AI Based Diabetes Prediction System

An AI-based Diabetes Prediction System is a healthcare application that harnesses artificial intelligence (AI) and machine learning techniques to forecast the risk or likelihood of an individual developing diabetes in the future. This system is designed to assist in early diagnosis and proactive management of diabetes. Here's a more detailed explanation of how it works:

ALGORITHM:LOGISTIC REGRESSION

Logistic regression is a commonly used algorithm in AI-based Diabetes Prediction Systems. It's particularly well-suited for binary classification tasks, such as predicting whether an individual is at risk of developing diabetes or not. Here's how logistic regression is applied in such systems:

***1. Data Collection**: Data related to individuals, including features like age, gender, family history of diabetes, BMI, blood glucose levels, and other relevant factors, is collected. The outcome variable, whether a person has diabetes or not (often encoded as 0 for non-diabetic and 1 for diabetic), is also part of the dataset.*

***2. Data Preprocessing**: Data preprocessing steps are carried out, which may include handling missing values, normalizing or standardizing the data, and encoding categorical variables.*

***3. Feature Selection/Engineering**: Relevant features are selected or engineered based on domain knowledge or using automated methods. In the context of diabetes prediction, important features might include age, BMI, and fasting blood glucose levels.*

***4. Model Training**: Logistic regression models are trained using the prepared dataset. The algorithm fits a logistic curve to the data, which is a sigmoid-shaped curve that models the probability of an individual having diabetes based on the selected features.*

***5. Parameter Estimation**: During training, logistic regression estimates the coefficients (weights) for each feature. These coefficients determine the impact of each feature on the probability of having diabetes. Positive coefficients indicate that as a feature's value increases, the probability of diabetes increases, while negative coefficients indicate the opposite.*

***6. Prediction**: Once the logistic regression model is trained, it can be used to make predictions. For a new individual, the model calculates a probability score. If the score is above a certain threshold (often set at 0.5), the model predicts that the individual is at risk of developing diabetes.*

***7. Model Evaluation**: The model's performance is evaluated using various metrics such as accuracy, precision, recall, F1-score, and area under the receiver operating characteristic curve (AUC-ROC). Cross-validation and testing on a separate dataset are typically used to assess the model's generalization performance.*

***8. Interpretability**: One of the advantages of logistic regression is its interpretability. You can easily interpret the impact of each feature on the predicted outcome by examining the coefficient values.*

Logistic regression is valuable in diabetes prediction systems due to its simplicity, interpretability, and effectiveness in binary classification tasks. However, it may not capture complex nonlinear relationships in the data as effectively as some other algorithms. Therefore, the choice of algorithm should be based on the specific characteristics of the dataset and the desired balance between interpretability and predictive accuracy. Often, a combination of algorithms is used to enhance the overall performance of the system.

PERFORMACE OF ALOGITTHM

The performance of a machine learning model, including logistic regression, in a Diabetes Prediction System is typically assessed using various evaluation metrics to gauge how well the model is making predictions. The choice of evaluation metrics depends on the specific goals of the system and the balance between different types of errors. Here are some common evaluation metrics used to assess the performance of a logistic regression model in a diabetes prediction system:

*1. **Accuracy**: Accuracy measures the proportion of correctly classified instances (both true positives and true negatives) out of the total instances. While it's a straightforward metric, it may not be ideal when there's a class imbalance (e.g., a lot more non-diabetic cases than diabetic cases).*

*2. **Precision**: Precision is the ratio of true positives to the total number of instances predicted as diabetic. It tells us how many of the predicted diabetic cases are actually true diabetic cases. High precision means fewer false positives.*

*3. **Recall (Sensitivity)**: Recall measures the ratio of true positives to the total number of actual diabetic cases. It tells us how many of the true diabetic cases were correctly identified by the model. High recall means fewer false negatives.*

*4. **F1-Score**: The F1-Score is the harmonic mean of precision and recall. It provides a balanced measure of a model's accuracy, particularly when there's an uneven class distribution.*

*5. **Specificity**: Specificity is the ratio of true negatives to the total number of actual non-diabetic cases. It measures the model's ability to correctly identify non-diabetic cases.*

*6. **ROC Curve (Receiver Operating Characteristic Curve)**: The ROC curve is a graphical representation of the model's ability to distinguish between diabetic and non-diabetic cases. The area under the ROC curve (AUC-ROC) is a common metric that quantifies the model's performance. A higher AUC-ROC indicates a better ability to discriminate between the two classes.*

*7. **Confusion Matrix**: A confusion matrix provides a detailed breakdown of the model's predictions, showing the number of true positives, true negatives, false positives, and false negatives.*

*8. **Accuracy Paradox**: Sometimes, accuracy alone can be misleading, especially in imbalanced datasets. An accuracy paradox occurs when a high level of accuracy is achieved by simply predicting the majority class (e.g., non-diabetic) for all instances. In such cases, other metrics like precision, recall, and F1-score become more important.*

*9. **Cross-Validation**: Cross-validation techniques, such as k-fold cross-validation, are often used to assess how well the model generalizes to new, unseen data. It helps estimate the model's performance on an independent dataset.*

The choice of the most appropriate evaluation metrics depends on the specific goals of your Diabetes Prediction System and the relative importance of minimizing false positives or false negatives. A well-balanced approach often involves considering a combination of these metrics to comprehensively assess the model's performance.

In practice, when using logistic regression for diabetes prediction, the model's performance is often measured in terms of accuracy, precision, recall, F1-score, and AUC-ROC, as these metrics provide a more comprehensive view of the model's ability to predict diabetes while considering both true positives and true negatives.

PROGRAM FOR DIABETES PREDICTION

1.Importing libraries

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

2.Importing Dataset

```
dataset = pd.read_csv('../input/diabetes-data-set/diabetes.csv')
```

3.Viewing the dataset, its dimensions, features and statistical summary

```
dataset.head()
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1

```
dataset.info()
```

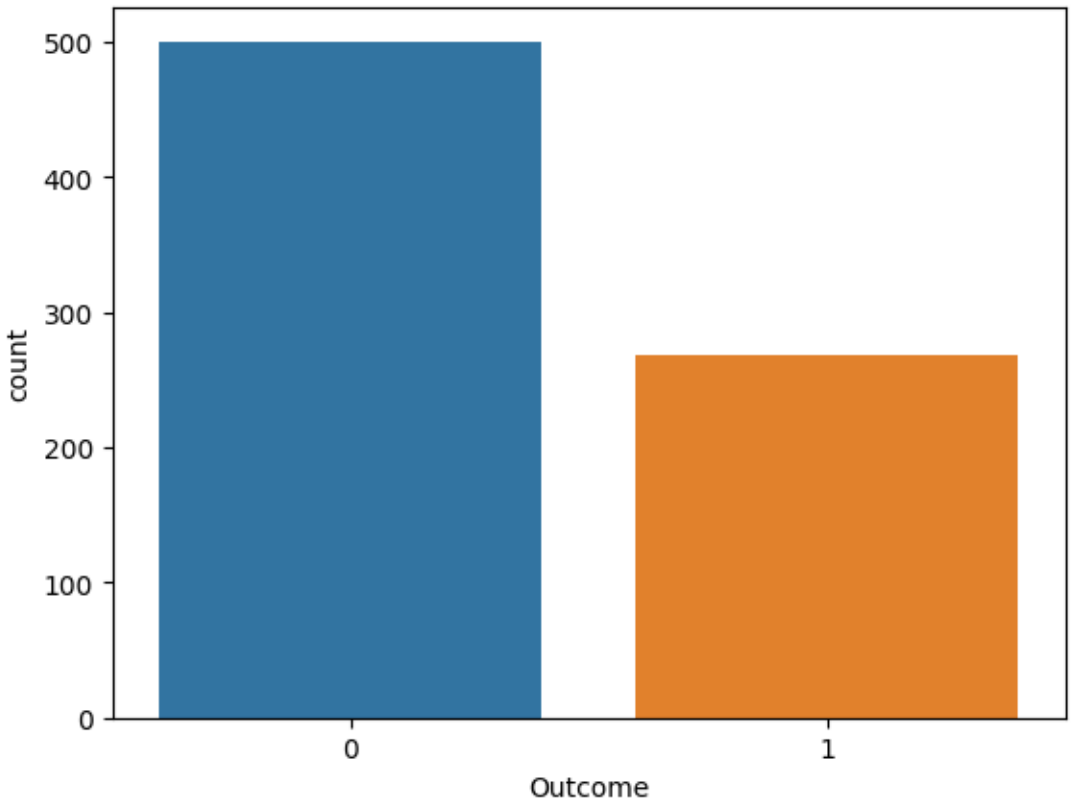
```
(768, 9)
```

```
dataset.info()
```

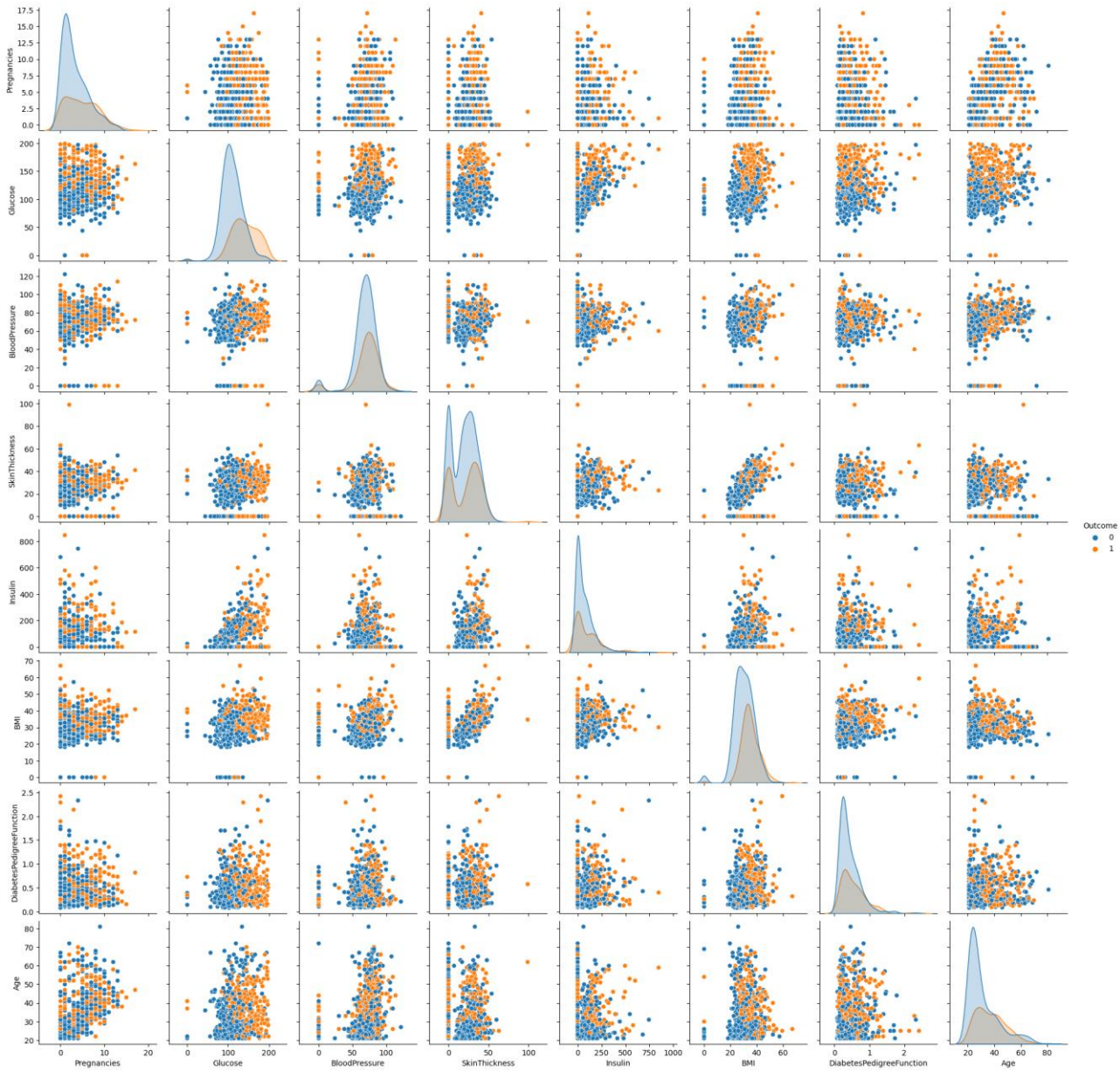
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 768 entries, 0 to 767
Data columns (total 9 columns):
 #   Column                    Non-Null Count  Dtype  
---  ------                    --------------  -----  
 0   Pregnancies               768 non-null    int64  
 1   Glucose                   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
```

```
sns.countplot(x = 'Outcome',data = dataset)
```

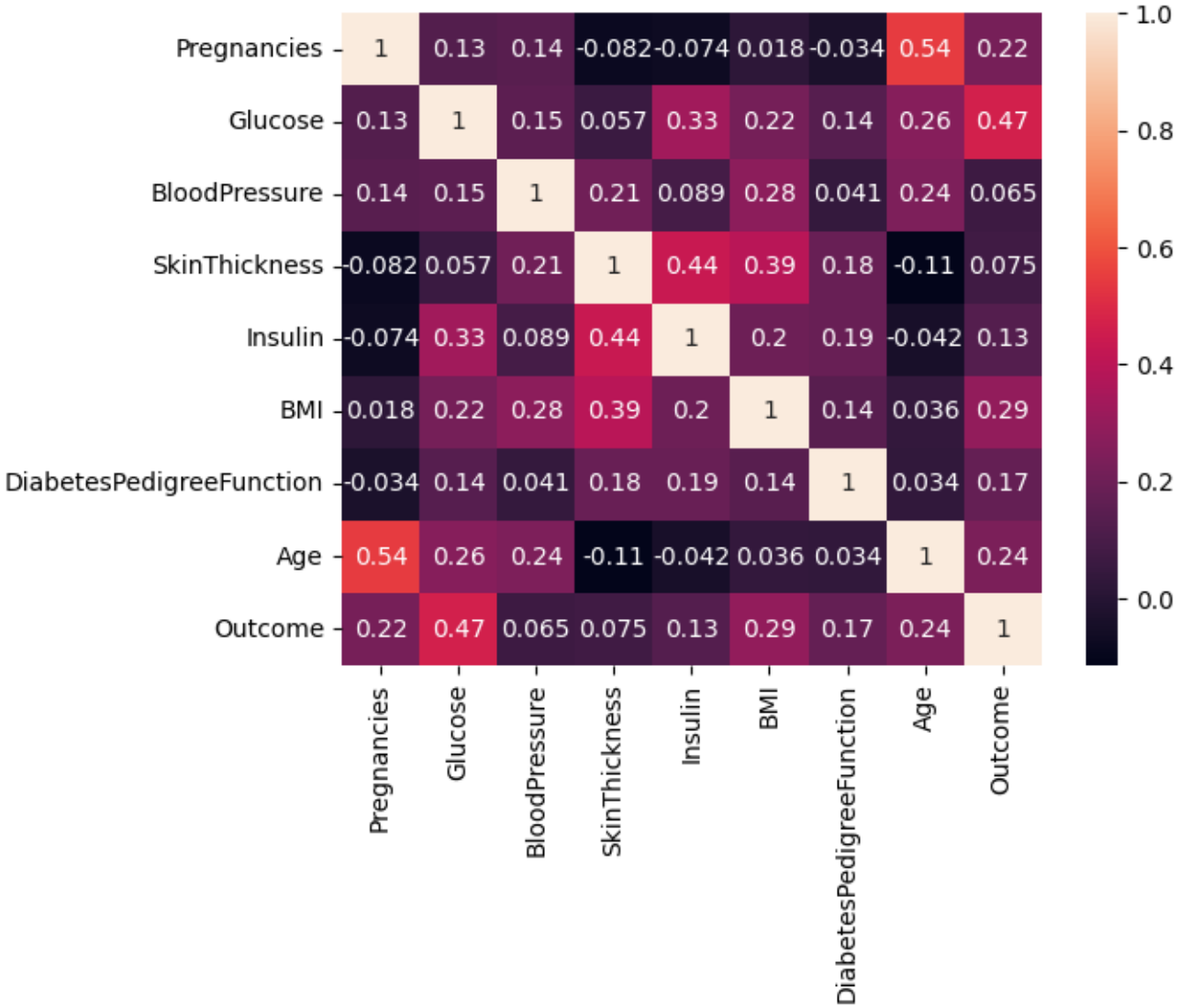
```
<Axes: xlabel='Outcome', ylabel='count'>
```



```
# Pairplot
sns.pairplot(data = dataset, hue = 'Outcome')
plt.show
```



```
# Heatmap
sns.heatmap(dataset.corr(), annot = True)
plt.show()
```



4.Processing the Data

```
# Replacing zero values with NaN
dataset_new = dataset
```

```
dataset_new[["Glucose", "BloodPressure", "SkinThickness", "Insulin", "BMI"]] =
dataset_new[["Glucose", "BloodPressure", "SkinThickness", "Insulin",
"BMI"]].replace(0, np.NaN)


# Count of NaN
dataset_new.isnull().sum()
```

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

```
# Replacing NaN with mean values
dataset_new["Glucose"].fillna(dataset_new["Glucose"].mean(), inplace = True)
dataset_new["BloodPressure"].fillna(dataset_new["BloodPressure"].mean(), inplace
= True)
dataset_new["SkinThickness"].fillna(dataset_new["SkinThickness"].mean(), inplace
= True)
dataset_new["Insulin"].fillna(dataset_new["Insulin"].mean(), inplace = True)
dataset_new["BMI"].fillna(dataset_new["BMI"].mean(), inplace = True)

dataset_new.isnull().sum()
```

```
Pregnancies                 0
Glucose                     0
BloodPressure               0
SkinThickness               0
Insulin                     0
BMI                         0
DiabetesPedigreeFunction    0
Age                         0
Outcome                     0
dtype: int64
```

5.Logistic Regression

```
y = dataset_new['Outcome']
X = dataset_new.drop('Outcome', axis=1)
```

```
# Splitting X and Y
from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(X, y, test_size = 0.20,
random_state = 42, stratify = dataset_new['Outcome'] )

from sklearn.linear_model import LogisticRegression
model = LogisticRegression()
model.fit(X_train, Y_train)
y_predict = model.predict(X_test)

y_predict
```

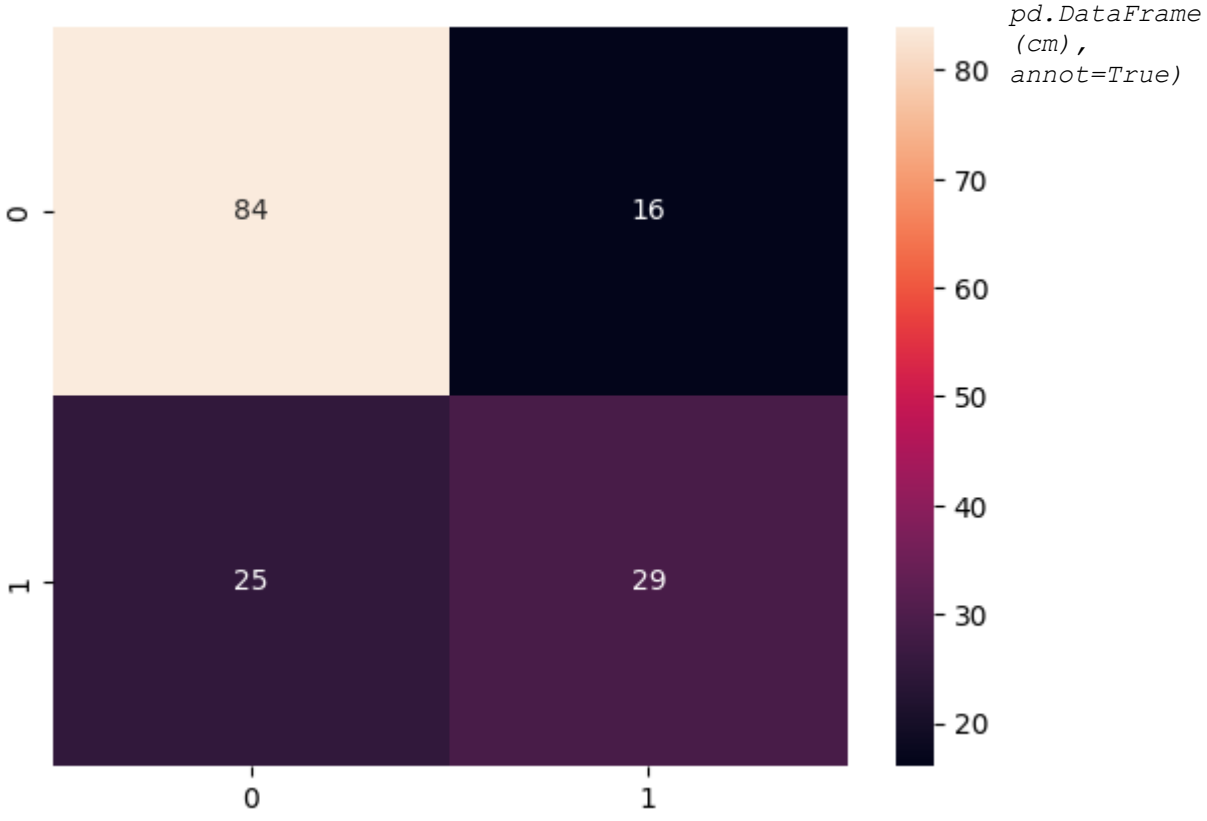
```
array([1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1,
       0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0,
       0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 1, 0, 0,
       1, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 1,
       0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 1, 1, 0, 0, 0, 1,
       0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0])
```

```
# Confusion matrix
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(Y_test, y_predict)
cm
```

```
array([[84, 16],
       [25, 29]])
```

```
# Heatmap of Confusion matrix
```

```
sns.heatmap(
pd.DataFrame
(cm),
annot=True)
```



```
from sklearn.metrics import accuracy_score
accuracy =accuracy_score(Y_test, y_predict)
accuracy
```

```
0.7337662337662337
```

```
y_predict = model.predict([[1,148,72,35,79.799,33.6,0.627,50]])
print(y_predict)
if y_predict==1:
    print("Diabetic")
else:
    print("Non Diabetic")
```

```
Diabetic
```

CONCLUSION

In conclusion, an AI-based Diabetes Prediction System is a valuable tool in healthcare for early diagnosis and proactive management of diabetes. These systems use artificial intelligence and machine learning algorithms, with logistic regression being one of the commonly used models, to predict the risk of an individual developing diabetes.Overall, AI-based Diabetes Prediction Systems, with logistic regression and other machine learning algorithms, have the potential to assist healthcare professionals in the early diagnosis and management of diabetes, ultimately improving patient outcomes and quality of care. However, it's important to emphasize that these systems are designed to support healthcare professionals and not replace their

expertise. They should be used as complementary tools to enhance healthcare decision-making