes, based on p-value and odds ratio Maniruzzaman (2020). (OR). In the prediction of diabetic patients, we have taken four classifiers such as naïve Bays (NB), decision tree (DT), adaboost (AB) and random forestry (RF). These protocols were adopted and repeated in 20 trails by three types of partition protocols (K2, K5, and K10). Performance of these classifiers is assessed with accuracy (ACC) and curve area (AUC).

In this study, Lejla Alic (2019) revisits the data from the San Antonio Heart Study[15] and uses data science to estimate prospects for growth of diabetes mellitus. They use the supporting vector machine and ten aspects to develop the projected model,

which are known excellent in the literature as an excellent indication of actual diabetes.

In this article [16] they introduced Minyechil Alehegn, Rahul Joshi and Dr. Preeti Mulay (2019)

The Ensemble Method (PEM) proposed to improve precision. Vector Machine support, Naive Net support,

Decision Stump is used separately and an ensemble method is also developed. It has been found that

The maximum efficiency was demonstrated by PEM.

This analysis[17] was carried out by Amani Yahyaoui, Akhtar Jamil (2019), with a detailed analysis of machine learning and in -depth learning algorithms for detection of diabetes. The results indicated that RF is stronger for classification of diabetes in all rounds of tests, resulting in 83.67 percent total accuracy for diabetes prediction. The predictive accuracy of SVM was 65.38% while the DL method on our data generated 76.81%.

Faisal Faruque (2019): In this [10] paper, authors tried to avoid the side effects of diabetes early on. First of all, to find out this, researchers try to predict different risk factors related to the disease. To find the best choice, four different machine learning algorithms were observed and we note that C4.5 Decision Tree is the best choice for their diabetes prediction.

The following [18] article should identify the important factors for the cause of diabetes: Debadri Dutta, Debpriyo Paul, Parthajeet Ghosh (2018). A lot of research has focused on parameters and feature selection in areas of use in which tens or a large number of variables are available. We will also focus on the most important aspects to predict the chances of a person developing diabetes.

Hang (2018) A model for the relevant disease prediction of the Feed - Forecast Neural Network [3] was proposed. This paper proposed a framework for early detection and hence disease prevention by taking the major risk factors into account. The UCI repository dataset has been used to develop the training algorithm into an ANN framework.

Mirza (2018): This paper uses SMOTE and DT classificator methods to develop a Diabetes Prediction Model[3]. It is a hercules task to classify imbalanced data, particularly in medical computer science. This was an important motivator for the development of an SMOTE classification. Both methods were combined with the aim of improving diabetic prediction's predictive accuracy by eliminating class imbalances. The system proposed consists of two levels. The data imbalance is eliminated in the first stage using SMOTE and then the disease is identified with the DT classification in the next phase.

Dadgar (2017): a combination of feature selection and the neural network method with the diabetes prediction genetic algorithm. This paper proposes a method based on the UTA algorithm and the neural network of two layers. This is revised and combined with genomic weights to improve the classification of diabetes. There are two stages in the development of this process, the selection and estimation of features based on the UTA algorithm. The UCI repository's Pima dataset was used to test this article with 87.46% precision. The approach of this study provided a highly accurate output of diabetes prediction, especially in comparison with other models, they considered a time factor for analysis as well.

3.1 Proposed System

The proposed approach predicts the disease of diabetes in patients with optimum accuracy. We're going to talk about multiple machine learning, an algorithm that can help with prediction and decision-making. We can use more than one algorithm to get better prediction precision.

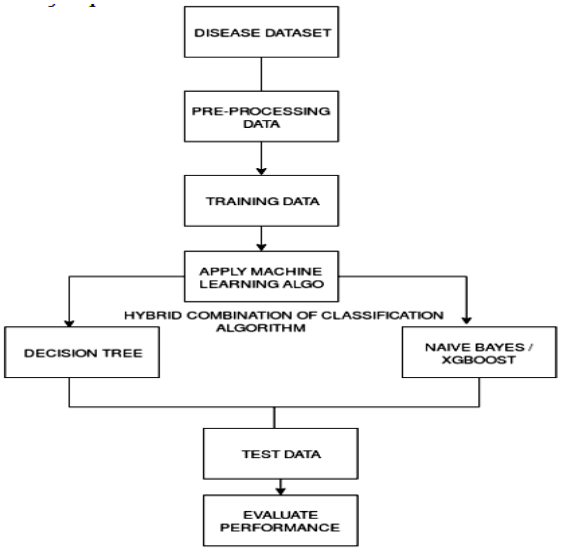

Figure 3: Proposed Diagram

III. IMPLEMENTATION

Implementation steps

In this section, we shall analyse the real steps taken during the m experiment. The step-by-step approach used to determine data for diabetes prediction and to estimate the accuracy of data will be explained. The key steps below are the method:

We chose the PIMA Indian Diabetes Dataset which includes 768 instances in two classes:

Diabetic and non-diabetic with 8 separate risk factors: two-time plasma blocking, diastolic blood pressures, triceps skin fold thickness, two-hour serum insulin concentration, pedigree diabetes function include. Diabetic and non-diabetic diabetes with 8 different blood risk factors.

Feature selection is the process in which we choose the features that are most relevant to your variable prediction or performance, automatically or manually. If our data includes irrelevant attributes, then the accuracy of the models can be decreased.

- We take a dataset of diabetes.
- The framework uses the Feature Selection: Further selection of features and Backward Feature selection for the pre-processing level. Five different classifiers are trained and we determine which classifier offers great precision. We also used ADABoost, Decision Tree, XGBoost, Voting Classifier, Stacking Classifier.
- Random Forest, ADABoost and Logistic Regression are used for Stacking Classifying and the meta specification of XGBoost.
- The best of all five classificatory in the accuracy aspects were found in Adaboost and Stacking Classifier because they provide greater accuracy.
- The following screenshots are used to understand better the flow and desired outcomes of our implementation steps. The ADABoost classification will be shown step by step. For decision tree, XG boost, voting and stacking classificators we have done similar measures.

IV. RESULTS

The Data

The diabetes data set was originated from UCI Machine Learning Repository.

```
import pandas as pd

import numpy as np

import matplotlib.pyplot as plt
```

%matplotlibinlinediabetes = pd.read_csv (‘diabetes.csv’)

print(diabetes.columns)

Index([‘Pregnancies’, ‘Glucose’,

‘BloodPressure’, ‘SkinThickness’, ‘Insulin’,

‘BMI’, ‘DiabetesPedigreeFunction’, ‘Age’,

‘Outcome’], dtype=’object’) diabetes.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

The diabetes data set consists of 768 data points, with 9 features each:

print("dimension of diabetes data: {}".format(diabetes.shape))

dimension of diabetes data: (768, 9)

“Outcome” is the feature we are going to predict, 0 means No diabetes, 1 means diabetes. Of these 768 data points, 500 are labelled as 0 and 268 as 1:

print(diabetes.groupby('Outcome').size())

```
Outcome
0    500
1    268
dtype: int64
```

import seaborn as snssns.countplot(diabetes ['Outcome'],label="Count")

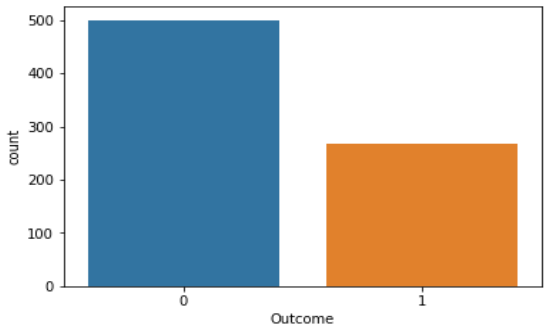

Figure 4: Diabetes information

diabetes.info()

```
RangeIndex: 768 entries, 0 to 767
Data columns (total 9 columns):
Pregnancies                 768 non-null int64
Glucose                     768 non-null int64
BloodPressure               768 non-null int64
SkinThickness               768 non-null int64
Insulin                     768 non-null int64
BMI                         768 non-null float64
DiabetesPedigreeFunction    768 non-null float64
Age                         768 non-null int64
Outcome                     768 non-null int64
dtypes: float64(2), int64(7)
memory usage: 54.1 KB
```

k-Nearest Neighbors

The k-NN algorithm is arguably the simplest machine learning algorithm. Building the model consists only of storing the training data set. To make a prediction for a new data point, the algorithm finds the closest data points in the training data set — its “nearest neighbors.”

First, let’s investigate whether we can confirm the connection between model complexity and accuracy:

from sklearn.model_selection import train_test_splitX_train, X_test, y_train, y_test = train_test_split(diabetes.loc[:, diabetes.columns != 'Outcome'], diabetes ['Outcome'], stratify=diabetes['Outcome'], random_state=66)from sklearn.neighbors import

KNeighborsClassifiertraining_accuracy = [] test_accuracy = []

try n_neighbors from 1 to 10 neighbors_settings = range(1, 11) for

```
n_neighbors in neighbors_settings:
 # build the model

knn = KNeighborsClassifier

(n_neighbors=n_neighbors)

knn.fit(X_train, y_train)

 # record training set accuracy
training_accuracy.append(knn.score
(X_train, y_train))


 # record test set accuracy
test_accuracy.append(knn.score(X_test,
y_test))plt.plot(neighbors_settings,
training_accuracy, label="training
accuracy")

plt.plot(neighbors_settings, test_accuracy,
label="test accuracy")

plt.ylabel("Accuracy")

plt.xlabel("n_neighbors")

plt.legend()

plt.savefig('knn_compare_model')
```

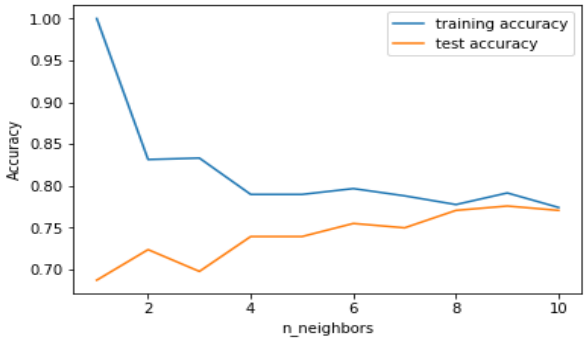

Figure 5: Training and test set accuracy

The above plot shows the training and test set accuracy on the y-axis against the setting of n_neighbour on the x-axis. Considering if we choose one single nearest neighbour, the prediction on the training set is perfect. But when more neighbour are considered, the training accuracy drops, indicating that using the single nearest neighbour leads to a model that is too complex. The best performance is somewhere around 9 neighbour.

The plot suggests that we should choose n_neighbour =9. Here we are:

```
knn = KNeighborsClassifier

(n_neighbors=9)

knn.fit(X_train, y_train)print('Accuracy of
K-NN classifier on training set:
{:.2f}'.format(knn.score(X_train, y_train)))
print('Accuracy of K-NN classifier on test
set: {:.2f}'.format(knn.score(X_test,
y_test)))
```

Accuracy of K-NN classifier on training set:

0.79

Accuracy of K-NN classifier on test set: 0.78

Logistic regression

Logistic Regression is one of the most common classification algorithms.

```
from sklearn.linear_model import

LogisticRegressionlogreg =

LogisticRegression().fit(X_train, y_train

print("Training set score: {:.3f}".forma

(logreg.score(X_train, y_train)))

print("Test set score: {:.3f}".format
(logreg.score(X_test, y_test)))
```

Training set accuracy: 0.781

Test set accuracy: 0.771

The default value of C=1 provides with 78% accuracy on the training and 77% accuracy on the test set.

```
logreg001 = LogisticRegression(C=0.01).fit
(X_train, y_train)

print("Training set accuracy: {:.3f}".format
(logreg001.score(X_train, y_train)))

print("Test set accuracy: {:.3f}".format
(logreg001.score(X_test, y_test)))
```

Training set accuracy: 0.700

Test set accuracy: 0.703

Using C=0.01 results in lower accuracy on both the training and the test sets.

```
logreg100 = LogisticRegression(C=100).fit
(X_train, y_train)

print("Training set accuracy: {:.3f}".format
(logreg100.score(X_train, y_train)))
print("Test set accuracy: {:.3f}".format
(logreg100.score(X_test, y_test)))
```

Training set accuracy: 0.785

Test set accuracy: 0.766

Using C=100 results in a little bit higher accuracy on the training set and little bit lower accuracy on the test set, confirming that less regularization and a more complex model may not generalize better than default setting.

Therefore, we should choose default value C=1.

Let's visualize the coefficients learned by the models with the three different settings of the regularization parameter C.

Stronger regularization (C=0.001) pushes coefficients more and more toward zero. Inspecting the plot more closely, we can also see that feature

"DiabetesPedigreeFunction", for C=100, C=1 and C=0.001, the coefficient is positive. This indicates that high "DiabetesPedigreeFunction" feature is related to a sample being "diabetes", regardless which model we look at.

```
diabetes_features = [x for i,x in enumerate
(diabetes.columns) if i!=8]plt.figure

(figsize=(8,6)) plt.plot

(logreg.coef_.T, 'o', label="C=1")

plt.plot(logreg100.coef_.T, '^',
label="C=100") plt.plot(logreg001.coef_.T,
'v', label="C=0.001")

plt.xticks(range(diabetes.shape[1]),

diabetes_features, rotation=90)

plt.hlines(0, 0, diabetes.shape[1])

plt.ylim(-5, 5)

plt.xlabel("Feature")

plt.ylabel("Coefficient magnitude")

plt.legend()

plt.savefig('log_coef')
```

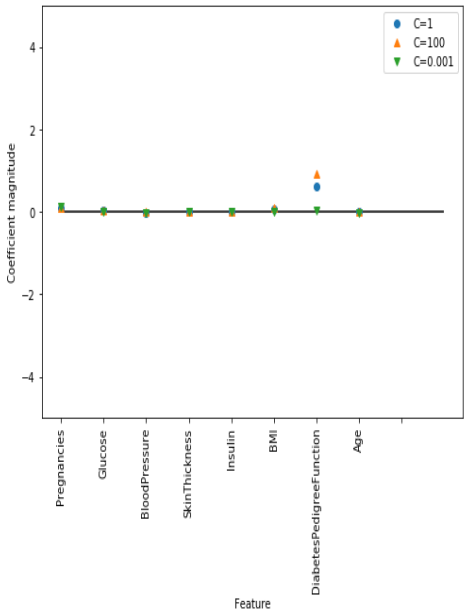

Figure 6: Feature generation and selection with coefficient

Decision Tree

from sklearn.tree import

DecisionTreeClassifiertree =

DecisionTreeClassifier(random_state=0)

tree.fit(X_train, y_train)

print("Accuracy on training set: {:.3f}".format(tree.score(X_train, y_train))) print("Accuracy on test set: {:.3f}".format (tree.score(X_test, y_test)))

Accuracy on training set: 1.000

Accuracy on test set: 0.714

The accuracy on the training set is 100%, while the test set accuracy is much worse. This is an indicative that the tree is overfitting and not generalizing well to new data. Therefore, we need to apply pre-pruning to the tree.

We set max_depth=3, limiting the depth of the tree decreases overfitting. This leads to a lower accuracy on the training set, but an improvement on the test set.

tree = DecisionTreeClassifier(max_depth=3,

random_state=0)

tree.fit(X_train, y_train)print("Accuracy on

training set: {:.3f}".format(tree.score(X_train,

y_train)))

print("Accuracy on test set: {:.3f}".format

(tree.score(X_test, y_test)))

Accuracy on training set: 0.773

Accuracy on test set: 0.740

Feature Importance in Decision Trees

Feature importance rates how important each feature is for the decision a tree makes. It is a number between 0 and 1 for each feature, where 0 means "not used at all" and 1 means "perfectly predicts the target". The feature importance always sum to 1:

print("Feature importances:\n{}".format (tree.feature_importances_))

Feature importances: [0.04554275 0.6830362 0. 0. 0. 0.27142106 0. 0.]

Then we can visualize the feature importances:

defplot_feature_importances_diabetes (model):

plt.figure(figsize=(8,6)) n_features = 8

plt.barh(range(n_features), model.feature_importances_, align='center')

plt.yticks(np.arange(n_features), diabetes_features)

plt.xlabel("Feature importance")

plt.ylabel("Feature")

plt.ylim(-1, n_features

plot_feature_importances_diabetes(tree) plt.savefig('feature_importance')

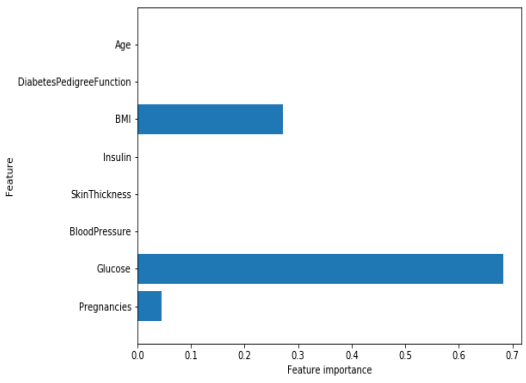

Figure 7: Feature importance

Feature "Glucose" is by far the most important feature.

Random Forest

Let's apply a random forest consisting of 100 trees on the diabetes data set:

from sklearn.ensemble import

RandomForestClassifierrf =

RandomForestClassifier(n_estimators=100,

random_state=0)

rf.fit(X_train, y_train)

print("Accuracy on training set:

{:.3f}".format(rf.score(X_train, y_train)))

print("Accuracy on test set: {:.3f}".format (rf.score(X_test, y_test)))

Accuracy on training set: 1.000

Accuracy on test set: 0.786

The random forest gives us an accuracy of 78.6%, better than the logistic regression model or a single decision tree, without tuning any parameters. However, we can adjust the max_features setting, to see whether the result can be improved.

rf1 = RandomForestClassifier

(max_depth=3, n_estimators=100,

random_state=0)

rf1.fit(X_train, y_train)

print("Accuracy on training set: {:.3f}".format(rf1.score(X_train, y_train)))

print("Accuracy on test set: {:.3f}".format (rf1.score(X_test, y_test)))

Accuracy on training set: 0.800

Accuracy on test set: 0.755

It did not, this indicates that the default parameters of the random forest work well.

Feature importance in Random Forest

plot_feature_importances_diabetes(rf)

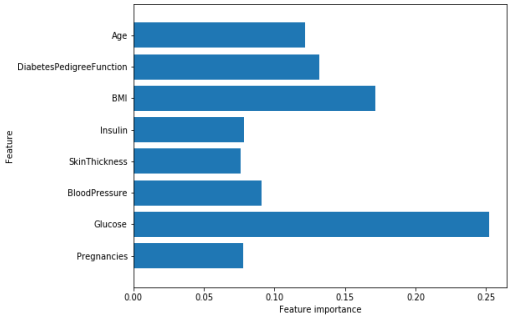

Figure 8: Feature importance diabetes

Similarly to the single decision tree, the random forest also gives a lot of importance to the "Glucose" feature, but it also chooses "BMI" to be the 2nd most informative feature overall. The randomness in building the random forest forces the algorithm to consider many possible explanations, the result being that the random forest captures a much broader picture of the data than a single tree.

Support Vector Machine

from sklearn.svm import SVCsvc = SVC() svc.fit(X_train, y_train)print("Accuracy on training set: {:.2f}".format(svc.score (X_train, y_train)))

print("Accuracy on test set: {:.2f}".format (svc.score(X_test, y_test)))

Accuracy on training set: 1.00

Accuracy on test set: 0.65

The model overfits quite substantially, with a perfect score on the training set and only 65% accuracy on the test set.

SVM requires all the features to vary on a similar scale. We will need to re-scale our data that all the features are approximately on the same scale:

```
from sklearn.preprocessing import MinMaxScalerscaler = MinMaxScaler()
X_train_scaled = scaler.fit_transform (X_train)

X_test_scaled = scaler.fit_transform (X_test)svc = SVC()

svc.fit(X_train_scaled, y_train)print ("Accuracy on training set: {:.2f}".format (svc.score(X_train_scaled, y_train))) print("Accuracy on test set: {:.2f}".format (svc.score(X_test_scaled, y_test)))
```

Accuracy on training set: 0.77

Accuracy on test set: 0.77

Scaling the data made a huge difference! Now we are actually underfitting, where training and test set performance are quite similar but less close to 100% accuracy. From here, we can try increasing either C or gamma to fit a more complex model.

```
svc = SVC(C=1000)

svc.fit(X_train_scaled, y_train)print ("Accuracy on training set: {:.3f}".format( svc.score(X_train_scaled, y_train))) print("Accuracy on test set: {:.3f}".format (svc.score(X_test_scaled, y_test)))
```

Accuracy on training set: 0.790

Accuracy on test set: 0.797

Here, increasing C allows us to improve the model, resulting in 79.7% test set accuracy.

V. Conclusions

Machine learning strategies can help physicians recognize and treat diabetic disorders. We can assume that increasing classification accuracy enables better results to be achieved for machine learning models. The success analysis is based on precision in all classification techniques, such as the decision-tab, logistic regression, the nearest neighborhood, naive bays, and SVM, random forest. We found that the precision of the current system is less than 70%, so we recommend the use of a mixture of classificators known as the hybrid solution. The combined strategy takes advantage of the merits of two or three approaches. We find that our method offers 75.32% accuracy of the Decision Tree Classifier, 77.48% accuracy of the XGBoost Classifier, 75.75% accuracy of the Vote Classifier and 80% accuracy of the Piling classifier. Therefore, we found that Stacking Classifier is the best of all the above classifiers.

VI. Future Scope

Comparative analyzes will be carried out in the future to evaluate the outputs of each algorithm as well as the hybrid if we have a large collection of diabetic data so that the best predictive analysis can be done. Initial diabetes diagnosis is not very sophisticated, and a basic approach to diabetes classification is not completely reliable for disease prediction. This is why we need a smart, hybrid-predictive analysis diagnostic device for diabetes that can function effectively and efficiently.

REFERENCES:

[1] Kalaiselvi, C., and G. M. Nasira. Classification and Prediction of Heart Disease from Diabetes Patients using Hybrid Particle Swarm Optimization and Library Support Vector Machine Algorithm.

[2] Zhang, Y., Lin, Z., Kang, Y., Ning, R., and Meng, Y. A Feed-Forward Neural Network Model for The Accurate Prediction of Diabetes Mellitus.

[3] Mirza, S., Mittal, S., and Zaman, M. (2018). Decision Support Predictive model for prognosis of diabetes using SMOTE and Decision tree. International Journal of Applied Engineering Research, 13(11),9277-9282.

[4] Yasen, M., Al-Madi, N., Obeid, N., Sumaya, P., and Abdullah II, K. Optimizing Neural Networks using Dragonfly Algorithm for Medical Prediction.

[5] Patil, R. N., and Tamane, S. C. (2018). Upgrading the Performance of KNN and Naïve Bayes in Diabetes Detection with Genetic Algorithm for Feature Selection.

[6] Choubey, D. K., and Paul, S. (2017). GA_RBF NN:a classification system for diabetes. International Journal of Biomedical Engineering and Technology, 23(1), 71-93.

[7] Jahangir, M., Afzal, H., Ahmed, M., Khurshid, K., and Nawaz, R. (2017). ECO AMLP: A Decision Support System using an Enhanced Class Outlier with Automatic Multilayer Perceptron for Diabetes Prediction .arXivpreprintarXiv:1706.07679.

[8] Ramesh, S., Caytiles, R.D., and Iyengar,N.C.S.(2017). A Deep Learning Approach to Identify Diabetes. Advanced Science and Technology Letters, 145, 44-49.

[9] Dadgar, S. M. H., and Kaardaan, M. A Hybrid Method of Feature Selection and Neural Network with Genetic Algorithm to Predict Diabetes.

[10] Md. Faisal Faruque, Asaduzzaman, Iqbal H. Sarker (2019). Performance Analysis of Machine Learning Techniques to predict Diabetes Mellitus, 2019 International Conference on Electrical, Computer and Communication Engineering (ECCE), 7-9 February, 2019.

[11] Chen,W., Chen,S., Zhang, H., and Wu, T.(2017,November).A hybrid prediction model for type 2 diabetes using K-means and decision tree. In Software Engineering and Service Science (ICSESS), 2017 8th IEEE International Conference on (pp.386-390).IEEE.

[12] Patil, R. N., and Tamane, S. C. A Novel Scheme for Predicting Type 2 Diabetes in Women: Using Kmeans with PCA AS Dimensionality Reduction.

[13] Sethi, H., Goraya, A., and Sharma, V. (2017). Artificial Intelligence based Ensemble Model for Diagnosis of Diabetes. International Journal of Advanced Research in Computer Science, 8(5).

[14] Shetty, S. P., and Joshi, S.(2016).A Tool for Diabetes Prediction and Monitoring Using Data Mining Technique. IJ Information Technology and Computer Science.

[15] LejlaAlic, Hasan T.Abbas, Marelyn Rios, Muhammad Abdul Ghani, and Khalid Qaraqe (2019) Predicting Diabetes in Healthy Population through Machine Learning.

[16] Minyechil Alehegn, Rahul Joshi and Dr. Preeti Mulay (2019) Analysis and Prediction of Diabetes Mellitus using Machine Learning Algorithm.

[17] Amani Yahyaoui, Akhtar Jamil

(2019) A Decision Support System for Diabetes Prediction Using Machine Learning and Deep Learning Techniques, 978-1-7281-3992-0/19 ©2019 IEEE.

[18] Debadri Dutta, Debpriyo Paul, Parthajeet Ghosh(2018) Analysing Feature Importances for Diabetes Prediction using Machine Learning, 2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON).

[19] Md. Maniruzzaman, Md. Jahanur Rahman, Benojir Ahammed, "Classification and prediction of diabetes disease using machine learning paradigm", Health Information Science and Systems, Springer Nature Switzerland AG 2020.

[20] Quinlan JR, Rivest RL. Inferring decision trees using the minimum description length principle. Inform Comput. 1989;80(3):227–48.

[21] Agrawal R, Ghosh S, Imielinski T, Iyer B, Swami A. An interval classifier for database mining applications. 1992. pp. 560–73.

[22] Breiman L, Friedman JH, Olshen RA, Stone CJ. Classification and regression trees. Belmont: Wadsworth International Group; 1984.

[23] Ash C, Farrow JAE, Wallbanks S, Collins MD. Phylogenetic heterogeneity of the genus bacillus revealed by comparative analysis of small subunit ribosomal RNA sequences. Lett Appl Microbiol. 1991;13:202–6.

[24] Audic S, Claverie JM. The significance of digital gene expression profiles. Genome Res. 1997;7:986–95.

[25] Wan V, Campbell W. Support vector machines for speaker verification and identification. In: IEEE proceeding. 2000.

[26] Chapelle O, Haffner P, Vapnik V. Support vector machines for histogram-based image classification. IEEE Trans Neural Netw. 1999;10 (5):1055–64.

[27] Lee JW, Lee JB, Park M, Song SH. An extensive evaluation of recent classification tools applied to microarray data. Comput Stat Data Anal. 2005;48:869–85.

[28] Yeung KY, Bumgarner RE, Raftery AE. Bayesian model averaging: development of an improved multi-class, gene selection and classification tool for microarray data. Bioinformatics. 2005;21:2394–402.

[29] Carter, J. A., Long, C. S., Smith, B. P., Smith, T. L., and Donati, G. L. (2019). Combining elemental analysis of toenails and machine learning techniques as a non-invasive diagnostic tool for the robust classification of type-2 diabetes. Expert Systems with Applications, 115, 245-255.

[30] Kavakiotis, I., Tsave, O., Salifoglou, A., Maglaveras, N., Vlahavas, I., and Chouvarda, I. (2017). Machine learning and data mining methods in diabetes research. Computational and structural biotechnology journal, 15, 104-116.

[31] Mahmud, S. M., Hossin, M. A., Ahmed, M. R., Noori, S. R. H., and Sarkar, M. N. I. (2018, August). Machine Learning Based Unified Framework for Diabetes Prediction. In Proceedings of the 2018 International Conference on Big Data Engineering and Technology (pp. 46-50). ACM

[32] Patil, R., and Tamane, S. (2018). A Comparative Analysis on the Evaluation of Classification Algorithms in the Prediction of Diabetes. International Journal of Electrical and Computer Engineering, 8(5), 3966.

[33] Dagliati, A., Marini, S., Sacchi, L., Cogni, G., Teliti, M., Tibollo, V., and Bellazzi, R. (2018). Machine learning methods to predict diabetes complications. Journal of diabetes science and technology, 12(2), 295-302

[34] Barik, R. K., Priyadarshini, R., Dubey, H., Kumar, V., and Yadav, S. (2018). Leveraging machine learning in mist computing tele-monitoring system for diabetes prediction. In Advances in Data and Information Sciences (pp. 95-104). Springer, Singapore.

[35] Choudhury, A., and Gupta, D. (2019). A Survey on Medical Diagnosis of Diabetes Using Machine Learning Techniques. In Recent Developments in Machine Learning and Data Analytics (pp. 67-78). Springer, Singapore.

[36] Samant, P., and Agarwal, R. (2017). Diagnosis of diabetes using computer methods: soft computing methods for diabetes detection using iris. Threshold, 8, 9.

* * * * *