International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-13 Issue-9, August 2024

Comparative Study of Machine Learning Based Diabetes Predictive System

Ratna Kumari Challa, Buduri Reddaiah, Kanusu Srinivasa Rao, Krishnaiah Pulluru, Ranga Swamy Sirisati, Venkata Narayana Reddy

Abstract*: Diabetes is one of the most lethal diseases in the world. It is also a precursor to various other disorders such as coronary failure, blindness, and kidney diseases. Patients often need to visit diagnostic centers to get their reports after consultation, which requires a significant investment of time and money. However, with the growth of machine learning methods, we now have the ability to address this issue. Advanced systems utilizing information processing can forecast whether a patient has diabetes or not. Furthermore, early prediction of the disease can provide patients with critical interventions before it fully develops. Data mining techniques can extract hidden information from large datasets of diabetes-related information. The aim of this research is to develop a system that can predict the diabetic risk level of a patient with higher accuracy. The model development is based on classification methods such as K-Nearest Neighbors, Decision Tree, and Support Vector Machine (SVM) algorithms. For K-Nearest Neighbors, the models achieve an accuracy of 71%, 78% for SVM, and 70% for the Decision Tree algorithm. The outcomes demonstrate a significant accuracy of these methods.

Keywords: Diabetes, SVM, KNN, Decision tree, Accuracy

I. INTRODUCTION

Diabetes is a condition characterized by a deficiency of insulin in the blood. Symptoms of high blood sugar include frequent urination, increased thirst, and heightened hunger. If left untreated, it can lead to severe complications and even death. An elevated blood sugar level is referred to as prediabetes. The effectiveness of a decision support system is determined by its accuracy. Therefore, the objective is to build a decision support system to predict and diagnose a certain disease with extreme amount of precision.

Manuscript received on 15 July 2024 | Revised Manuscript received on 20 July 2024 | Manuscript Accepted on 15 August 2024 | Manuscript published on 30 August 2024.
*Correspondence Author(s)

Ratna Kumari Challa, Department of Computer Science and Engineering, AP-IIIT, RGUKT, RK Valley, Idupulapaya, Kadapa (A.P.), India, Email: ratnamala3784@gmail.com, ORCID ID: 0000-0001-5077-8513

Buduri Reddaiah, Department of Computer Science and Technology, Yogi Vemana University, Kadapa (A.P.), India, Email: prof.reddaiah@yvu.edu.in, ORCID ID: 0000-0002-5851-2194

Kanusu Srinivasa Rao*, Department of Computer Science and Technology, Yogi Vemana University, Kadapa (A.P.), India, Email: kanususrinivas@gmail.com, ORCID ID: 0000-0002-9850-3110

Krishnaiah Pulluru, Department of Computer Science and Technology, Yogi Vemana University, Kadapa (A.P.), India, Email: krishna35sku@gmail.com

Ranga Swamy Sirisati, Department of Computer Science & Engineering, Vignan's Institute of Management and Technology for Women, Kondapur, Ghatkesar (Telangana), Email: sirisatiranga@gmail.com.

Venkata Narayana Reddy, Department of Computer Science and Technology, Yogi Vemana University, Kadapa (A.P.), India, Email: cnaremca@gmail.com.

II. BACKGROUND STUDY

A. Diabetes Predictive System

A diabetes prediction system is a highly useful tool in the healthcare field. This paper proposes an accurate system for diabetes prediction. The predictive system takes input from the patient and determines whether the patient has diabetes based on machine learning algorithm [9][10]. Through experiments, we can conclude which algorithm provides the most accurate results for predicting diabetes. In this paper, three algorithm: Support Vector Machine (SVM), K-Nearest Neighbors (KNN), and Decision Tree - are used to predict diabetes, as these algorithms provide better accuracy compared to others.

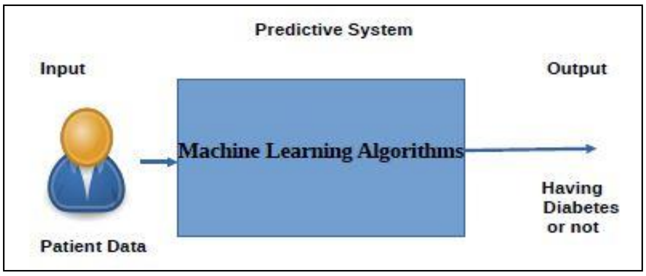


Fig.1. General Predictive System

B. Motivation

There has been a drastic increase in the rate of people suffering from diabetes over the past decade. The current human lifestyle is the main reason behind this growth in diabetes. In current medical diagnosis methods, there can be three different types of errors:-

1. False-negative: In this type, a patient who is actually diabetic receives test results indicating that they do not have diabetes.
2. False-positive: In this type, a patient who is not diabetic receives test results indicating that they have diabetes.
3. Unclassifiable: In this type, the system cannot diagnose a given case. This happens due to insufficient knowledge extraction from past data, resulting in a patient's condition being predicted as unclassified.

However, in reality, patients need to be accurately categorized as either diabetic or non-diabetic. Errors in diagnosis can lead to unnecessary treatments or neglect when treatment is necessary. To mitigate these impacts, there is a pressing need to develop a system using machine learning algorithms and data mining techniques that can provide precise results and minimize human effort.

III. LITERATURE SURVEY

A new data preparation method based on clustering algorithms for the diagnosis systems of heart and diabetes diseases has been proposed [10][11][12]. The most important factors that hinder pattern recognition from functioning rapidly and effectively are noisy and inconsistent data in databases. Existing research has explored various approaches for diabetes detection, including data mining techniques such as clustering [6] and classification. Algorithms like k-Nearest Neighbor (k-NN) [1], k-means, and branch and bound have been proposed for diabetes prediction. Comparative analysis has been conducted using a basic diabetic dataset. The importance of feature analysis in predicting diabetes [7] using machine learning techniques is also discussed. A significant issue in existing systems is high false positives. This proposed system aims to address these challenges through the following machine learning algorithms.

A. SVM: Support Vector Machine (SVM) is a supervised machine learning algorithm used for both classification and regression. The objective of the SVM algorithm is to find a hyperplane in an N-dimensional space that distinctly classifies the data points [1][2].

B. KNN: K-Nearest Neighbors (KNN) works by finding the distances between a query and all the examples in the data, selecting the specified number of examples (K) closest to the query, and then voting for the most frequent label (in clastion) or averaging the labels [5][6].

C. Decision Tree: A decision tree is a decision support tool that uses a tree-like model of decisions and their potential consequences, including the outcomes of chance events [3][4].

IV. PROPOSED SYSTEM

The proposed study focuses on the classification of the Indian PIMA dataset for diabetes as a binary classification problem. In this system, the model is trained using a dataset. Preprocessing steps are employed to handle null values and eliminate irrelevant data that may not contribute to the predictive system [14].

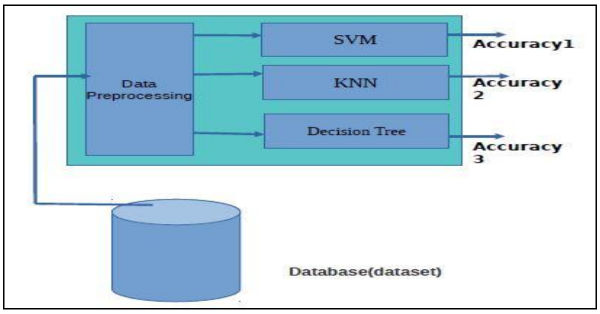


Fig.2. Diabetes Predictive System

The proposed system follows the following steps:

- Data Collection
- Data Preprocessing
- Data Analysis
- Standardizing the Data

A.Data Collection

Before collecting the data, we need to import all the libraries that we are going to use in this model.

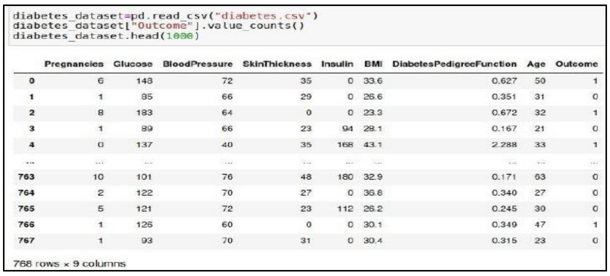
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn import svm #support vector machine
from sklearn.metrics import accuracy_score
import seaborn as sns
```

Then, we collect the data. We have already downloaded the data from the Kaggle dataset [8] and load it into our system using pandas. This dataset contains the following attributes:

Pregnancies: Number of pregnancies
Glucose: Glucose levels of the person
Blood Pressure: Blood pressure of the person
Skin Thickness: Skin thickness of the person
Insulin: Insulin levels of the person
BMI: Body Mass Index (BMI) of the person
Diabetes Pedigree Function: Diabetes pedigree function of the person
Age: Age of the person
Outcome: Output (indicating sificapresence or absence of diabetes)

B. Data Preprocessing

This phase of the model addresses inconsistent data to achieve more accurate and precise results. The dataset contains missing values, so we imputed missing values for selected attributes such as Glucose level, Blood Pressure, Skin Thickness, BMI, and Age, since these attributes cannot logically have values of zero. After imputation, the dataset is scaled to normalize all values.



```
diabetes_dataset=pd.read_csv("diabetes.csv")
diabetes_dataset["Outcome"].value_counts()
diabetes_dataset.head(1000)
```

	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
...	...	...	...	...	...	...	...	...	...
763	10	101	76	48	180	32.9	0.171	63	0
764	2	122	70	27	0	36.8	0.340	27	0
765	5	121	72	23	112	26.2	0.245	30	0
766	1	126	60	0	0	30.1	0.349	47	1
767	1	93	70	31	0	30.4	0.315	23	0

768 rows × 9 columns

Fig.3. Sample Data Set

C.Data Analysis

In this step, we analyze the data to build a model that predicts whether a person has diabetes or not. Now we plot the correlation matrix heatmap to analyze which factors are most correlated with our outcome. Here, 0 indicates Diabetes and 1 indicates no Diabetes.

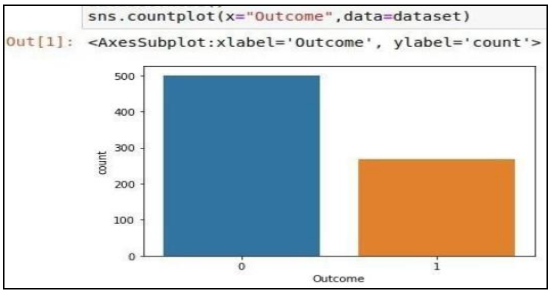


Fig.4. Plotting of Outcomes Columns

Retrieval Number: 100.1/ijitee.I995213090824
DOI: 10.35940/ijitee.I9952.13090824
Journal Website: www.ijitee.org

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

D. Standardizing the Data

Here, we designate the labels as x and y axes. On the x-axis, we include all values except the outcome, which is placed on the y-axis.

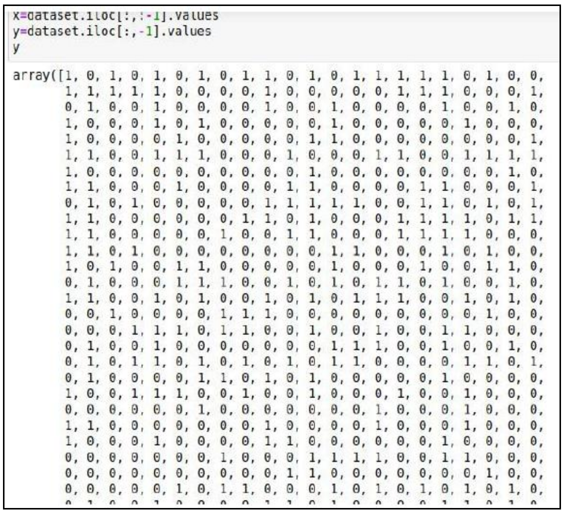


Fig.5. Y Labels

```
x=dataset.iloc[:,:-1].values
y=dataset.iloc[:,-1].values
x

array([[  6.   , 148.   ,  72.   , ...,  33.6  ,   0.627,  50.   ],
       [  1.   ,  85.   ,  66.   , ...,  26.6  ,   0.351,  31.   ],
       [  8.   , 183.   ,  64.   , ...,  23.3  ,   0.672,  32.   ],
       ...,
       [  5.   , 121.   ,  72.   , ...,  26.2  ,   0.245,  30.   ],
       [  1.   , 126.   ,  60.   , ...,  30.1  ,   0.349,  47.   ],
       [  1.   ,  93.   ,  70.   , ...,  30.4  ,   0.315,  23.   ]])
```

Fig.6. X Labels

```
In [24]: #splitting the dataset to test and train
         from sklearn.model_selection import train_test_split
         x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=0)
         x_train.shape
Out[24]: (614, 8)
```

Fig.7. Splitting the Dataset to Train and Test

E. Model Building

From here, we are going to build a model. For feature scaling, we use Standard Scalar, which scales the features and predicts values for new data.

```
#feature scaling
from sklearn.preprocessing import StandardScaler
sc=StandardScaler()
x_train=sc.fit_transform(x_train)
x_test=sc.transform(x_test)
```

For modeling, three classification algorithms are used.

a. K-neighbors Algorithm

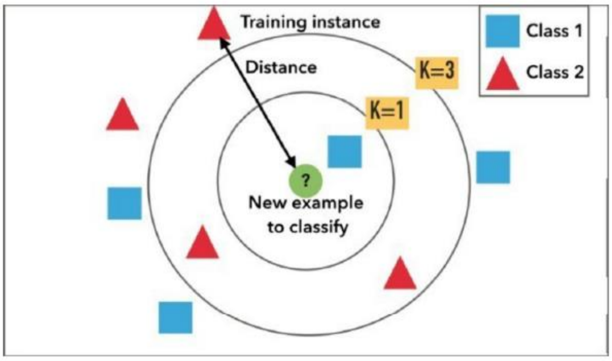


Fig.8. KNN Classification

KNN works by calculating the distances between a query and all examples in the data, selecting the specified number of examples (K) closest to the query. It then assigns the most frequent label among these neighbours as the prediction (in classification tasks) [5][6].

```
#model building
from sklearn.neighbors import KNeighborsClassifier
knn=KNeighborsClassifier(n_neighbors=25,metric="minkowski")
knn.fit(x_train,y_train)

KNeighborsClassifier(n_neighbors=25)
```

We train our data using the KNN algorithm with n_neighbors set to 25, which was determined using least squares. Then, we predict the Y values using the algorithm.

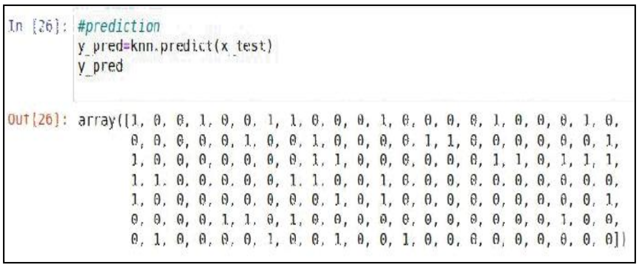


Fig.9. Predict the Y Values with KNN Algorithm

```
In [16]: #confusion matrix
         from sklearn.metrics import confusion_matrix
         cm=confusion_matrix(y_test,y_pred)
         sns.heatmap(cm,annot=True)
Out[16]: <AxesSubplot:>
```

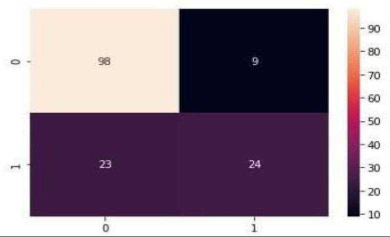


Fig.10. Confusion Matrix

```
In [17]: #accuracy
         from sklearn.metrics import accuracy_score
         accuracy_score(y_test,y_pred)

Out[17]: 0.7922077922077922
```

Fig.11. Accuracy with that Algorithm

b. Support Vector Machine Algorithm:

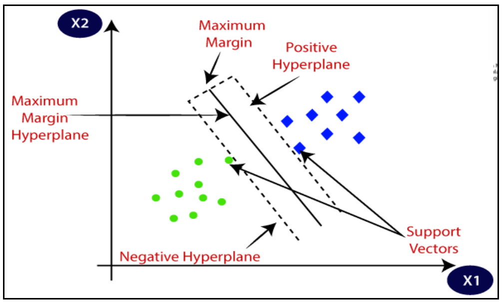


Fig.12. SVM Model

Support vectors are data points that are closer to the hyper plane and influence the position and orientation of the hyper plane.

Retrieval Number: 100.1/ijitee.I995213090824
DOI: 10.35940/ijitee.I9952.13090824
Journal Website: www.ijitee.org

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

By using these support vectors, SVM aims to maximize the margin of the classifier. Removing support vectors may alter the position of the hyper plane. These points are crucial for constructing our SVM model.

```
#training the model
classifier=svm.SVC(kernel="linear")

#training the support vector Machine classifier
classifier.fit(X_train,Y_train)

SVC(kernel='linear')

#model evaluation
#accuracy score on training data
X_train_prediction=classifier.predict(X_train)
training_data_accuracy=accuracy_score(X_train_prediction,Y_train)
print("accuracy of training data=",training_data_accuracy)

accuracy of training data= 0.7866449511400652
```

The classifier using the Support Vector Machine (SVM) algorithm classifies our data and is used to predict with our trained model. It achieves an accuracy score of approximately 0.78.

c. Decision Tree

A decision tree is a decision support tool that utilizes a tree-like model of decisions and their potential consequences, including chance event outcomes, resource costs, and utility. It represents an algorithm containing conditional control statements in a graphical form.

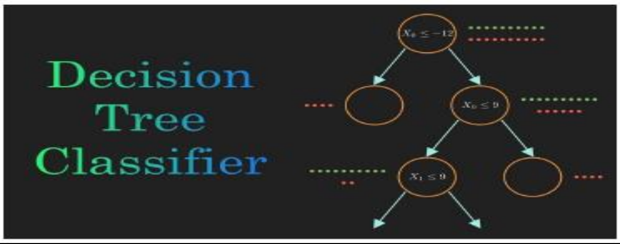


Fig.13. Decision Tree Model

We train our data using the Decision Tree algorithm. Then, we predict the Y values using the algorithm.

```
from sklearn.tree import DecisionTreeClassifier

dtree = DecisionTreeClassifier()
dtree.fit(X_train, Y_train)
```

Fig.14. Decision Tree Classifier

```
from sklearn import metrics

predictions = dtree.predict(X_test)
print("Accuracy Score =", format(metrics.accuracy_score(Y_test,predictions)))

Accuracy Score = 0.6883116883116883
```

Fig.15. Predict the Y Values with Decision Tree Algorithm

```
              precision    recall  f1-score   support

           0       0.74      0.81      0.77       100
           1       0.57      0.46      0.51        54

    accuracy                           0.69       154
   macro avg       0.65      0.64      0.64       154
weighted avg       0.68      0.69      0.68       154

[[81 19]
 [29 25]]
```

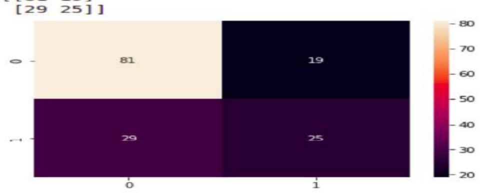


Fig.16. Classification Report and Confusion Matrix of the Decision Tree Model

It gives nearly 0.688 Accuracy score.

```
from sklearn.metrics import classification_report, confusion_matrix

print(classification_report(Y_test,predictions))
cm_dc=confusion_matrix(Y_test,predictions)
sns.heatmap(cm_dc,annot=True)
print(confusion_matrix(Y_test, predictions))
```

```
In [14]: #making predictive system
input_data=(0,66,9,29,0,26.6,0,31)
input_data_as_numpy_array=np.asarray(input_data)
input_data_reshaped=input_data_as_numpy_array.reshape(1,-1)
#standardized the input data
std_data=scaler.transform(input_data_reshaped)
print(std_data)
prediction=classifier.predict(std_data)
print(prediction)
if(prediction[0]==0):
    print("no diabetics")
else:
    print("yes diabetics")

[[-1.14185152 -1.71804212 -3.10731749  0.53090156 -0.69289057 -0.68442195
  -1.42512243 -0.19067191]]
[0]
no diabetics
```

Fig.17. Making Predictive System

V. RESULTS

The thought experiments are conducted on a set of data samples. The following are some screenshots of the execution of data samples for the prediction of results.

Input 1:

```
input_data=(0,190,76,48,180,32.9,0.171,63)
```

Output 1:

```
[[-1.14185152  2.1628039   0.35643175  1.72273472  0.87003069  0.1151693
  -0.90868214  2.5321362 ]]
[1]
yes diabetics
```

Input 2:

```
input_data=(0,137,40,35,168,43.1,2.228,33)
```

Output 2:

```
[[-1.14185152  0.5040552  -1.50468724  0.90726993  0.76583594  1.4097456
   5.30370198 -0.0204964 ]]
[1]
yes diabetics
```

Input 3:

```
input_data=(10,101,76,48,180,32.9,0.171,63)
```

Output 3:

```
[[ 1.82781311 -0.62264204  0.35643175  1.72273472  0.87003069  0.1151693
  -0.90868214  2.5321362 ]]
[0]
no diabetics
```

Accuracy of Algorithms

Table- I: Comparison with SVM, KNN, Decision Tree

Classifiers	Accuracy Score
SVM	0.7866449511400652
Decision Tree	0.7012987012987013
KNN	0.7112987012997987

Based on the table above, it is evident that SVM achieves a high accuracy of 78%.

Future Scope: The future scope of this research is to develop a system capable of predicting diabetes based on new data provided by users at any time. This advancement aims to contribute significantly to reducing the number of diabetes cases worldwide, thereby offering substantial benefits to the healthcare sector [10][13].

VI. CONCLUSION

Diabetes represents a heterogeneous group of diseases characterized by chronic elevation of glucose in the blood. The primary goal of the American Diabetes Association [10] is "To prevent and cure diabetes and to improve the lives of all people affected by diabetes." Through the application of Support Vector Machine (SVM), K-Nearest Neighbors (KNN), and Decision Tree techniques, we achieved accuracies of 78%, 70%, and 71% respectively. Based on these results, SVM emerges as the most effective algorithm for predicting diabetes.

DECLARATION STATEMENT

After aggregating input from all authors, I must verify the accuracy of the following information as the article's author.

- **Conflicts of Interest/ Competing Interests:** Based on my understanding, this article has no conflicts of interest.
- **Funding Support:** This article has not been sponsored or funded by any organization or agency. The independence of this research is a crucial factor in affirming its impartiality, as it has been conducted without any external sway.
- **Ethical Approval and Consent to Participate:** The data provided in this article is exempt from the requirement for ethical approval or participant consent.
- **Data Access Statement and Material Availability:** The adequate resources of this article are publicly accessible.
- **Authors Contributions:** The authorship of this article is contributed equally to all participating individuals.

REFERENCES

1. M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt and B. Scholkopf, "Support vector machines," in IEEE Intelligent Systems and their Applications", vol. 13, no. 4, pp. 18-28, July-Aug. 1998, doi: 10.1109/5254.708428. https://doi.org/10.1109/5254.708428
2. Kecman, V. Support Vector Machines – An Introduction. In: Wang, L. (eds) ,"Support Vector Machines: Theory and Applications", Studies in Fuzziness and Soft Computing, vol 177. Springer, Berlin, Heidelberg. https://doi.org/10.1007/10984697_1. https://doi.org/10.1007/10984697_1
3. A. Navada, A. N. Ansari, S. Patil and B. A. Sonkamble, "Overview of use of decision tree algorithms in machine learning," 2011 IEEE Control and System Graduate Research Colloquium, Shah Alam, Malaysia, 2011, pp. 37-42, doi: 10.1109/ICSGRC.2011.5991826. https://doi.org/10.1109/ICSGRC.2011.5991826
4. B. Charbuty and A. Abdulazeez, "Classification Based on Decision Tree Algorithm for Machine Learning", JASTT, vol. 2, no. 01, pp. 20 - 28, Mar. 2021. https://doi.org/10.38094/jastt20165
5. Rokach, L., Maimon, O. (2005). Decision Trees. In: Maimon, O., Rokach, L. (eds), "Data Mining and Knowledge Discovery Handbook", Springer, Boston, MA. https://doi.org/10.1007/0-387-25465-X_9 https://doi.org/10.1007/0-387-25465-X_9
6. Uddin, S., Haque, I., Lu, H. et al.," Comparative performance analysis of K-nearest neighbour (KNN) algorithm and its different variants for disease prediction", Sci Rep 12, 6256 (2022). https://doi.org/10.1038/s41598-022-10358-x.
7. Guo, G., Wang, H., Bell, D., Bi, Y., Greer, K. (2003). ,"KNN Model-Based Approach in Classification",. In: Meersman, R., Tari, Z., Schmidt, D.C. (eds) On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE. OTM 2003. Lecture Notes in Computer Science, vol 2888. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-39964-3_62.
8. Pima Indians Diabetes Database https://www.kagegle.com/uciml/pima-indians-diabetes-databas Predict the onset of diabetes based on diagnostic measures
9. Talha Mahboob Alam, Muhammad Atif Iqbal, Yasir Ali, Abdul Wahab, Safdar Ijaz, Talha Imtiaz Baig, Ayaz Hussain, Muhammad Awais Malik, Muhammad Mehdi Raza, Salman Ibrar, Zunish Abbas,
10. "A model for early prediction of diabetes", Informatics in Medicine Unlocked, Volume 16, 2019, 100204, ISSN 2352-9148, https://doi.org/10.1016/j.imu.2019.100204.
11. Krishnamoorthi, Raja & Joshi, Shubham & Almarzouki, Hatim & Shukla, Piyush & Rizwan, Ali & Kalpana, C. & Tiwari, Basant. (2022).," A Novel Diabetes Healthcare Disease Prediction Framework Using Machine Learning Techniques. Journal of Healthcare Engineering", 2022. 1-10. 10.1155/2022/1684017.
12. American Diabetes Association.,"Classification and diagnosis of diabetes: Standards of Medical Care in Diabetes",2018.
13. Diabetes Care 2018;41(Suppl. 1):S13–S27, https://doi.org/10.2337/dc18-S002
14. Yilmaz, N., Inan, O., & Uzer, M. S. (2014). ,"A New Data Preparation Method Based on Clustering Algorithms for Diagnosis Systems of Heart and Diabetes Diseases", Journal of Medical Systems, 38(5). doi:10.1007/s10916-014-0048-7
15. Aishwarya Mujumdar, V Vaidehi, "Diabetes Prediction using Machine Learning Algorithms", Procedia Computer Science, Volume 165, 2019, Pages 292-299, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2020.01.047.
16. Modak, S.K.S., Jha, V.K. Diabetes prediction model using machine learning techniques. Multimed Tools Appl 83, 38523–38549 (2024). https://doi.org/10.1007/s11042-023-16745-4

AUTHORS PROFILE

Dr. Ratnakumari Challa is working as Assistant Professor in the department of Computer Science and Engineering, RGUKT, RK Valley, IIIT-A, Kadapa, Andhra Pradesh. She has published more than 25 research articles in National and International Journals, Conference and Symposiums. Her interested research areas are Machine learning, Computer Vision and Security & Privacy.

Dr. Buduri. Reddaiah is working as Associate Professor in the department of Computer Science and Technology, Yogi Vemana University, Kadapa, Andhra Pradesh. His research interests are in security and Artificial Intelligence. With a focus on Artificial Intelligence and machine learning, His research endeavors to enhance design of the model for better performance.

Dr. Kanusu Srinivasa Rao, is working as an Associate Professor in the Department of Computer Science and Technology, Yogi Vemana University, Kadapa. He has presented more than 20 research articles in National and International Journals, Conference and Symposiums. His main area of interest includes Image Processing, Cryptography and Network Security, AI & Machine Learning. As the corresponding author, he embodies the collaborative spirit of this research team.

Dr. Krishnaiah Pulluru is working as Academic Consultant in the department of Computer Science and Technology, Kadapa, Andhra Pradesh. His research areas is Artificial Intelligence & Machine learning, Cryptography and Network Security. He published many papers in this area.

Dr. S. Ranga Swamy received his Ph.D. degree in Computer Science & Engineering from ANU, Guntur in 2019. He is having nearly 12 years of teaching and Research experience. He is currently working as Associate Professor, Department of C.S.E, Vignan's Institute of Management and Technology for Women, Ghatkesar, Hyderabad, Telangana, India. He has a total of 25 Research publications at International/National Journals, 12 Conferences and 10 Patent Publications. He is a life member of ISTE, IEI, SDIWC and IAENG. His current research areas are Cloud Computing, Software Engineering, IoT and Machine Learning and Deep Learning.

Venkata Narayana Reddy is working as Academic Consultant in the department of Computer Science and Technology, Kadapa, Andhra Pradesh. His research areas is Artificial Intelligence & Machine learning, Cryptography and Network Security. He published many papers in this area.

Retrieval Number: 100.1/ijitee.I995213090824
DOI: [10.35940/ijitee.I9952.13090824](https://doi.org/10.35940/ijitee.I9952.13090824)
Journal Website: www.ijitee.org

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)