VIT
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

SCHOOL OF INFORMATION TECHNOLOGY AND ENGINEERING

MASTER OF COMPUTER APPLICATION

DATA MINING AND BUSINESS INTELLIGENCE

"Comparative Study of Logistic Regression, k-NN, and Random Forest Models for Gestational Diabetes Prediction"

Guided by

Dr. HARSHITA PATEL

TEAM MEMBERS

SHAKTHIVEL A - 22MCA0015

JOTHIKA V – 22MCA0060

KEERTHANA Y – 22MCA0074

MONIKA R – 22MCA0270

ABSTRACT

Diabetes is the most common noncommunicable disease among people in the world due to changes in food habits. Diabetes is a disease which perpetuates in the metabolic method. It causes high blood sugar levels in an individual. The pancreas produces one of the most essential hormones of the human body, the insulin. The insulin extracts blood sugar and transports it for storage or to be used as cellular energy. For a diabetic patient, the body either produces insufficient insulin or incapable of using the insulin that has been developed. Diabetes can be broadly categorized into four types: Type1, Type2, Pre-diabetes and Gestational diabetes. In our paper we are going to predict diabetes. Diabetes occurs only during pregnancy when hormones are blocked by insulin. It affects how your cells use sugar(glucose). It causes high blood sugar that can affect pregnancy and baby's health. We are going to predict Gestational Diabetes during pregnancy using Logistic regression, KNN AND random forest as it is one of the simple and most effective Classification algorithms which helps in building the fast machine learning models for predictions by comparing this we are going to predict the gestational diabetes and which gives highest accuracy that is the best model for predicting gestational diabetes.

Keywords: Gestational diabetes, Insulin, Pregnancy, Female, Logistic Regression.

DATATSET-1

1	Pregnanci	Glucose	BloodPres	SkinThickr	Insulin	BMI	DiabetesP	Age	Outcome
2	6	148	72	35	0	33.6	0.627	50	1
3	1	85	66	29	0	26.6	0.351	31	0
4	8	183	64	0	0	23.3	0.672	32	1
5	1	89	66	23	94	28.1	0.167	21	0
6	0	137	40	35	168	43.1	2.288	33	1
7	5	116	74	0	0	25.6	0.201	30	0
8	3	78	50	32	88	31	0.248	26	1
761	6	190	92	0	0	35.5	0.278	66	1
762	2	88	58	26	16	28.4	0.766	22	0
763	9	170	74	31	0	44	0.403	43	1
764	9	89	62	0	0	22.5	0.142	33	0
765	10	101	76	48	180	32.9	0.171	63	0
766	2	122	70	27	0	36.8	0.34	27	0
767	5	121	72	23	112	26.2	0.245	30	0
768	1	126	60	0	0	30.1	0.349	47	1
769	1	93	70	31	0	30.4	0.315	23	0

DATASET-2

	Age	Gender	Family_Di	highBP	Physically	BMI	Smoking	Alcohol	Sleep	SoundSle	RegularM	JunkFood	Stress	BPLevel	Pregancie	Pdiabetes	UriationFr	Diabetic
2	50-59	Male	no	yes	one hr or	39	no	no	8	6	no	occasiona	sometime	high	0	0	not much	no
3	50-59	Male	no	yes	less than	28	no	no	8	6	yes	very ofter	sometime	normal	0	0	not much	no
4	40-49	Male	no	no	one hr or	24	no	no	6	6	no	occasiona	sometime	normal	0	0	not much	no
5	50-59	Male	no	no	one hr or	23	no	no	8	6	no	occasiona	sometime	normal	0	0	not much	no
6	40-49	Male	no	no	less than	27	no	no	8	8	no	occasiona	sometime	normal	0	0	not much	no
7	40-49	Male	no	yes	none	21	no	yes	10	10	no	occasiona	sometime	high	0	0	not much	yes
8	less than 4	Male	no	no	one hr or	24	no	no	8	8	no	occasiona	sometime	normal	0	0	not much	no
9	less than 4	Male	no	no	less than	20	no	no	7	7	yes	occasiona	sometime	low	0	0	not much	no
10	40-49	Male	yes	no	one hr or	23	no	no	7	7	no	occasiona	sometime	normal	0	0	not much	no
11	less than 4	Male	no	no	more thar	20	no	no	8	8	o	occasiona	sometime	normal	0	0	not much	no
12	less than 4	Male	no	no	none	20	no	no	7	7	no	occasiona	not at all	normal	0	0	not much	no
13	40-49	Male	no	no	less than	26	yes	no	8	7	no	occasiona	sometime	normal	0	0	not much	no
14	less than 4	Female	no	no	less than	21	no	no	6	6	no	occasiona	sometime	normal	1	0	not much	no
15	less than 4	Female	no	no	one hr or	22	no	no	8	7	no	occasiona	very ofter	normal	2	0	not much	no

DATASET-3

Gestational Diabetic Dat Set

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1	Case Num	Age	No of Pregnan	Gestati	BMI	HDL	Family His	unexplain	Large Chil	PCOS	Sys BP	Dia BP	OGTT	Hemoglot	Sedentary	Prediabet	Class Label	(GDM /Non GDM)	
2	1	22	2	1		55	0	0	0	0	102	69		12	0	0	0		
3	2	26	2	1		53	0	0	0	0	101	63		12.4	0	0	0		
4	3	29	1	0		50	0	0	0	0	118	79		14.3	0	0	0		
5	4	28	2	1		51	0	0	0	0	99	70		15	0	0	0		
6	5	21	2	1		52	0	0	0	0	116	65		15	0	0	0		
7	6	29	2	1		51	0	0	0	0	98	63		15.2	0	0	0		
8	7	26	2	1		51	0	0	0	0	94	68		15	0	0	0		
9	8	27	1	0		52	0	0	0	0	116	63		12	0	0	0		
10	9	26	1	0		57	0	0	0	0	108	62		14	0	0	0		
11	10	21	2	1		52	0	0	0	0	98	78		13	0	0	0		
12	11	21	2	1		56	0	0	0	0	100	76		14	0	0	0		
13	12	26	2	1		50	0	0	0	0	110	68		13	0	0	0		
14	13	27	2	1		55	0	0	0	0	105	61		13.6	0	0	0		
15	14	25	2	1		58	0	0	0	0	106	80		15	0	0	0		
16	15	22	1	0		53	0	0	0	0	109	61		15.9	0	0	0		
17	16	22	2	1		57	0	0	0	0	107	80		14	0	0	0		
18	17	27	1	0		57	0	0	0	0	100	61		13	0	0	0		
19	18	21	2	1		58	0	0	0	0	105	78		13	0	0	0		
20	19	28	1	0		50	0	0	0	0	112	77		12.1	0	0	0		
21	20	20	2	1		54	0	0	0	0	97	74		13	0	0	0		
22	21	20	1	0		50	0	0	0	0	93	66		12	0	0	0		
23	22	23	1	0		51	0	0	0	0	116	72		14	0	0	0		

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
3505	3504	43	3	1	23.4	53	1	1	1	0	143	89	185	17.8	0	1	0
3506	3505	26	4	2	18.7	23	0	0	1	0	168	112	150	12.1	0	1	1
3507	3506	31	2	1	32	61	1	0	0	0	130	99	123	14.6	0	1	1
3508	3507	33	4	1	37.4	29	1	1	1	0	140	115	124	16.8	1	0	1
3509	3508	40	3	0	29.4	24	0	0	1	1	178	74	152	12.8	1	0	0
3510	3509	39	2	1	28.5	26	1	1	1	0	130	110	188	14.1	1	0	1
3511	3510	42	3	0	21.3	24	0	0	1	1	160	68	158	13.6	1	0	0
3512	3511	26	1	2	39.1	29	1	1	1	1	149	114	148	16.5	0	1	0
3513	3512	45	2	1	27.4	56	1	0	1	0	130	103	165	14.1	0	0	0
3514	3513	38	1	2	19.9	31	0	1	0	1	180	67	153	17.9	1	1	0
3515	3514	40	3	1	33.7	40	1	1	0	0	121	71	120	13.3	0	0	0
3516	3515	35	1	0	29.8	39	1	0	0	1	140	84	120	15.4	0	0	0
3517	3516	33	2	0	22.7	48	0	1	1	0	148	83	179	13.9	1	1	0
3518	3517	45	2	0	18.9	29	0	0	1	0	131	105	157	14.9	1	0	1
3519	3518	37	2	0	32	31	0	0	0	1	167	99	126	13	0	1	1
3520	3519	43	4	0	22	68	0	1	1	0	173	109	191	13.3	0	0	1
3521	3520	35	3	0	30	22	0	1	1	1	182	118	186	12.7	0	0	1
3522	3521	31	4	1	24.1	32	0	0	1	0	150	107	187	13.4	1	1	1
3523	3522	26	3	1	34.5	43	1	1	0	1	166	85	164	14.2	0	0	1
3524	3523	35	2	2	23.6	56	1	0	1	0	178	81	141	15.3	0	1	1
3525	3524	37	2	0	23.3	28	1	0	1	1	139	115	133	13.3	0	1	0
3526	3525	43	2	0	28.6	30	1	1	0	0	121	63	179	15.8	1	0	0
3527																	

GDM-Final2022

Source: Kaggle

DATASET DETAILS:

This dataset is originally from the National Institute of Diabetes and Digestive and Kidney Diseases. Several constraints were placed on the selection of these instances from a larger database. In particular, all patients here are females at least 21 years old of Pima Indian heritage.

The datasets consist of several medical predictor variables and one target variable, outcome. Predictor variables includes the number of pregnancies the patient has had, their BMI, insulin level, age, and so on.

PROBLEM DESCRIPTION

The Symptoms of Gestational Diabetes are not identified sometimes it may lead to cause death of baby. So to avoid this we are predicting gestational diabetes during pregnancy using Logistic Regression, KNN and Random forest we are going to predict the gestational diabetes by comparing accuracy and by doing so we can save baby and mother from death.

DATA REQUIREMENTS

Patient information: The application should collect basic patient information such as age, gender, BMI, and whether the patient is pregnant.

Medical history: The application should collect information on the patient's medical history, including blood pressure, skin thickness, insulin level, and glucose level.

FUNCTIONAL REQUIREMENTS

SOFTWARE REQUIREMENT:

IDE: VSCode or Jupyter Notebook or PyCharm etc or any other IDE

Python Package: pandas, numpy, matplotlib, Sklearn, Dense, LSTM, Dropout, keras.

HARDWARE REQUIREMENTS NEEDED FOR THE USER ARE AS FOLLOWS:

Processor: Intel i5 or above
RAM: Minimum 225MB or more.
Hard Disk: Minimum 2 GB of space
Input Device: Keyboard
Output Device: Screens of Monitor or a Laptop

DATASET PREPROCESSING

Data input: The application should be able to input patient data, either manually or through a CSV file.

Data preprocessing: The application should preprocess the input data, including removing any missing values and scaling the data.

Feature selection: The application should select the most relevant features for the prediction model to reduce the number of unnecessary features.

Model training and testing: The application should train a machine learning model on the preprocessed and selected features data, and then test the model on a separate test dataset to ensure its accuracy.

Prediction: The application should allow the user to input patient data, and then use the trained model to predict whether the patient is at risk of developing diabetes.

Model interpretation: The application should provide a detailed interpretation of the model's predictions, including the significance of each feature and its contribution to the final prediction.

PROPOSED METHODS

Logistic Regression

Logistic regression predicts the output of a categorical dependent variable. Therefore the outcome must be a categorical or discrete value. It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, it gives the probabilistic values which lie between 0 and 1.The dependent variable must be categorical in nature. The independent variable should not have multi-collinearity. Regression equations are given below: Equation of the straight line can be written as: Logistic Regression in Machine Learning In Logistic Regression y can be between 0 and 1 only, so for this let's divide the above equation by (1-y):Logistic Regression in Machine Learning. But we need range between –[infinity] to +[infinity], then take logarithm of the equation it will become: Logistic Regression in Machine Learning The above equation is the final equation for Logistic Regression**.**

Random forest

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on

the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model.

As the name suggests, "Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the random forest takes the prediction From each tree and based on the majority votes of predictions, and it predicts the final output. The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.

KNN

K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique. K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories. K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm. K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems. K-NN is a non-parametric algorithm, which means it does not make any assumption on underlying data. It is also called a lazy learner algorithm because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset. KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data.

KNN Working

- **Step-1:** Select the number K of the neighbors
- **Step-2:** Calculate the Euclidean distance of K number of neighbors
- **Step-3:** Take the K nearest neighbors as per the calculated Euclidean distance.
- **Step-4:** Among these k neighbors, count the number of the data points in each category.
- **Step-5:** Assign the new data points to that category for which the number of the neighbor is maximum.
- **Step-6:** Our model is ready.

PREPROCESSING METHODS

Preprocessing is an important step in any machine learning project, and it is particularly important for datasets like the Pima Indian Diabetes dataset which may contain missing values, outliers, or other anomalies that need to be addressed before the data can be used for training

a model. Here are some common preprocessing techniques that can be used for the Pima Indian Diabetes dataset:

Missing values handling: One common issue with the Pima Indian Diabetes dataset is missing values. This can be addressed by either removing the rows with missing values or by imputing the missing values with a suitable value (such as the mean or median of the remaining values).

Outlier detection and removal: Outliers can have a significant impact on the performance of machine learning models, so it is important to identify and remove them. One way to do this is by using statistical methods such as the Z-score or Interquartile Range (IQR).

Feature scaling: Since the Pima Indian Diabetes dataset contains features with different scales, it is important to scale the data before training a model. This can be done using techniques such as normalization or standardization.

Feature selection: Feature selection is the process of selecting the most relevant features for the prediction model to reduce the number of unnecessary features. This can be done using techniques such as correlation analysis or feature importance ranking.

Data splitting: The Pima Indian Diabetes dataset can be split into training and testing datasets to evaluate the performance of the model. The split can be done randomly or using stratified sampling to ensure that both datasets have a similar distribution of target classes.

LITERATURE SURVEY

[1]This paper Assissted Reproductive Technology(ART) it has been employed to overcome the infertility problem and also it is the final treatment for the infertility couples. In the other studies GDM has the high preference after the ART procedures. GDM is considered as the major metabolic disorder of the pregnancy and still it has not been established completely. The aim of the study is to access the GDM predictive factor in women who convinced and underwent to the ART procedures. In the case of the inclusion and exclusion crietria in a hospital total 270 women who were pregnant in that 135 with GDM and other 135 are with the non GDM problem. To get detected form the pre-gestational disease all patients were examined with the test of Fasting Blood Sugar(FBS). There occurs the trained nurses for initiation of ART cycle of the pre-pregnancy measured height and weight. In this the statistical analysis of the chi-square test was applied to compare the cause of the typer of infertility and also the history of abortions and the qualitative variables between the two groups. At last the Binary logistic Regression has been employed to predict the risk factor of the GDM development after the ART cycle.so these are the consequences and terms followed by the of reproductive technology in gestational diabetes in pregnancies.

[2] In this study, there also includes gestational diabetes mellitus(GDM) there occurs null viperous women aged 18-23 who reported about the pregency upto the age of 37-42 were included. GDM is an increasingly common obstetric complications that occurs upto 5-15% of pregnancy. Evidence and potential for preconception and prevention of GDM comes from the observational studies. Though several studies have reported on the risk factor that involved in the osteology of GDM. This study aims to develop a model based on multiple and easily accessible maternal characteristics of the first pregency that is based on the women's absolute risk of developing GDM. Candidates predictor is selected based upon the literature association with GDM. Internal validity were also be accessed on the account of the potential outfitting and the optimization of the and the model of the dataset. In this study they have validated the model of the preconception prediction on the GDM. This model helps in the clinician to identify the women's of high risk in GDM in the first appointment and thereby they will inform you about the early scaning, monitoring and treatment that should be done on this basics. This research paper is based and also conducted by the Australian Longitudinal Study on women's health and were located in the news castle and also it is the university of the queens land.

[3] In this study, they presented a comprehensive review of the state-of-the-art on the glycemic control in the domains of data mining based diabetes diagnosis and prediction techniques and their classification based on the underlying models used. Based on the data mining based techniques for diabetes detection, classification and prediction, we provide a comprehensive classification of the commonly used diabetes diagnosis and prediction techniques. Moreover, they evaluated different schemes on parameters like, algorithm/model, type of input data (data input), plug-n-play capability, etc. On the basis of this analysis and evaluation, they conclude that for accurate detection, classification, and prediction of the disease, they preprocess the data and use hybrid techniques, which incorporate different models in parallel instead of using an individual model. For preprocessing, they want to use dimensionality reduction, denoising, feature selection, and feature extraction techniques in combination with the classification and prediction schemes for optimal performance and results.

[4] In this study, a systematic experimental study is followed by using various algorithms to predict the diabetes. Logistic regression with all features gives better result (ACC = 84.70 %) for prediction of diabetes in comparison of other algorithms. they performed feature engineering to select the suitable features, has a better performance with respect to other models. It means Logistic regression with some features gives the best accuracy of 85 % and they cannot predict a person has which type of diabetes, they future our aim is to predict the type of diabetes and analyze the feature of each indicator, which may improve the accuracy of diabetes prediction in that paper.

[5] In this study, Early prediction of diabetes will result in improved results. This paper presents a diabetes prediction model with the help of data mining techniques. they apply Naive Bayes, Logistic Regression, Random Forest, and Support Vector Machine techniques to predict diabetes disease. To analyse the proposed mechanism, a real dataset is collected from Kaggle. Accuracy, confusion, and sensitivity matrices are used to assess performance. In the logistic regression model, the accuracy is high, i.e., 82.46% as compared to other models. whereas in SVM the accuracy is low, i.e., 79.22% was compared to other models. In that paper future, it is intended to continue working on it and apply more classification algorithms to predict diabetes datasets. It is also meant to suggest a new way to make predictions about diabetes outcomes more accurate.

[6] This study proposes a new implementation of the Bayesian Classifier to analyze raw medical data and determine the risk of diabetes diagnosis based on a patient's medical history. The classifier uses basic patient information to calculate the likelihood of a positive diabetes diagnosis, providing a valuable tool for both healthcare professionals and the general public in detecting and preventing diabetes. The study highlights the potential of using such classifiers as a low-cost and accessible method for diabetes risk assessment.

[7] A recent study examined multiple machine learning (ML) models to predict gestational diabetes mellitus (GDM) in women using the PIMA dataset. To enhance data quality, the study imputed missing data and utilized feature selection techniques. The researchers compared the accuracies of various ML algorithms using ROC and AUC scores and analyzed the confusion matrix parameters to evaluate model effectiveness and errors. The study provides valuable insights to healthcare organizations regarding hospital readmission rates for women with postpartum diabetes. Further improvement in ML algorithm accuracy can be achieved by hyperparameter tuning and testing on different sparse datasets.

[8] In this study when compared to current screening strategies, ML methods are attractive for predicting GDM. To expand their use, the importance of quality assessments and unified diagnostic criteria should be further emphasized. In this ML methods demonstrate high performance and will be a more selective and cost-effective screening method for GDM. The importance of quality assessment and unified diagnostic criteria should be further

[9] In this case study, the increasing prevelance of the gestational diabetes millitus(GDM) is contributing to the rising global burden of the type 2 diabetes. It has aimed to built the preconception based GDM predictor to enable early intervention.They have also assessed the associations of top predictors with GDM and adverse birth outcomes.The results of multivariate logistic regression model showed that each 1 mmol/mol increase in preconception HbA1c was positively associated with increased risks of GDM (p = 0.001, odds ratio (95% CI) 1.34 (1.13–1.60)) and preterm birth (p = 0.011, odds ratio 1.63 (1.12–2.38)). The Optimal control of

preconception HbA1c may aid in preventing GDM and reducing the incidence of preterm birth. They have trained predictor has been deployed as a web application that can be easily employed in GDM intervention programs, prior to conception.

[9] In this case the Gestational diabetes mellitus (GDM) can cause adverse consequences to both mothers and their newborns.The outstanding performance of artificial intelligence (AI) in disease diagnosis in previous studies demonstrates its promising applications in GDM diagnosis.This study aims to investigate the implementation of a well-performing AI algorithm in GDM diagnosis in a setting, which requires fewer medical equipment and staff and to establish an app based on the AI algorithm. This study also explores possible progress if our app is widely used.GDM was diagnosed according to American Diabetes Association (ADA) 2011 diagnostic criteria. Age and fasting blood glucose were chosen as critical parameters. Accuracy, sensitivity, and other criteria were calculated for each algorithm.The areas under the receiver operating characteristic curve (AUROC) of external validation dataset for support vector machine (SVM), random forest, AdaBoost, k-nearest neighbors (kNN), naive Bayes (NB), decision tree, logistic regression (LR), extreme gradient boosting (XGBoost), and gradient boosting decision tree (GBDT) were 0.780, 0.657, 0.736, 0.669, 0.774, 0.614, 0.769, 0.742, and 0.757, respectively. SVM also retained high performance in other criteria.

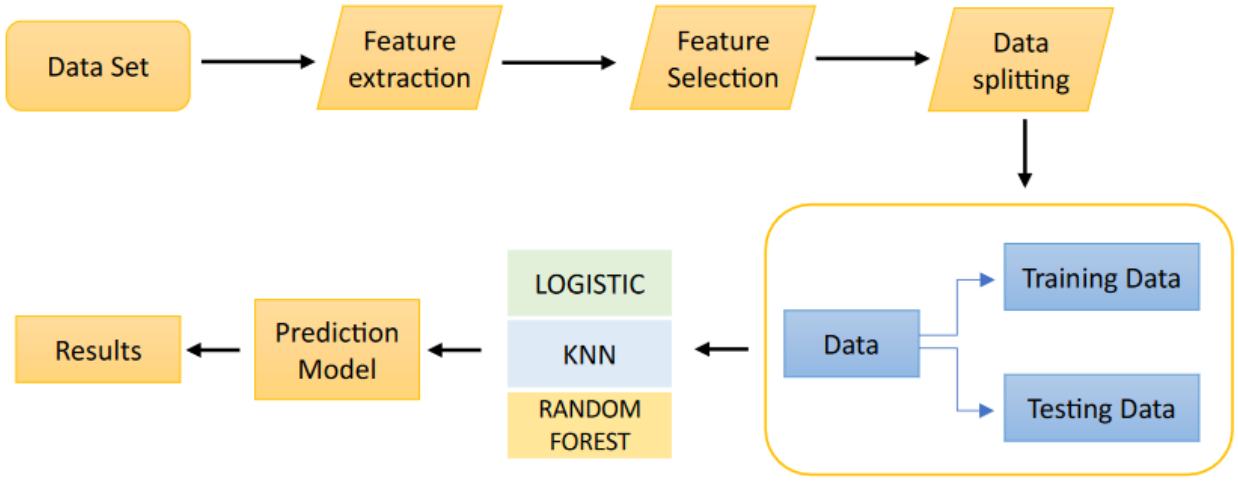
[10] This study aims to investigate the implementation of a well-performing AI algorithm in GDM diagnosis in a setting, which requires fewer medical equipment and staff and to establish an app based on the AI algorithm. This study also explores possible progress if our app is widely used.An AI model that included 9 algorithms was trained on 12,304 pregnant outpatients with their consent who received a test for GDM in the obstetrics and gynecology department of the First Affiliated Hospital of Jinan University, a local hospital in South China, between November 2010 and October 2017. GDM was diagnosed according to American Diabetes Association (ADA) 2011 diagnostic criteria. Age and fasting blood glucose were chosen as critical parameters.

The Usage of classification has been applied in various areas, including in health areas. One of the classification methods used is Naive Bayes.

[12] In this study ,&e data for the proposed model were collected from lab-Oratories in the Iraqi Kurdistan Region. &e dataset includes 1012 instances and 7 attributes: age, pregnancy number,Weight, height, BMI, heredity, and blood sugar test. A mixed Prediction model in the proposed model has been developed To identify gestational diabetes. With the help of the elbow technique, the KMeans algorithm was used to cluster the data into an optimal number of clusters. &e Mahalanobis distance method is used to select the most related cluster Which is most closely connected to the new samples. In the prediction section, classification techniques such as DT, RF,NB, and KNN with 92% accuracy and SVM and LR with 90% accuracy were used

for ensemble techniques. &e obtained accuracy for the ensemble max voting method is 92%. Finally, the findings show that using a mix of Means clustering, elbow method, Mahala Nobis distance, and ensemble learning significantly improves prediction accuracy. In future work, we will try developing the proposed model for an adaptive healthcare application to predict diabetes instances, especially in the mentioned geographic area for which the necessary attention has not been paid to this issue. &e application will be designed to get a new sample and add it to the dataset. &is method updates the database daily, so each time the model is trained, it will have more data to work With.

ARCHITECTURE DIAGRAM



CODE

Libraries

```
import pandas as pd
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, f1_score, roc_auc_score
from sklearn.metrics import roc_curve, auc
import matplotlib.pyplot as plt
```

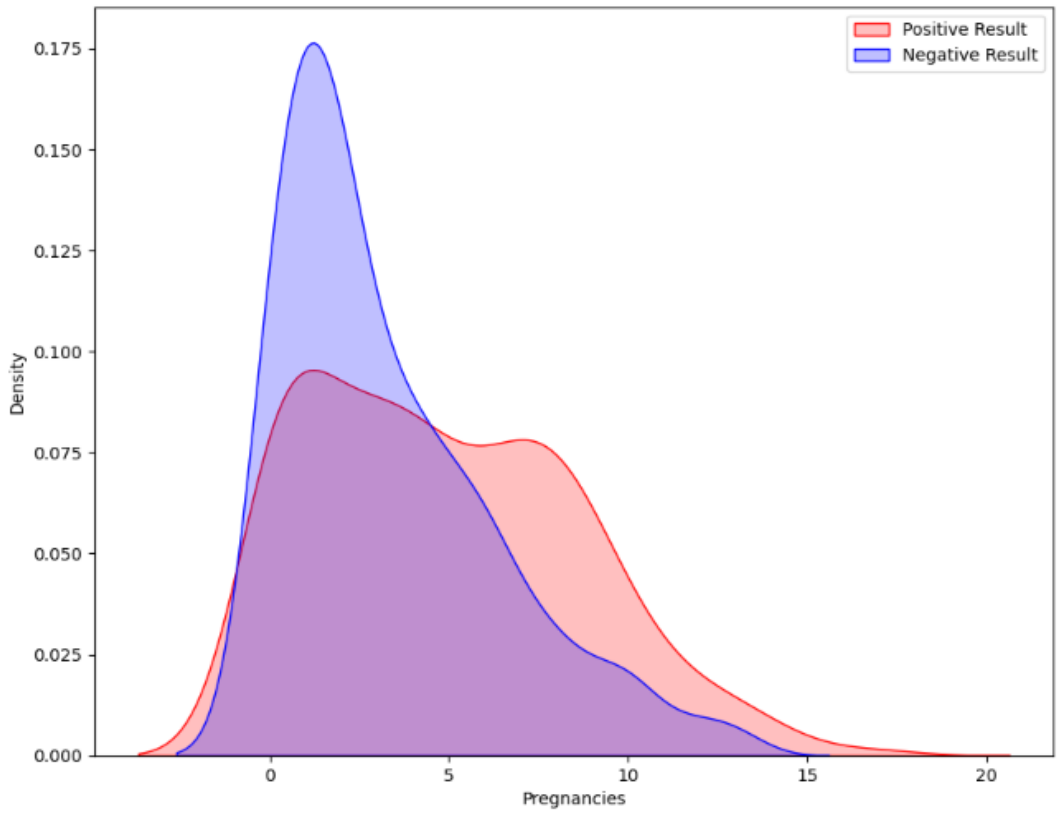
Importing the dataset:

```
data = pd.read_csv(r'C:\Users\91915/Diabetes.csv')

# First step is getting familiar with the structure of the dataset
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 768 entries, 0 to 767
Data columns (total 9 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Pregnancies               768 non-null    int64
 1   Glucose                   768 non-null    int64
 2   BloodPressure             768 non-null    int64
 3   SkinThickness             768 non-null    int64
 4   Insulin                   768 non-null    int64
 5   BMI                       768 non-null    float64
 6   DiabetesPedigreeFunction  768 non-null    float64
 7   Age                       768 non-null    int64
 8   Outcome                   768 non-null    int64
dtypes: float64(2), int64(7)
memory usage: 54.1 KB
```

```
plt.figure(figsize = (10, 8))

# Plotting density function graph of the pregnancies and the target variable
kde = sns.kdeplot(data["Pregnancies"][data["Outcome"] == 1], color = "Red", shade = True)
kde = sns.kdeplot(data["Pregnancies"][data["Outcome"] == 0], ax = kde, color = "Blue", shade= True)
kde.set_xlabel("Pregnancies")
kde.set_ylabel("Density")
kde.legend(["Positive Result", "Negative Result"])
```

```
<matplotlib.legend.Legend at 0x244f9eb80a0>
```



```
# Splitting the dependent and independent features
X = data.drop(["Outcome"], axis = 1)
Y = data["Outcome"]

# Splitting the dataset into the training and testing dataset
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.90, random_state =0)
```

```
# Logistic Regression model
logreg = LogisticRegression()
logreg.fit(X_train, Y_train)
logreg_pred = logreg.predict(X_test)
logreg_prob = logreg.predict_proba(X_test)[:, 1]
```

```
# KNN model
knn = KNeighborsClassifier()
knn.fit(X_train, Y_train)
knn_pred = knn.predict(X_test)
knn_prob = knn.predict_proba(X_test)[:, 1]
```

```
# Random Forest model
rf = RandomForestClassifier()
rf.fit(X_train, Y_train)
rf_pred = rf.predict(X_test)
rf_prob = rf.predict_proba(X_test)[:, 1]
```

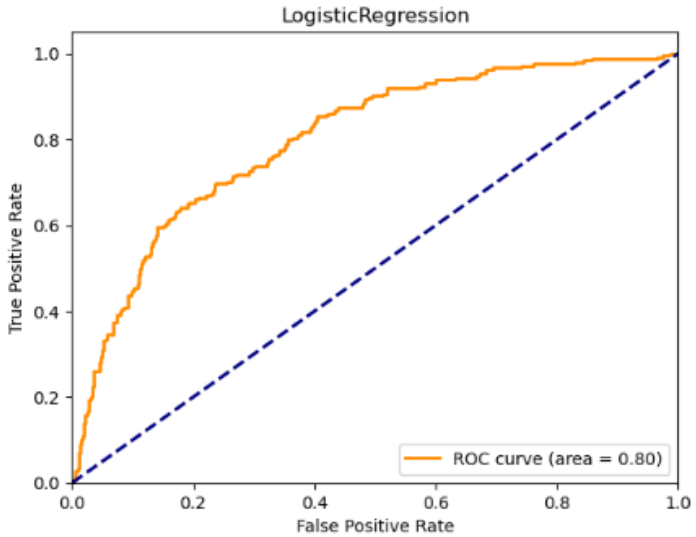
```
# Performance Metrics
logreg_acc = accuracy_score(Y_test, logreg_pred)
logreg_f1 = f1_score(Y_test, logreg_pred)
logreg_auc = roc_auc_score(Y_test, logreg_prob)

knn_acc = accuracy_score(Y_test, knn_pred)
knn_f1 = f1_score(Y_test, knn_pred)
knn_auc = roc_auc_score(Y_test, knn_prob)

rf_acc = accuracy_score(Y_test, rf_pred)
rf_f1 = f1_score(Y_test, rf_pred)
rf_auc = roc_auc_score(Y_test, rf_prob)
```

```
# Print the performance metrics
print("Logistic Regression:")
print("Accuracy:", logreg_acc)
print("F1-Score:", logreg_f1)
print("AUC/ROC:", logreg_auc)

print("\nK-Nearest Neighbors:")
print("Accuracy:", knn_acc)
print("F1-Score:", knn_f1)
print("AUC/ROC:",knn_auc)

print("\nRandom Forest:")
print("Accuracy:", rf_acc)
print("F1-Score:", rf_f1)
print("AUC/ROC:", rf_auc)
```

```
Logistic Regression:
Accuracy: 0.7658959537572254
F1-Score: 0.6431718061674009
AUC/ROC: 0.8005113454777885

K-Nearest Neighbors:
Accuracy: 0.6647398843930635
F1-Score: 0.4341463414634147
AUC/ROC: 0.6578870474364242

Random Forest:
Accuracy: 0.7471098265895953
F1-Score: 0.5700245700245701
AUC/ROC: 0.7946902250833219
```

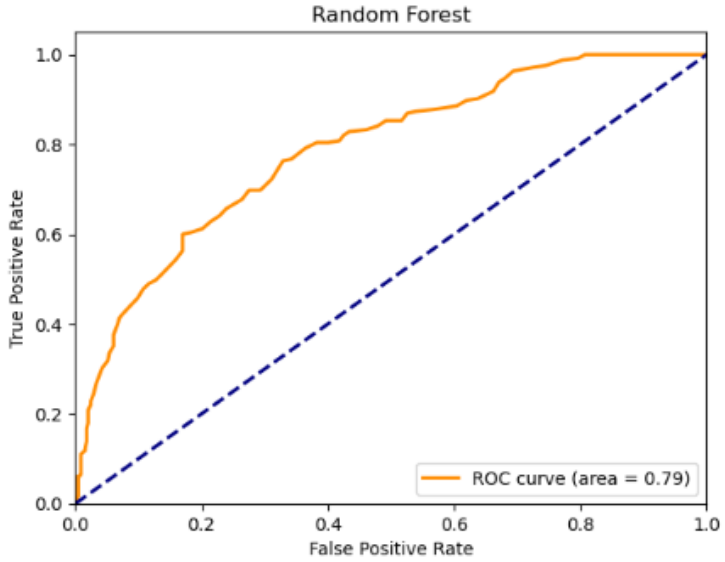
```
# Plot ROC Curve for  LogisticRegression
fpr, tpr, _ = roc_curve(Y_test, logreg_prob)
roc_auc = auc(fpr, tpr)
```

```
plt.figure()
plt.plot(fpr, tpr, color='darkorange', lw=2, label='ROC curve (area = %0.2f)' % roc_auc)
plt.plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('LogisticRegression')
plt.legend(loc="lower right")
plt.show()
```



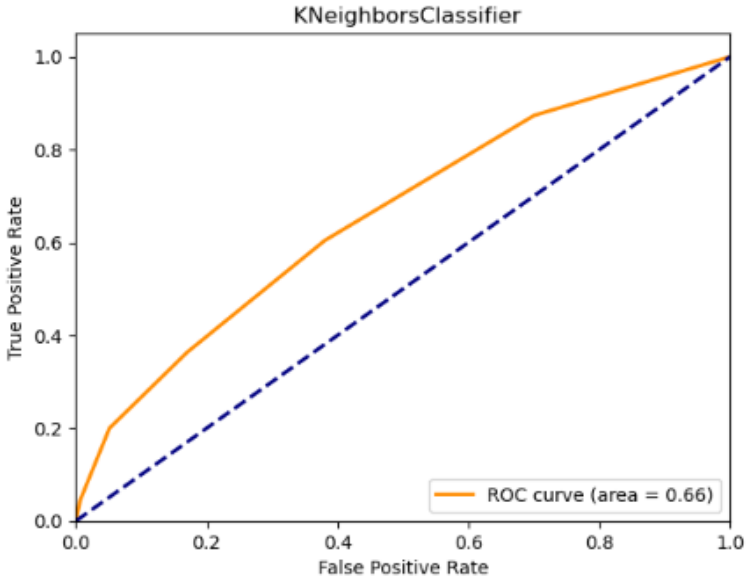
```
# Plot ROC Curve for Random Forest
fpr, tpr, _ = roc_curve(Y_test, rf_prob)
roc_auc = auc(fpr, tpr)
```

```
plt.figure()
plt.plot(fpr, tpr, color='darkorange', lw=2, label='ROC curve (area = %0.2f)' % roc_auc)
plt.plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Random Forest')
plt.legend(loc="lower right")
plt.show()
```




```
# Plot ROC Curve for  KNeighborsClassifier
fpr, tpr, _ = roc_curve(Y_test, knn_prob)
roc_auc = auc(fpr, tpr)
```

```
plt.figure()
plt.plot(fpr, tpr, color='darkorange', lw=2, label='ROC curve (area = %0.2f)' % roc_auc)
plt.plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title(' KNeighborsClassifier')
plt.legend(loc="lower right")
plt.show()
```



CODE DATASET-3

```
import pandas as pd
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, f1_score, roc_auc_score
from sklearn.metrics import roc_curve, auc
import matplotlib.pyplot as plt
```

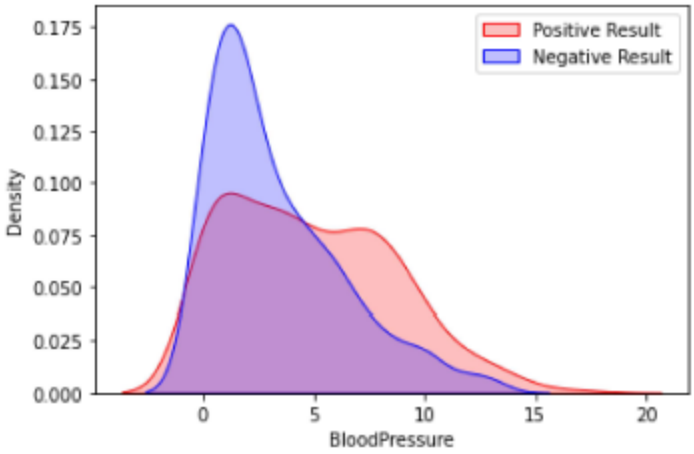
```
# Load the dataset
data = pd.read_csv('Gestational Diabetic DataSet.csv')
plt.figure(figsize = (10, 8))
```

```
<Figure size 720x576 with 0 Axes>
```


```
<Figure size 720x576 with 0 Axes>
```


```
# Plotting density function graph of the pregnancies and the target variable
kde = sns.kdeplot(data["BloodPressure"][data["Outcome"] == 1], color = "Red", shade =
kde = sns.kdeplot(data["BloodPressure"][data["Outcome"] == 0], ax = kde, color = "Blue
kde.set_xlabel("BloodPressure")
kde.set_ylabel("Density")
kde.legend(["Positive Result", "Negative Result"])
```

```
<matplotlib.legend.Legend at 0x12718d83040>
```




```
# Splitting the dependent and independent features

X = data.drop(["Outcome"], axis = 1)
Y = data["Outcome"]

# Splitting the dataset into the training and testing dataset

X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.90, random_sta
```

```
# Logistic Regression model

logreg = LogisticRegression()
logreg.fit(X_train, Y_train)
```

```
logreg_pred = logreg.predict(X_test)
logreg_prob = logreg.predict_proba(X_test)[:, 1]
```

```
C:\Users\sanja\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:814: Con
vergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n iter i =  check optimize result(
```

```
# KNN model

knn = KNeighborsClassifier()
knn.fit(X_train, Y_train)
knn_pred = knn.predict(X_test)
knn_prob = knn.predict_proba(X_test)[:, 1]
```

```
# Random Forest model
rf = RandomForestClassifier()
rf.fit(X_train, Y_train)
rf_pred = rf.predict(X_test)
rf_prob = rf.predict_proba(X_test)[:, 1]
```

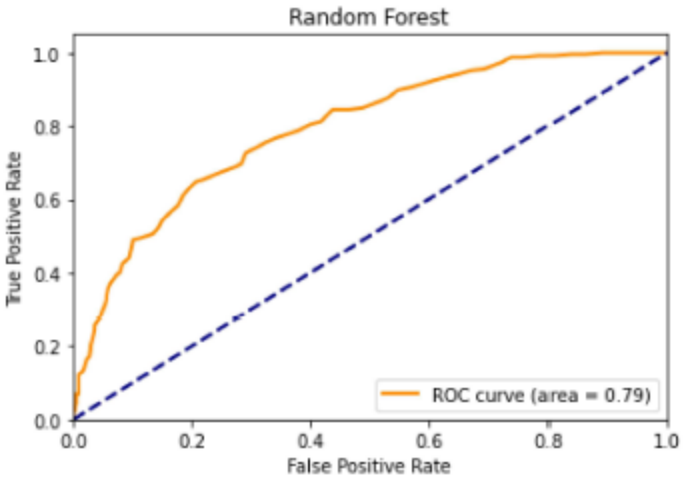
```
# Performance Metrics
logreg_acc = accuracy_score(Y_test, logreg_pred)
logreg_f1 = f1_score(Y_test, logreg_pred)
logreg_auc = roc_auc_score(Y_test, logreg_prob)

knn_acc = accuracy_score(Y_test, knn_pred)
knn_f1 = f1_score(Y_test, knn_pred)
knn_auc = roc_auc_score(Y_test, knn_prob)

rf_acc = accuracy_score(Y_test, rf_pred)
rf_f1 = f1_score(Y_test, rf_pred)
rf_auc = roc_auc_score(Y_test, rf_prob)
```

```
# Print the performance metrics
print("Logistic Regression:")
print("Accuracy:", logreg_acc)
print("F1-Score:", logreg_f1)
print("AUC/ROC:", logreg_auc)

print("\nK-Nearest Neighbors:")
print("Accuracy:", knn_acc)
print("F1-Score:", knn_f1)
print("AUC/ROC:",knn_auc)

print("\nRandom Forest:")
print("Accuracy:", rf_acc)
print("F1-Score:", rf_f1)
print("AUC/ROC:", rf_auc)
```

```
Logistic Regression:
Accuracy: 0.7658959537572254
F1-Score: 0.6431718061674009
AUC/ROC: 0.8005022143085423

K-Nearest Neighbors:
Accuracy: 0.6647398843930635
F1-Score: 0.4341463414634147
AUC/ROC: 0.6578870474364242

Random Forest:
Accuracy: 0.7543352601156069
F1-Score: 0.5853658536585367
AUC/ROC: 0.792763548372369
```

```
# Plot ROC Curve for Random Forest

fpr, tpr, _ = roc_curve(Y_test, rf_prob)
roc_auc = auc(fpr, tpr)
```

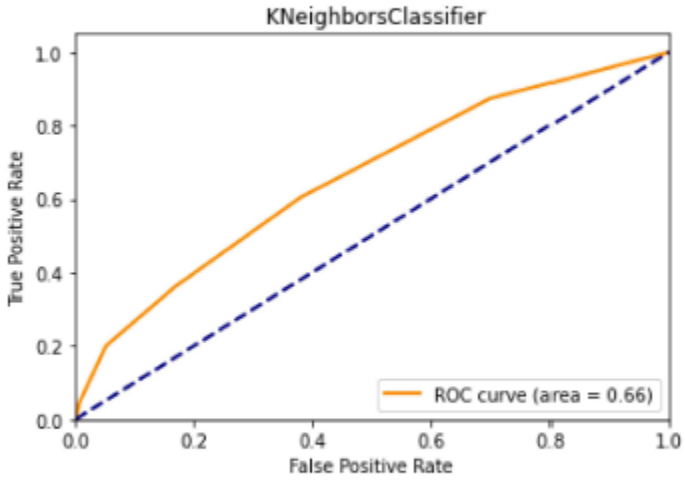
```
plt.figure()
plt.plot(fpr, tpr, color='darkorange', lw=2, label='ROC curve (area = %0.2f)' % roc_au
plt.plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Random Forest')
plt.legend(loc="lower right")
plt.show()
```




```
# Plot ROC Curve for  KNeighborsClassifier
fpr, tpr, _ = roc_curve(Y_test, knn_prob)
roc_auc = auc(fpr, tpr)
```

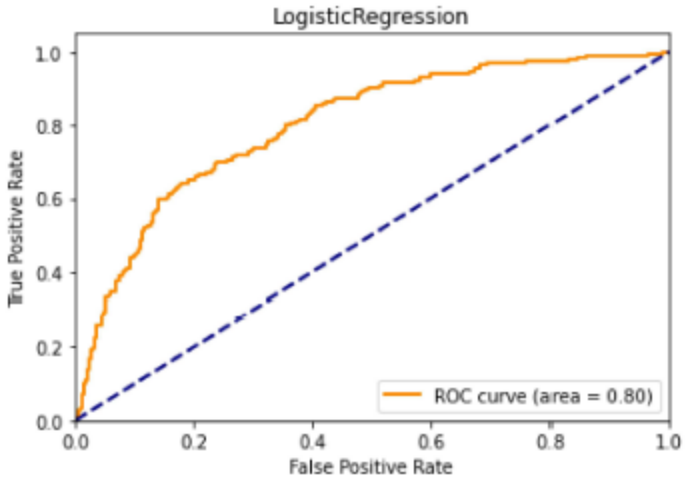
```
plt.figure()
plt.plot(fpr, tpr, color='darkorange', lw=2, label='ROC curve (area = %0.2f)' % roc_au
plt.plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
```

```
plt.ylabel('True Positive Rate')
plt.title(' KNeighborsClassifier')
plt.legend(loc="lower right")
plt.show()
```




```
# Plot ROC Curve for  LogisticRegression
fpr, tpr, _ = roc_curve(Y_test, logreg_prob)
roc_auc = auc(fpr, tpr)
```

```
plt.figure()
plt.plot(fpr, tpr, color='darkorange', lw=2, label='ROC curve (area = %0.2f)' % roc_au
plt.plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('LogisticRegression')
plt.legend(loc="lower right")
plt.show()
```



FUTURE WORK

Perform an extensive hyperparameter search for each model to optimize their performance. Use techniques such as grid search, random search, or Bayesian optimization to find the best combination of hyperparameters for each algorithm. This process can help improve the accuracy of the models and also can Evaluate the impact of class imbalance on model performance. Since gestational diabetes might be a relatively rare condition compared to non-gestational diabetes cases, the dataset might be imbalanced. Experiment with different strategies to address this issue, such as oversampling the minority class (gestational diabetes) or under sampling the majority class (non-gestational diabetes) using techniques like SMOTE or random under sampling.

CONCLUSION

Based on the comparative study, logistic regression demonstrated the highest accuracy among the logistic regression, KNN ,and random forest models for predicting gestational diabetes. This finding suggests that logistic regression may be a suitable algorithm for early detection and identification of gestational diabetes in pregnant women. However, further research and validation on larger and diverse datasets are necessary to confirm the generalizability and robustness of these findings.

REFRENCES

1. https://pubmed.ncbi.nlm.nih.gov/29730813/
2. https://pubmed.ncbi.nlm.nih.gov/30296462/
3. https://ieeexplore.ieee.org/document/7984706
4. https://iopscience.iop.org/article/10.1088/1757-899X/1116/1/012135
5. https://www.sciencedirect.com/science/article/pii/S2665917422002392
6. https://ejmcm.com/article_2022.html
7. https://ieeexplore.ieee.org/document/9258470
8. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8968560/
9. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9180245/
10. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7525402/
11. https://iopscience.iop.org/article/10.1088/1742-6596/1282/1/012005
12. https://pubmed.ncbi.nlm.nih.gov/35294369/