DIABETES PREDICTION SYSTEM

Diabetes Prediction System: Predicting the Likelihood of Diabetes Development

Introduction:

The Diabetes Prediction System is an innovative tool developed to assist in predicting the likelihood of an individual developing diabetes. By considering various factors such as age, gender, body mass index (BMI), blood pressure, and family history of diabetes, this system provides a comprehensive risk assessment.

Utilizing cutting-edge machine learning algorithms, the system analyzes a vast dataset of diabetes patients and compares the input data of an individual to generate a prediction on their probability of developing diabetes in the future.

Purpose:

The primary goal of the Diabetes Prediction System is to raise awareness about the risk of diabetes and encourage individuals to take proactive steps for prevention or early intervention. It serves as a valuable tool for healthcare professionals, enabling them to identify high-risk individuals promptly and provide personalized care plans.

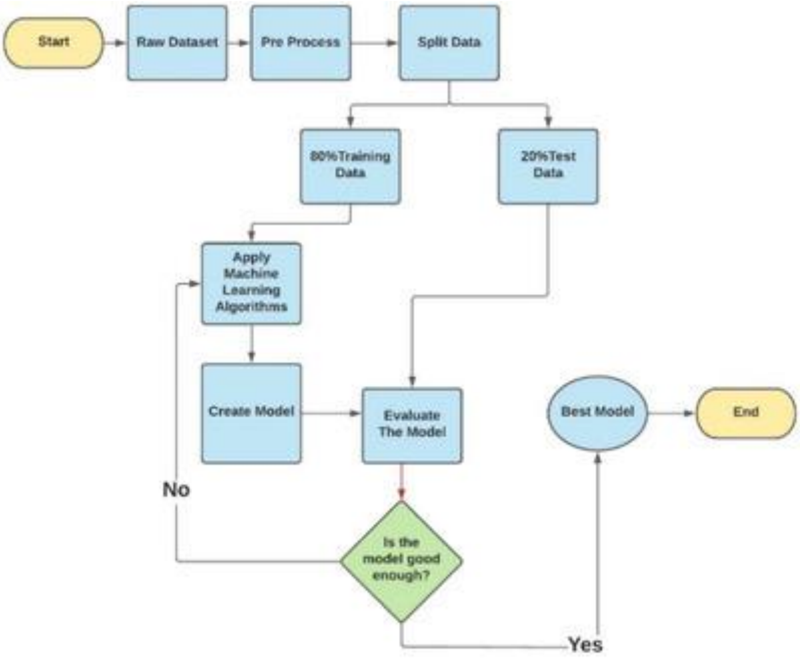
Important Note:

It is crucial to understand that the Diabetes Prediction System is not intended as a replacement for medical advice or diagnosis. If you have any concerns regarding your health, it is always recommended to consult with a qualified healthcare professional who can provide accurate medical guidance and support.

By leveraging advanced technology and data analysis, the Diabetes Prediction System aims to contribute to proactive healthcare and empower individuals to make informed decisions for their well-being.

PROPOSED SYSTEM

This section describes the working procedures and implementation of various machine learning techniques to design the proposed automatic diabetes prediction system. Figure 1 shows the different stages of this research work. First, the dataset was collected and preprocessed to remove the necessary discrepancies from the dataset, for example, replacing null instances with mean values, dealing with imbalanced class issues etc. Then the dataset was separated into the training set and test set using the holdout validation technique. Next, different classification algorithms were applied to find the best classification algorithm for this dataset. Finally, the best-performed prediction model is deployed into the proposed website and smartphone application framework.



DATASET

The Pima Indian dataset is an open-source dataset [6] that is publicly available for machine learning classification, which has been used in this work along with a private dataset. It contains 768 patients' data, and 268 of them have developed diabetes.

Figure 2 shows the ratio of people having diabetes in the Pima Indian dataset. Table 1 demonstrates the eight features of the open-source Piman Indian dataset.

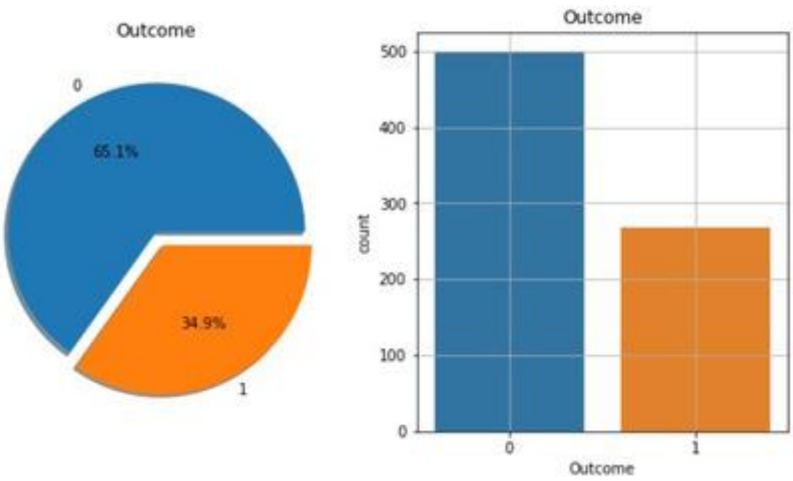


TABLE 1

Features of the Pima Indian Dataset

Pregnancies	**Skin thickness**	**Diabetes pedigree function**
Glucose	Insulin	Age
Blood pressure	BMI	

TABLE 2

Features of the RTML private dataset

Features	**Minimum**	**Maximum**	**Average**
Pregnancies	0	8	1.61
Glucose (mg/dL)	52.2	274	109.39
Blood pressure (mm Hg)	5.9	115	71.09
Skin thickness (mm)	2.9	23.3	10.78
BMI (kg/m^2)	2.61	41.62	22.69
Age (years)	17	77	27.02

There are several tools and technologies commonly used in the development of AI-based diabetes prediction systems. Here are some of the key ones:

I. Machine Learning Algorithms:
Machine learning algorithms form the backbone of AI prediction systems. Commonly used algorithms include logistic regression, support vector machines (SVM), random forests, gradient boosting, and artificial neural networks.

II. Programming Languages:
Popular programming languages used in developing AI systems include Python and R. They provide libraries and frameworks like TensorFlow, Keras, scikit-learn, and PyTorch, which have pre-built functions and models specifically designed for machine learning tasks.

III. Data Preprocessing Tools:

Preprocessing tools help clean and transform raw data into a suitable format for training machine learning models. This may involve handling missing values, normalizing data, feature scaling, and encoding categorical variables. Libraries like Pandas and NumPy in Python are commonly used for data preprocessing.

IV. Feature Selection and Engineering Tools:

These tools help in selecting relevant features from the dataset and creating new features to improve the accuracy of the prediction models. Techniques like Principal Component Analysis (PCA), Recursive Feature Elimination (RFE), and information gain can be applied using libraries like scikit-learn.

V. Data Visualization Tools:

Visualization tools like Matplotlib, Seaborn, and Plotly are used to explore and visualize data patterns, correlations, and distributions. They help in gaining insights from the data and making informed decisions about feature selection and model design.

VI. Cloud Platforms:

Cloud computing platforms, such as Amazon Web Services (AWS), Microsoft Azure, and Google Cloud, provide infrastructure and services for hosting and deploying AI models. They offer scalable resources, GPU instances for faster computations, and APIs for integrating AI models into applications.

VII. Evaluation Metrics:

Metrics like accuracy, precision, recall, F1-score, and area under the curve (AUC) are commonly used to evaluate the performance of diabetes prediction models. Libraries like scikit-learn provide built-in functions for calculating these metrics.

VIII. It is important to note that the choice of tools may vary depending on the specific requirements, preferences, and expertise of the development team.

Phase2

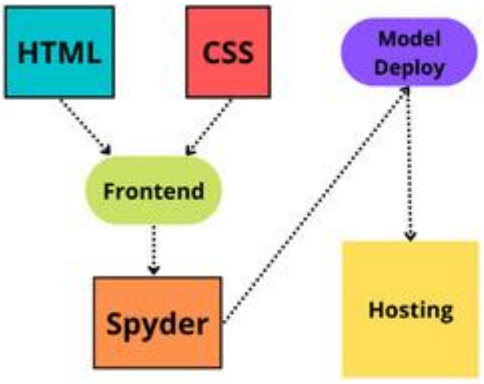
```
# Import the required libraries
 import numpy as np
 import pandas as pd from sklearn.model_selection
import train_test_split from sklearn
import svm from sklearn.metrics
 import accuracy_score
 import pickle diabetes_dataset = pd.read_csv('diabetes.csv')

# Print the first 5 rows of the dataset
diabetes_dataset.head()
 diabetes_dataset.shape

 #prints (768, 9)
 # To get the statistical measures of the data
 diabetes_dataset.describe()
diabetes_dataset['Outcome'].value_counts()
 X = diabetes_dataset.drop(columns = 'Outcome', axis=1)
Y = diabetes_dataset['Outcome']

 # To print the independent variables
print(X)
print(Y)
 X_train, X_test, Y_train, Y_test = train_test_split(X,Y, test_size = 0.2, stratify=Y,
random_state=2)
 print(X.shape, X_train.shape, X_test
 classifier = svm.SVC(kernel='linear')
classifier.fit(X_train, Y_train)
 X_train_prediction = classifier.predict(X_train)
 training_data_accuracy = accuracy_score(X_train_prediction, Y_train)

 print('Accuracy score of the training data : ', training_data_accuracy)

# Accuracy score on the test data
 X_test_prediction = classifier.predict(X_test)
 test_data_accuracy = accuracy_score(X_test_prediction, Y_test)
print('Accuracy score of the test data : ',
input_data = (5,166,72,19,175,25.8,0.587,51)
```

```
# Change the input_data to numpy array
input_data_as_numpy_array = np.asarray(input_data)

# Reshape the array for one instance
 input_data_reshaped = input_data_as_numpy_array.reshape(1,-1)
prediction = classifier.predict(input_data_reshaped)
print(prediction)
 if (prediction[0] == 0):

 print('The person is not diabetic')
else:
 print('The person is diabetic')
 filename = 'trained_model.sav'
 pickle.dump(classifier, open(filename, 'wb'))

 # Load the saved model
 loaded_model = pickle.load(open('trained_model.sav', 'rb'))
```

DEPLOYMENT OF THE PREDICTION SYSTEM

The proposed machine learning-based diabetes prediction system has been deployed into a website and smartphone application framework to work instantaneously on real data.

Web application: We have used HTML and CSS for the frontend part of the proposed website. After that, we finalized the machine learning model XGBoost with ADASYN, as it provided the best performance. The model deployment has been done with Spyder, a Python environment platform that works with Anaconda. Figure 5 shows the illustration of the website application development process.



PROGRAM

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.ensemble import RandomForestClassifier, BaggingClassifier,
GradientBoostingClassifier
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score
from sklearn.metrics import confusion_matrix, classification_report

import warnings
warnings.filterwarnings('ignore')
data = pd.read_csv("/kaggle/input/pima-indians-diabetes-database/diabetes.csv")
data.head()
print("Shape of data is", data.shape)
print("="*50)

data.info()
data.isnull().sum()
data.duplicated().sum()
data.hist(figsize=(15, 10))
plt.show()
corr = data.corr()
corr
plt.figure(figsize=(8,5))
```

```
sns.heatmap(corr, cmap='YlGnBu', annot=True, vmin=-1, vmax=1)

plt.title('Relation between features and Diabetes')

plt.show()
```

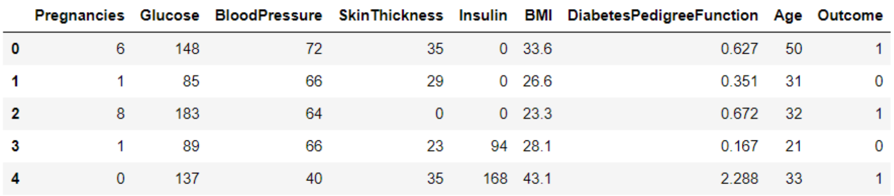
Importing Libraries

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

sns.set()

from mlxtend.plotting import plot_decision_regions
import missingno as msno
from pandas.plotting import scatter_matrix
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier

from sklearn.metrics import confusion_matrix
from sklearn import metrics
from sklearn.metrics import classification_report
import warnings
warnings.filterwarnings('ignore')
%matplotlib inline
```

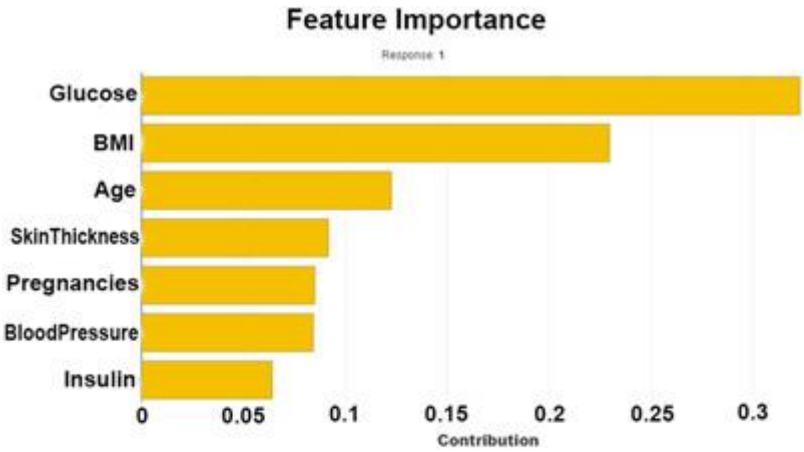
Here we will be reading the dataset which is in the CSV format

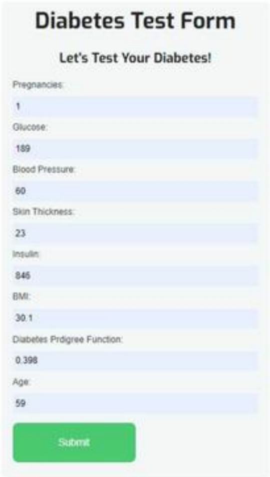
```
diabetes_df = pd.read_csv('diabetes.csv')
diabetes_df.head()
```

Output:



	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




Diabetes Test Form

Let's Test Your Diabetes!

Pregnancies:
1

Glucose:
189

Blood Pressure:
60

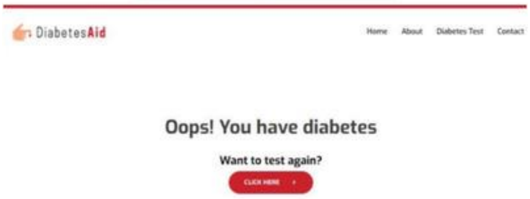
Skin Thickness:
23

Insulin:
846

BMI:
30.1

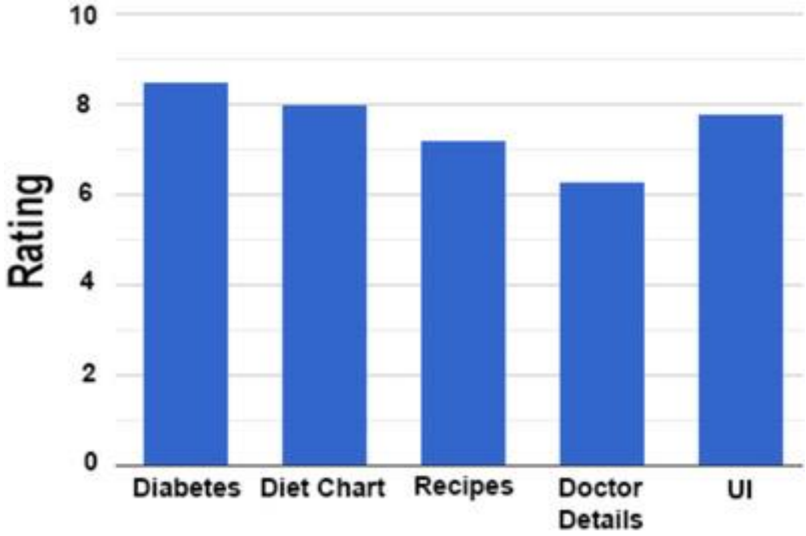
Diabetes Prdigree Function:
0.398

Age:
59



Oops! You have diabetes

Want to test again?



Advantages to a diabetes prediction system:

- Early detection: A diabetes prediction system can identify individuals who are at a higher risk of developing diabetes at an early stage. This allows for timely intervention and preventive measures, which can help in managing the condition effectively.

- Personalized care: By predicting the likelihood of developing diabetes, the system can provide personalized recommendations and interventions based on individual risk factors. This can include lifestyle modifications, such as diet and exercise plans, and regular monitoring to prevent or delay the onset of diabetes.

- Improved health outcomes: By identifying individuals at risk and providing targeted interventions, a diabetes prediction system can potentially reduce the incidence of diabetes-related complications. This can lead to improved health outcomes and a better quality of life for those at risk.

- Cost-effective: Preventive measures are generally more cost-effective than treating chronic conditions. By identifying and managing individuals at risk of diabetes, a prediction system can potentially reduce healthcare costs associated with diabetes management, including medications, hospitalizations, and other treatments.

- Empowering individuals: A diabetes prediction system can empower individuals to take control of their health by providing them with information about their risk factors. This knowledge can motivate them to make healthier lifestyle choices and actively engage in preventive measures.

It is important to note that a diabetes prediction system should always be used as a tool to support healthcare professionals in making informed decisions and should not replace medical advice or diagnosis from a qualified healthcare provider.

Disadvantages diabetes prediction system

- False positives and false negatives: No prediction system is perfect, and there is always a chance of false positive or false negative results. False positives can lead to unnecessary anxiety and stress for individuals who may not actually develop diabetes, while false negatives can provide a false sense of security and delay necessary intervention.

- Reliance on data accuracy: A diabetes prediction system heavily relies on accurate and up-to-date data. If the input data is incomplete, incorrect, or outdated, it can lead to inaccurate predictions and recommendations.

- Ethical concerns: There may be ethical concerns regarding the privacy and security of personal health information used by the prediction system. It is important to ensure that adequate measures are in place to protect sensitive data and adhere to privacy regulations.

- Overdiagnosis and overtreatment: The use of a diabetes prediction system may result in overdiagnosis, where individuals who would otherwise never develop diabetes are unnecessarily labeled as being at risk. This can lead to overtreatment, including unnecessary medication use or invasive procedures, with associated risks and costs.

- Emotional impact: Receiving a prediction of being at risk for diabetes can cause significant emotional distress for some individuals. It is important to provide appropriate counseling and support to those who receive such predictions to help them navigate their emotions and make informed decisions.

- It is crucial to approach diabetes prediction systems with caution, keeping in mind that they should complement rather than replace personalized medical advice from qualified healthcare professionals.

CONCLUSION

Diabetes can be a reason for reducing life expectancy and quality. Predicting this chronic disorder earlier can reduce the risk and complications of many diseases in the long run. In this paper, an automatic diabetes prediction system using various machine learning approaches has been proposed. The open-source Pima Indian and a private dataset of female Bangladeshi patients have been used in this work. SMOTE and ADASYN preprocessing techniques have been applied to handle the issue of imbalanced class problems. This research paper reported different performance metrics, that is, precision, recall, accuracy, F1 score, and AUC for various machine learning and ensemble techniques. The XGBoost classifier achieved the best performance with 81% accuracy and an F1 score and AUC of 0.81 and 0.84, respectively, with the ADASYN approach. Next, the domain adaptation technique has been applied to demonstrate the versatility of the proposed prediction system. Finally, the best-performed XGBoost framework has been deployed into a website and smartphone application to predict diabetes instantly. There are some future scopes of this work, for example, we recommend getting additional private data with a larger cohort of patients to get better results. Another extension of this work is combining machine learning models with fuzzy logic techniques and applying optimization approaches.