Machine Learning For Early Diabetes Detection And Diagnosis With KNN

Tadiboyina Teja[1], PVRD Prasada Rao[2], Kuncham Sreenivasa Rao[3], K Sreerama murthy[4], P .Anil kumar[5], J. Balaraju[6]

[1,2]Koneru Lakshmaiah Education Foundation, Green fields,Vaddeswaram , A.P, India
[3]Faculty of Science and Technology (ICFAITech), ICFAI Foundation for Higher Education, Hyderabad
[4]Koneru Lakshmaiah Education Foundation, Hyderabad-500043, Telangana, India.
[5]Kommuri Pratap Reddy Institute of Technology, Hyderabad, India.
[6]School of Engineering, Anurag University, Hyderabad, India.
*pvrdprasad@kluniversity.in

Abstract Diabetes mellitus is a hastily developing international health problem that necessitates early detection and effective management to prevent severe complications. This study leverages machine learning, specifically the K-Nearest Neighbors algorithm, to predict and diagnose diabetes at an early stage. By analyzing a diverse dataset that includes biological, sociological, and clinical features, the studies goals to develop a robust predictive model. The application of KNN, alongside other machine learning techniques, permits for the advent of tools that can assess individual risk, enabling personalized remedy plans and optimizing healthcare management. The findings of this research could appreciably decorate early diabetes detection, leading to better patient outcomes and contributing to the fight against the diabetes epidemic. This study underscores the the capacity of machine gaining knowledge in transforming public health strategies and providing actionable insights for healthcare practitioners.

Keywords— Diabetes Prediction, Epidemic Control, Risk Assessment, Diabetes Mellitus, Early Detection, Machine Learning, Knn.

I. INTRODUCTION

Diabetes Mellitus, a chronic metabolic disease caused by high blood sugar, is affecting greater people worldwide than ever before. According to the International Diabetes Federation, there have been 463 million diabetic sufferers worldwide in 2019 if powerful measures aren't taken, this quantity is anticipated to increase to seven-hundred million patients by 2045. Diabetes is a major concern in society due to its potential complications such as heart disease, kidney failure, and blindness [1]. Danger. Machine learning is a part of artificial intelligence that advances science and prediction. Machine learning algorithms can examine huge information, discover hidden patterns, and predict outcomes based on data relationships. The primary intention of this approach is to develop correct predictive models that can discover people at excessive threat for diabetes. These models typically use different techniques, such as medical data (age, gender, family history), demographic data (BMI, waist circumference), and biochemical markers (blood sugar, lipid profile). Increasing use of threat evaluation and prediction. The capacity to process large, complex data, identify small differences, and create data-driven predictions is opening up new approaches to manage diseases like diabetes. We have the capacity to alternate the way we identify people at risk for diabetes using machine learning. We have to decorate the use advanced technology to enhance the diagnosis and management of diabetes, thereby improving health outcomes and reducing healthcare costs [2]. The methods, results, and implications of this examine is explored in more detail within the following sections, highlighting the future promise of machine gaining knowledge of in diabetes care [3].

II. K-NEAREST NEIGHBORS (KNN)

K-Nearest Neighbors is a truthful but effective system gaining knowledge of algorithm used for each e classification and also regression tasks [4]. The underlying precept is that similar instances exist in the close proximity within the feature space.

Non-Parametric Nature: KNN doesn't rely on assumptions about distribution of the data, making it adaptable to various datasets.

Lazy Learning: Unlike other algorithms that develop an explicit model during the training phase, KNN does not create a generalized model [5]. Instead, it stores all training data and performs calculations at the time of prediction.

A. Dataset Overview for Diabetes analysis

Pima Indians diabetes Dataset: This dataset is commonly used in studies associated with diabetes diagnosis. It contains several health-related variables that could assist are expecting the likelihood of diabetes, alongside a target variable indicating to the presence and the absence of the disease [6].

Features:

- Pregnancies: wide variety of instances the patient has been pregnant
- Glucose: Plasma glucose interest after 2 hours of an oral glucose tolerance check
- Blood Pressure: Diastolic blood pressure(measured in mm Hg)
- Skin Thickness: Triceps skinfold thickness (measured in mm)
- Insulin: Serum insulin (measured in mu U/ml)
- BMI: Body mass index calculated as weight in kilograms divided by height in meters squared
- Diabetes Pedigree Function: A score that estimates diabetes risk based on family history
- Age: Age of the patient (in years)
- Outcome: A binary variable indicating the presence (1) or absence (0) of diabetes.

B. Data Preprocessing

- Preprocessing is essential to put together the information for analysis and ensure that KNN performs effectively [7].
- Managing Missing Data:
- Imputation: Replace missing values with estimates, often calculated as the mean or median of the available data.
- Removing Records: alternatively, records with lacking data might be discarded, especially when the missing data comprises a small part of the datasets.
- Normalization/Standardization:
- Normalization: Rescale the information to fit within a range, typically [0, 1], which helps in treating all features equally when calculating distances [8].
- Standardization: Adjust the records in order that it has a mean of 0 and a standard deviation of 1. This method is mainly beneficial for data that approximates a Gaussian distribution.KNN is the highly sensitive to scale of data because of its reliance on distance metrics which includes Euclidean distance [9].

Dataset Splitting:

- Training Set: Generally comprises 70-85% of data, used to train a model.
- Testing Set: The remaining 15-25% is set aside to evaluating the model's predictive accuracy [10].

C. Implementing KNN for Diabetes Prediction

Selecting 'k':

Choosing the number of neighbors, 'k', is critical. A lower value of 'k' can make the model overly touchy to noise, even as a higher value may cause the model to smooth over important distinctions [11].

Cross-validation is often used to decide the optimal value of 'k'.

Distance Metrics:

Euclidean Distance: The most regularly used distance metric, calculated as the straight-line distance between points in the feature space.

Manhattan Distance: An alternative metric that measures distance because the sum of the absolute differences across dimensions.

Minkowski Distance: A general distance measure that encompasses both Euclidean and Manhattan distances, depending on the parameter [12].

Training the Model: In K-Nearest Neighbors (KNN), the model retains all training data. For predictions, it calculates the distance between a new observation and all training points, then selects the 'k' closest neighbors to make the prediction.

Making Predictions:

Classification: The predicted class label is decided by means of a majority vote among the 'k' nearest neighbors.

Regression: The predicted value is the average of the values from the 'k' nearest neighbors.

D. **Addressing Challenges in KNN**

Computational Demand: Since KNN computes distances for every prediction, it can be slow with large datasets.

Memory Requirements: KNN stores the whole training dataset that can be annoying in terms of memory, especially for large or high-dimensional data.

Optimal 'k' Selection: Determining the best value for 'k' can be challenging and typically involves empirical testing or cross-validation.

Outlier Sensitivity: KNN can be influenced by outliers, which may degrade its performance.

III. OBJECTIVE

These objectives collectively aim to enhance the application of the machine learning within the assessment and prediction of diabetes, ultimately leading to better healthcare outcomes for individuals at risk of this chronic condition [13].

A. Develop Precise Predictive Models: The number one objective is to design machine learning models capable of accurately predicting an individual's risk of developing diabetes. These models should integrate diverse input factors, including clinical, demographic, and biochemical data, to generate dependable risk assessments.
B. Identify Key Predictive Features: Utilize feature selection and engineering techniques to pinpoint the most tremendous threat factors for diabetes. this can allow healthcare experts to focus on the critical factors identified, aiding in early risk detection and intervention.
C. Compare Various Machine Learning Algorithms: Assess and examine the effectiveness of various machine learning algorithms, such as SVM, Logistic Regression, Random Forests, Decision Trees, and Deep Learning models. This comparison will help in selecting the maximum appropriate algorithm for diabetes prediction primarily primarily based at the traits of the dataset and overall performance metrics [14].
D. Evaluate Model Generalization: Make certain that the models developed can generalize well to new, unseen data. This involves testing their robustness and potential to make correct predictions outside of training dataset.
E. Enhance Model Interpretability: Work on improving the interpretability of system learning models to make them more accessible to healthcare professionals.Techniques inclusive

of characteristic significance analysis and visualization should be employed to help understand the elements contributing to the diabetes risk predictions [15].

F. Measure Prediction Performance**:** Use famous evaluation metrics like precision, recollect, accuracy, F1-score, and AUC-ROC to gauge the predictive success of the models. These metrics provide a complete view of the models' performance.

G. Contribute to Public Health: By figuring out people at excessive threat of growing diabetes, this objective facilitates early prevention and intervention strategies. The aim is to improve healthcare outcomes promptly through a mixture of medical and lifestyle changes.

H. Share Findings and Recommendations: Communicate the study's methods, results, and recommendations to the healthcare industry, policymakers, and the general public. Advocate for the usage of machine learning-based diabetes prediction as a precious tool for improving healthcare.

IV. LITERATURE REVIEW

a) The first study, authored by Alsulaiman and Alshurideh, offers an overview of large information frameworks and machine-learning algorithms utilized in healthcare, particularly for diabetes detection. It outlines various approaches and their relevance to diabetes research.

b) The third study by Rajalakshmi et al. explores a ramification of machine learning algorithms employed in assessing, managing, and diagnosing diabetes risk. The study discusses the strengths and the weaknesses of different approaches.

c) In their fifth paper, Gulshan et al. present a deep learning model designed to evaluate retinal images and detect diabetic retinopathy. This deep learning version outperformed skilled ophthalmologists, highlighting the critical function of the machine-learning in identifying diabetic eye situations which includes diabetic retinopathy and macular edema [16].

d) Alharthi et al.'s sixth paper makes a speciality of the usage of the machine-learning to increase diabetes of prediction models based on dietary patterns. The study emphasizes the significance of considering lifestyle and nutritional factors when predicting diabetes risk.

e) The sixth study by Koli and Patil applies machine learning strategies to forecast the progression of type two diabetes in patients. The researchers developed a model capable of identifying individuals at risk for complications, thereby enabling early intervention.

v. IMPORTANCE OF K-NEAREST NEIGHBORS IN DIABETES DIAGNOSIS

A. Simplicity and Accessibility

Easy to Understand: K-Nearest Neighbors is a straightforward algorithm, making it easy to implement and understand, even for people who are not experts in machine learning. This simplicity is essential in healthcare settings, where clear and transparent methods are preferred.

B. Accurate Classification

Effective for Medical Data: Within the context of diabetes diagnosis, KNN effectively classifies patients based on their similarities to others in the dataset. That is in particular crucial in medicine, where small differences in patient information could have large implications for diagnosis and treatment.

C. No Preset Model

Adapts to Data Complexity: KNN’s non-parametric nature means it doesn’t require a predetermined model or equation to describe the data. This is beneficial in diabetes diagnosis, where the relationship between variables may be complex and non-linear, allowing the algorithm to adapt to the actual data without constraints.

D. Personalized Diagnosis

Tailored Predictions: KNN offers personalized predictions by comparing a patient’s data with that of others who have similar medical profiles. This individualized approach is particularly important in diabetes diagnosis, where patient-specific factors play a critical role.

E. Flexibility and Customization

Adjustable Parameters: KNN allows for customization, such as choosing the number of neighbors ('k') or the distance metric used to measure similarity. This flexibility makes it possible to tailor the algorithm to fit the specific needs of the dataset or the healthcare environment in which it’s being applied.

F. Rapid Screening Tool

Quick Deployment: KNN can be quickly set up as an preliminary screening tool for figuring out potential diabetes cases. This is particularly valuable in clinical settings where rapid assessments are needed to flag high-risk patients for further testing or treatment.

G. Cost-Effective

Efficient for Small Datasets: In scenarios where the dataset is manageable in size, KNN operates efficiently, making it a cost-effective choice for clinics and healthcare facilities with limited computational resources.

H. Complementary to Other Models

Can Enhance Ensemble Methods: KNN can be used alongside other machine learning models as part of an ensemble approach, enhancing the general accuracy and the robustness of the diagnosis. By combining different models, the strengths of KNN can complement those of more complex algorithms [17].

Benchmark for Comparison: Due to its simplicity and reliability, KNN is frequently used as a benchmark to examine the overall performance of other, more sophisticated models. This allows researchers to higher understand the relative strengths of different approaches within the context of diabetes diagnosis.

VI. PERFORMANCE EVALUTION METHODS AND METRICS

To assess the effectiveness of system gaining knowledge of models for diabetes detection and analysis, and to make well-informed decisions about their implementation, it's essential to utilize appropriate performance metrics and techniques [18]. Below are key metrics and methods that may be completed in this context:

A. F1 Score: The F1 score is the harmonic suggest of precision and recall. It offers a balanced degree of these two metrics, making it mainly beneficial when each false positives and false negatives are of equal concern.

B. ROC-AUC: The ROC-AUC evaluates a model's capability to differentiate among advantageous and poor instances throughout one-of-a-kind chance thresholds.This metric is especially relevant for binary class responsibilities and situations involving imbalanced datasets [19].

C. Confusion Matrix: A confusion matrix affords a comprehensive breakdown. It forms the foundation for calculating metrics consisting of precision and recall.

D. Mean of Absolute Error (MAE) and Mean of Squared Error (MSE): Those metrics are regularly utilized in regression duties, wherein the goal is to expect a continuous variable, like blood glucose levels. MAE measures the common absolute distinction among expected and real values, on the identical time as MSE calculates the common squared distinction.

E. Accuracy: Accuracy is a typically used metric that shows the percentage of accurate predictions out of the total predictions made. But, it may be deceptive in scenarios involving imbalanced datasets.

F. Precision: Precision quantifies the ratio of true positive predictions to the entire range of positive predictions. It is critical for assessing the model's ability to reduce false positives, which is particularly important in contexts where false positives have serious consequences.

G. Recall (Sensitivity or True Positive Rate): Recall measures the percentage of real positive cases that the model correctly identifies. It is crucial in conditions wherein false negatives are costly, as it assesses the model's ability to capture all relevant instances.

H. ROC Curve: The ROC curve illustrates the trade-off between sensitivity and specificity across various probability thresholds [20]. The place beneath the curve (AUC) is a beneficial metric for evaluating distinctive models.

I. Model Interpretability: In healthcare applications like diabetes detection, understanding how a model makes predictions is crucial. Tools such as SHapley Additive exPlanations (SHAP) or Local Interpretable Model-Agnostic Explanations (LIME) may be used to benefit insights into the factors influencing the model's decisions.

VII. IMPLEMENTING STRATEGY

To successfully deploy an machine learning model, several critical steps want to be followed during the implementation of the strategy [21]. Here are the key actions to consider:

A. Choose the Appropriate Algorithm: Select the machine learning algorithm or algorithms which might be best desirable to address your specific problem. Consider factors such as the type of task (classification, regression, or clustering), the size of the dataset, and the computational resources available.

B. Data Collection and Preparation**:** Collect applicable information from numerous sources, ensuring that the data is accurate, organized, and pertinent to the problem you are trying to solve [22]. Perform data preprocessing, which may involve handling missing values, detecting outliers, and conducting feature engineering to create meaningful features for modeling. Divide the data into schooling, validation, and check sets for model development and evaluation.

C. Model Development: Use the training data to build, train, and optimize your machine learning model(s). Observe strategies like grid search or random search to fine-tune hyperparameters. Assess the model's performance using appropriate metrics and validation techniques.

D. Model Evaluation: Evaluate the model's performance on the validation dataset using relevant metrics (e.g., accuracy, F1-score, RMSE). Address any issues related to overfitting or underfitting by making necessary adjustments to the model.

E. Monitoring and Maintenance: Continuously monitor the model's performance in a real-world environment. Set up alerts for anomalies or drops in performance. Regularly retrain the model with new data to keep it up-to-date and ensure it continues to perform well.

VIII. METHODOLOGY

Implementing an automated learning approach for analyzing and predicting diabetes involves specific steps tailored to the project's objectives and the healthcare sector. Key actions to undertake include:

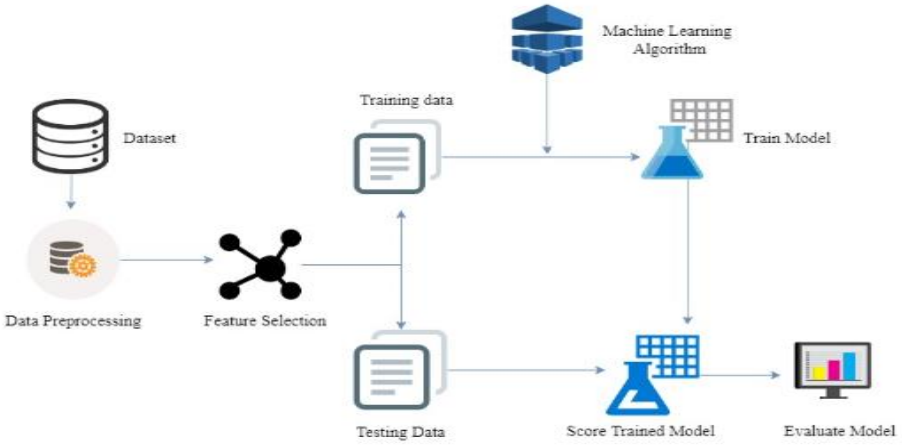

Fig. 1. Architectural Diagram

A. Data Collection and Preprocessing: Collect comprehensive medical records, consisting of affected person demographics, medical records, lab results (such as blood glucose levels), medication details, and lifestyle choices. Make certain compliance with information privateness regulations, such as HIPAA in the USA. Conduct data preprocessing to address problems like lacking values, outliers, and noise. Normalize or standardize numerical features as needed [23].

B. Feature Engineering: Derive meaningful features from raw data [24]. This could contain calculating metrics which incorporates frame mass index, insulin sensitivity indices, or diabetes risk scores. Incorporate domain-specific features that medical experts consider critical for diabetes prediction. Make sure that the dataset is balanced and representative, and split it into training, validation, and check sets.

C. Model Selection**:** Choose machine gaining knowledge of strategies which could accurately predict diabetes. Popular options include Support Vector Machines (SVMs), deep learning models such as neural network

D. Model Development: Develop and train the selected machine learning model(s) using the training dataset. Consider hyper parameter tuning to enhance model performance.

E. Model Evaluation: Evaluate the model's performance using relevant metrics like the F1 rating, place below ROC curve (AUC-ROC), precision, and recall. Address issues such as class imbalance using techniques like oversampling, under sampling, or cost-sensitive learning [25].

IX. RESULTS

```
<class 'pandas.core.frame.DataFrame'>
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
```

Fig. 2. Basic EDA and sttistical Analysis

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

Fig. 3. Describing data set

```
diabetes_data_copy = diabetes_data.copy(deep = True)
diabetes_data_copy[['Glucose','BloodPressure','SkinThickness','Insulin','BMI']] = diabetes_data_copy[['Glucose','BloodPressure','SkinThickness','Insulin','BMI']].replace(0,np.NaN)print(diabetes_data_copy.isnull().sum())
```

```
Pregnancies                   0
Glucose                       5
BloodPressure                35
SkinThickness               227
Insulin                     374
BMI                          11
DiabetesPedigreeFunction      0
Age                           0
Outcome                       0
dtype: int64
```

Fig. 4. Replacing the 0 or missing values with NaN

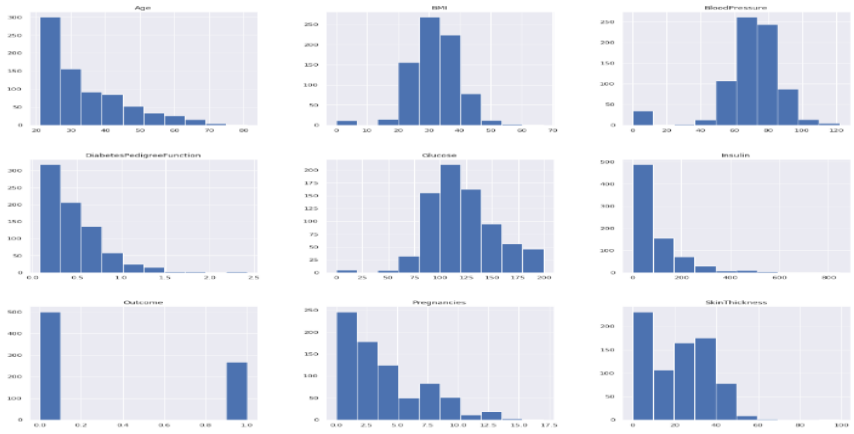

Fig. 4.1. To filling the NaN values and also plotting

diabetes_data_copy['Glucose'].fillna(value=diabetes_data_copy['Glucose'].mean(), inplace=True)

diabetes_data_copy['BloodPressure'].fillna(value=diabetes_data_copy['BloodPressure'].mean(), inplace=True)

diabetes_data_copy['SkinThickness'].fillna(value=diabetes_data_copy['SkinThickness'].median(), inplace=True)

diabetes_data_copy['Insulin'].fillna(value=diabetes_data_copy['Insulin'].median(), inplace=True)

diabetes_data_copy['BMI'].fillna(value=diabetes_data_copy['BMI'].median(), inplace=True)

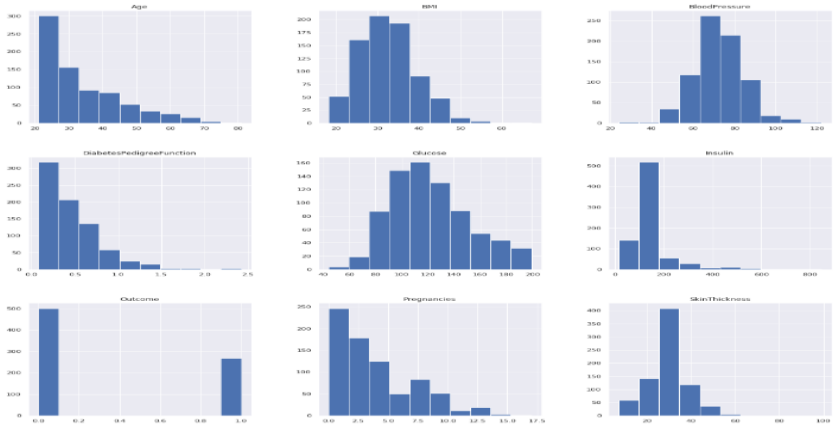

Fig. 4.2. Inputing the values of Nan for columns based totally at the distribution

p = diabetes_data_copy.hist(figsize = (20,20))

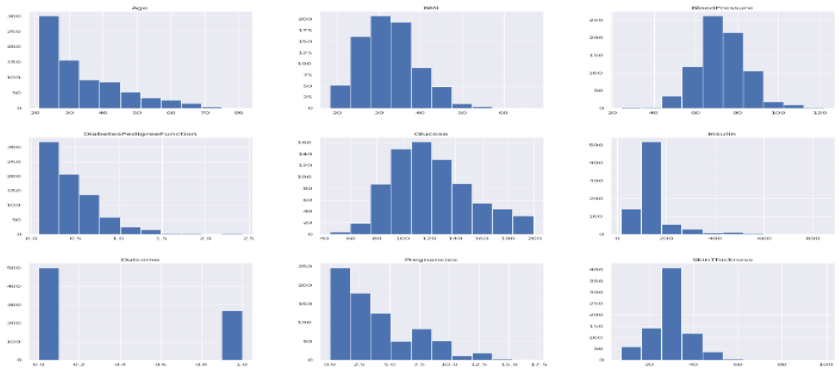

Fig. 4.3 NaN removel Plotting after

A) Skewness:

A left-skewed distribution is characterized by a long tail on the left side. This type of distribution, also known as negatively skewed, has a significant negative tail on the number line, and its peak is positioned to right of the mean.In contrast, a right-skewed distribution features a long tail on the right side. This positive-skew distribution has a substantial positive tail on the number line, with its peak located to the left of the mean. If we check the shape of the dataset using diabetes_data.shape, it will display dimensions such as (768, 9).

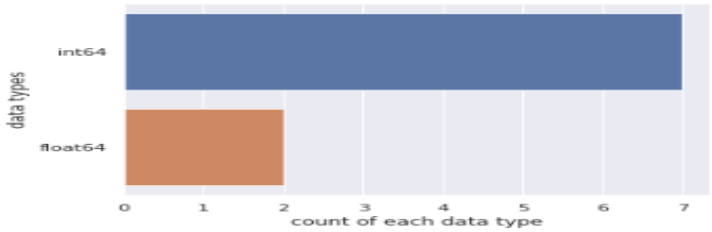

Fig. 5. Plotting data types

B) Null count analysis:import missingno as msno p=msno.bar(diabetes_data)

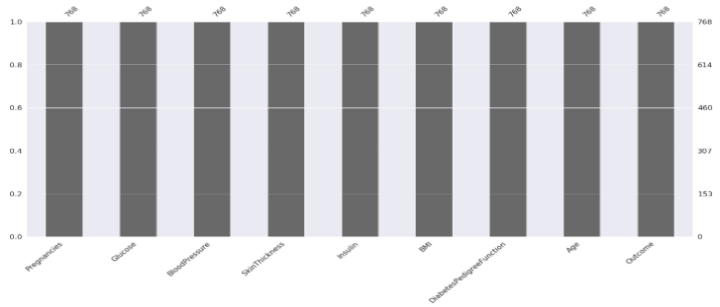

Fig. 6. Null count breakdown

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

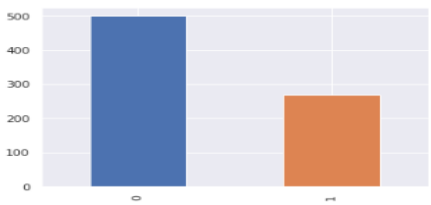

Fig. 7. Display equality of data types

C) Scatter Matrix:

```
from pandas.tools.plotting import scatter_matrix
p=scatter_matrix(diabetes_data,figsize=(25, 25))
```

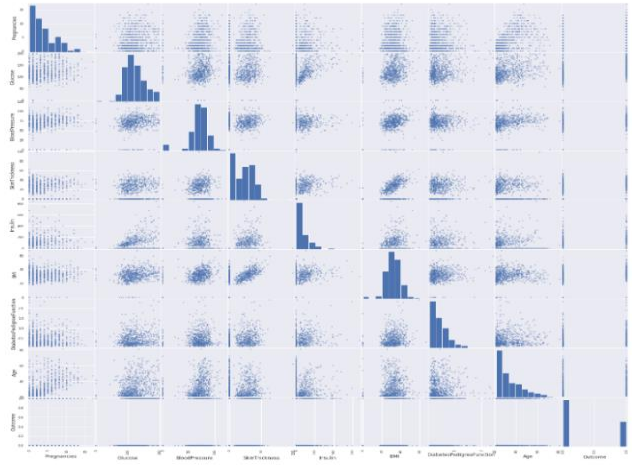

Fig. 8. Scatter matric for uncleaned data

Fig. 9. Paired charts for clean data

D) Pair Plot: A pair plot is created using Seaborn with the hue='Outcome' parameter to visualize relationships between variables in the diabetes_data_copy dataset. Pearson's correlation coefficient helps determine the strength and direction of the linear relationship among two variables. This coefficient ranges from -1 to +1, where +1 indicates a perfect positive correlation, -1 indicates a perfect negative correlation, and 0 means no correlation.

A heatmap is a graphical representation where data is shown through varying colors, making it easier to visualize both simple and complex information.

E) Heat Map:

```
plt.figure(figsize=(12, 10))
```

heatmap = sns.heatmap(diabetes_data_copy.corr(), annot=True, cmap='RdYlGn')

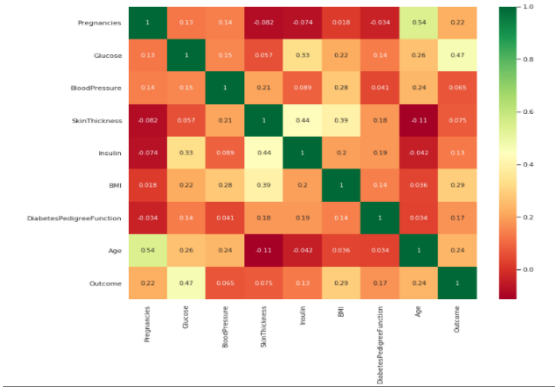

Fig. 10.1. Unclean data

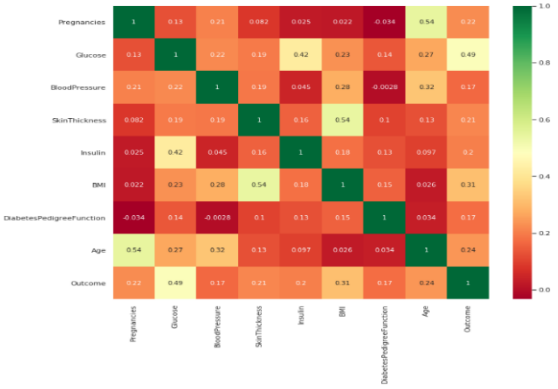

Fig. 10.2. Clean data

plt.figure(figsize=(12,10))

A heatmap is created using Seaborn with the correlation matrix of diabetes_data_copy, where the annot=True parameter displays the correlation values on the map, and cmap='RdYlGn' specifies the color scheme.

F) Data Scaling: The data, denoted as ZZZ, is normalized to have a mean μ=0\mu = 0μ=0 and a standard deviation σ=1\sigma = 1σ=1. This normalization is performed using the specified formula.

$$z = \frac{x_i - \mu}{\sigma}$$

from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()

scaled_features = scaler.fit_transform(diabetes_data_copy.drop(["Outcome"], axis=1))

X = pd.DataFrame(scaled_features, columns=['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI', 'DiabetesPedigreeFunction', 'Age'])

X.head()

Y = diabetes_data_copy.Outcome

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age
0	0.639947	0.865108	-0.033518	0.670643	-0.181541	0.166619	0.468492	1.425995
1	-0.844885	-1.206162	-0.529859	-0.012301	-0.181541	-0.852200	-0.365061	-0.190672
2	1.233880	2.015813	-0.695306	-0.012301	-0.181541	-1.332500	0.604397	-0.105584
3	-0.844885	-1.074652	-0.529859	-0.695245	-0.540642	-0.633881	-0.920763	-1.041549
4	-1.141852	0.503458	-2.680669	0.670643	0.316566	1.549303	5.484909	-0.020496

Fig.11. Scaling

G) Testing Training Split and Cross Validation Methods:

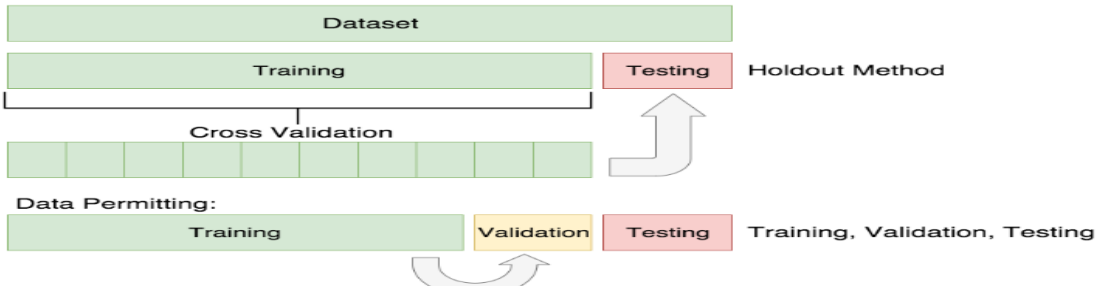

Fig. 12. Flow Chart

```python
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier

# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=1/3, random_state=42,
    stratify=y)

# Lists to store the accuracy scores for the training and testing sets
test_scores = []
train_scores = []

# Loop through a range of k values to find the best k
for i in range(1, 15):
    knn = KNeighborsClassifier(i)
    knn.fit(X_train, y_train)
    train_scores.append(knn.score(X_train, y_train))
    test_scores.append(knn.score(X_test, y_test))

# Find the maximum training accuracy and the corresponding k value(s)
max_train_score = max(train_scores)
train_scores_ind = [i for i, v in enumerate(train_scores) if v == max_train_score]
print('Highest training accuracy: {}% with k = {}'.format(max_train_score*100, list(map
    (lambda x: x+1, train_scores_ind))))

# Find the maximum testing accuracy and the corresponding k value(s)
max_test_score = max(test_scores)
test_scores_ind = [i for i, v in enumerate(test_scores) if v == max_test_score]
print('Highest testing accuracy: {}% with k = {}'.format(max_test_score*100, list(map(lambda
    x: x+1, test_scores_ind))))
```

Max test score: 0.785735

Fig. 12.1. Python code

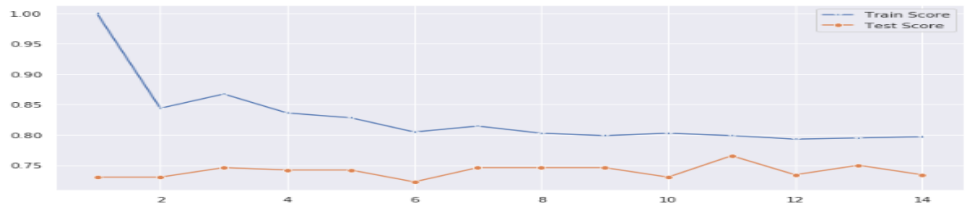

Fig. 12. 2 Visualization

The optimal outcome is achieved with k=11k = 11k=11, so this value is used for the final model. The KNeighborsClassifier is initialized with k=11k = 11k=11 and then trained using knn.fit (X_train, y_train). Ultimately, the overall performance of the version is evaluated with knn.score (X_test, y_test).

- 0.785735

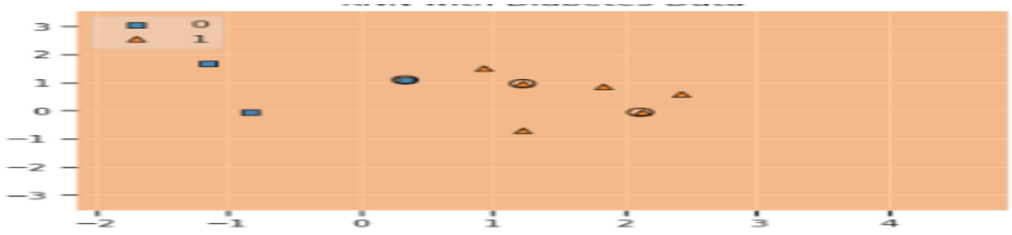

Fig. 13. KNN with Diabetes Data

H) Confusion Matrix:

n=165	Predicted: NO	Predicted: YES	
Actual: NO	TN = 50	FP = 10	60
Actual: YES	FN = 5	TP = 100	105
	55	110	

Fig.14.1. Data

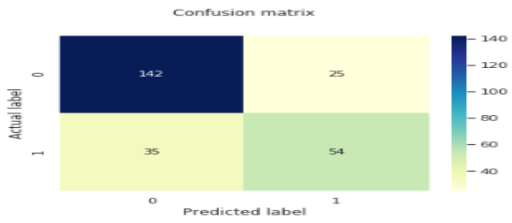

Fig.14.2. Confusion Matrix

X. DISCUSSION:

The K-nearest neighbours (KNN) approach works well when the dataset is small, the data distribution is non-linear or irregular, and interpretability is important. Its non-parametric nature makes it simple to use and understand, making it ideal for rapid prototyping and exploratory research. KNN, unlike many different algorithms, does no longer require specific training periods; rather, it predicts from the full dataset, making it ideal for online or streaming data. KNN is immune to outliers and can handle noisy data effectively since it predicts based on the majority class or nearest neighbours. Its lazy learning strategy allows for adaption to changing datasets without requiring frequent retraining. However, KNN's efficiency decreases with large datasets due to its instance-based nature, resulting in high computing costs during prediction and storage requirements for the full dataset. Furthermore, its performance is heavily impacted by the hyper parameter k (the number of neighbours) and may suffer in a high dimensional environments due to curse of the dimensionality. As a result, while KNN is simple and adaptable, its performance varies relying at the dataset's characteristics and the analytic objectives.

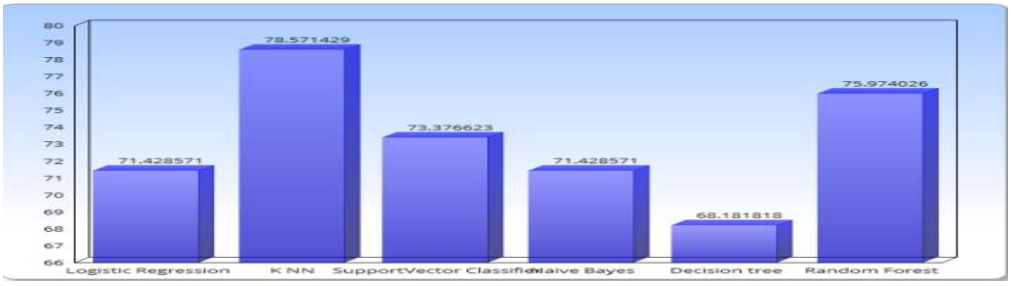

Fig. 15. Contrast between different methodologies

XI. CONCLUSION

In summary, Applying machine learning for early diabetes detection is crucial in addressing the global diabetes epidemic. These advanced technologies enable swift responses, improving patient outcomes and reducing the financial and personal burdens linked to diabetes-related complications. By utilizing diverse datasets and sophisticated algorithms, these tools provide accurate risk assessments. As technology continues to evolve, the potential for early diabetes detection holds great promise in reshaping preventive healthcare. In the near future, the most effective diabetes analysis will attention on predicting the disease and raising awareness through the usage of neural network technology.

REFERENCES

[1] F. A. Khan, K. Zeb, M. Al-Rakhami, A. Derhab and S. A. C. Bukhari, "Detection and Prediction of Diabetes Using Data Mining: A Comprehensive Review," in IEEE Access, vol. 9, pp. 43711-43735, 2021, doi: 10.1109/ACCESS.2021.3059343.

[2] M. Chen, J. Yang, J. Zhou, Y. Hao, J. Zhang and C. -H. Youn, "5G-Smart Diabetes: Toward Personalized Diabetes Diagnosis with Healthcare Big Data Clouds," in IEEE Communications Magazine, vol. 56, no. 4, pp. 16-23, April 2018, doi: 10.1109/MCOM.2018.1700788.

[3] H. Yin, B. Mukadam, X. Dai and N. K. Jha, "DiabDeep: Pervasive Diabetes Diagnosis Based on Wearable Medical Sensors and Efficient Neural Networks," in IEEE Transactions on Emerging Topics in Computing, vol. 9, no. 3, pp. 1139-1150, 1 July-Sept. 2021, doi: 10.1109/TETC.2019.2958946.

[4] U. Ahmed et al., "Prediction of Diabetes Empowered With Fused Machine Learning," in IEEE Access, vol. 10, pp. 8529-8538, 2022, doi: 10.1109/ACCESS.2022.3142097.

[5] N. Barakat, A. P. Bradley and M. N. H. Barakat, "Intelligible Support Vector Machines for Diagnosis of Diabetes Mellitus," in IEEE Transactions on Information Technology in Biomedicine, vol. 14, no. 4, pp. 1114-1120, July 2010, doi: 10.1109/TITB.2009.2039485.

[6] Y. Sun and D. Zhang, "Diagnosis and Analysis of Diabetic Retinopathy Based on Electronic Health Records," in IEEE Access, vol. 7, pp. 86115-86120, 2019, doi: 10.1109/ACCESS.2019.2918625.

[7] E. Gomes Filho, P. R. Pinheiro, M. C. D. Pinheiro, L. C. Nunes and L. B. G. Gomes, "Heterogeneous Methodology to Support the Early Diagnosis of Gestational Diabetes," in IEEE Access, vol. 7, pp. 67190-67199, 2019, doi: 10.1109/ACCESS.2019.2903691.

[8] K. Zarkogianni et al., "A Review of Emerging Technologies for the Management of Diabetes Mellitus," in IEEE Transactions on Biomedical Engineering, vol. 62, no. 12, pp. 2735-2749, Dec. 2015, doi: 10.1109/TBME.2015.2470521.

[9] J. Tulloch, R. Zamani and M. Akrami, "Machine Learning in the Prevention, Diagnosis and Management of Diabetic Foot Ulcers: A Systematic Review," in IEEE Access, vol. 8, pp. 198977-199000, 2020, doi: 10.1109/ACCESS.2020.3035327.

[10] L. Qiao, Y. Zhu and H. Zhou, "Diabetic Retinopathy Detection Using Prognosis of Microaneurysm and Early Diagnosis System for Non-Proliferative Diabetic Retinopathy Based on Deep Learning Algorithms," in IEEE Access, vol. 8, pp. 104292-104302, 2020, doi: 10.1109/ACCESS.2020.2993937.

[11] K. Khalfallah et al., "Noninvasive Galvanic Skin Sensor for Early Diagnosis of Sudomotor Dysfunction: Application to Diabetes," in IEEE Sensors Journal, vol. 12, no. 3, pp. 456-463, March 2012, doi: 10.1109/JSEN.2010.2103308.

[12] N. L. Fitriyani, M. Syafrudin, G. Alfian and J. Rhee, "Development of Disease Prediction Model Based on Ensemble Learning Approach for Diabetes and Hypertension," in IEEE Access, vol. 7, pp. 144777-144789, 2019, doi: 10.1109/ACCESS.2019.2945129.

[13] S. Samreen, "Memory-Efficient, Accurate and Early Diagnosis of Diabetes Through a Machine Learning Pipeline Employing Crow Search-Based Feature Engineering and a Stacking Ensemble," in IEEE Access, vol. 9, pp. 134335-134354, 2021, doi: 10.1109/ACCESS.2021.3116383.

[14] R. Ferdousi, M. A. Hossain and A. E. Saddik, "Early-Stage Risk Prediction of Non-Communicable Disease Using Machine Learning in Health CPS," in IEEE Access, vol. 9, pp. 96823-96837, 2021, doi: 10.1109/ACCESS.2021.3094063.

[15] D. Sopic, A. Aminifar, A. Aminifar and D. Atienza, "Real-Time Event-Driven Classification Technique for Early Detection and Prevention of Myocardial Infarction on Wearable Systems," in IEEE Transactions on Biomedical Circuits and Systems, vol. 12, no. 5, pp. 982-992, Oct. 2018, doi: 10.1109/TBCAS.2018.2848477.

[16] T. Zhu, K. Li, P. Herrero and P. Georgiou, "Deep Learning for Diabetes: A Systematic Review," in IEEE Journal of Biomedical and Health Informatics, vol. 25, no. 7, pp. 2744-2757, July 2021, doi: 10.1109/JBHI.2020.3040225.

[17] D. Sierra-Sosa et al., "Scalable Healthcare Assessment for Diabetic Patients Using Deep Learning on Multiple GPUs," in IEEE Transactions on Industrial Informatics, vol. 15, no. 10, pp. 5682-5689, Oct. 2019, doi: 10.1109/TII.2019.2919168.

[18] Z. Gao, J. Li, J. Guo, Y. Chen, Z. Yi and J. Zhong, "Diagnosis of Diabetic Retinopathy Using Deep Neural Networks," in IEEE Access, vol. 7, pp. 3360-3370, 2019, doi: 10.1109/ACCESS.2018.2888639.

[19] M. S. Islam, M. K. Qaraqe, S. B. Belhaouari and M. A. Abdul-Ghani, "Advanced Techniques for Predicting the Future Progression of Type 2 Diabetes," in IEEE Access, vol. 8, pp. 120537-120547, 2020, doi: 10.1109/ACCESS.2020.3005540.

[20] Q. Wang, W. Cao, J. Guo, J. Ren, Y. Cheng and D. N. Davis, "DMP_MI: An Effective Diabetes Mellitus Classification Algorithm on Imbalanced Data With Missing Values," in IEEE Access, vol. 7, pp. 102232-102238, 2019, doi: 10.1109/ACCESS.2019.2929866.

[21] A. Nishanth and T. Thiruvaran, "Identifying Important Attributes for Early Detection of Chronic Kidney Disease," in IEEE Reviews in Biomedical Engineering, vol. 11, pp. 208-216, 2018, doi: 10.1109/RBME.2017.2787480.

[22] E. M. Moreno et al., "Type 2 Diabetes Screening Test by Means of a Pulse Oximeter," in IEEE Transactions on Biomedical Engineering, vol. 64, no. 2, pp. 341-351, Feb. 2017, doi: 10.1109/TBME.2016.2554661.

[23] B. Zhang, B. V. K. Vijaya Kumar and D. Zhang, "Detecting Diabetes Mellitus and Nonproliferative Diabetic Retinopathy Using Tongue Color, Texture, and Geometry Features," in IEEE Transactions on Biomedical Engineering, vol. 61, no. 2, pp. 491-501, Feb. 2014, doi: 10.1109/TBME.2013.2282625.

[24] N. Bhaskar, V. Bairagi, E. Boonchieng and M. V. Munot, "Automated Detection of Diabetes From Exhaled Human Breath Using Deep Hybrid Architecture," in IEEE Access, vol. 11, pp. 51712-51722, 2023, doi: 10.1109/ACCESS.2023.3278278.

[25] M. Bernardini, L. Romeo, P. Misericordia and E. Frontoni, "Discovering the Type 2 Diabetes in Electronic Health Records Using the Sparse Balanced Support Vector Machine," in IEEE Journal of Biomedical and Health Informatics, vol. 24, no. 1, pp. 235-246, Jan. 2020, doi: 10.1109/JBHI.2019.2899218.