**AI\_PHASE-5**

**AI-BASED**

**DIABETESPREDICTION**

**SYSTEM**

**NAME:RAJ MOHAN.S**

**REG.NO:610821205036**

**DEPT:B-TECH-IT**

**YEAR:3RD**

**Phase 1: Problem Definition and**

**Design Thinking**

In this part you will need to understand the problem statement and create a document on what have you understood and how will you proceed ahead with solving the problem.

**Problem Definition:**

An AI-based diabetes prediction system is a computational tool that uses artificial intelligence algorithms and machine learning techniques to assess the risk of an individual developing diabetes.AI is used to spot patterns in behavior that lead to either high or low blood sugar levels in diabetes patients. A system is used to predict whether a patient has diabetes based on some of its health-related details such as BMI (Body Mass Index), blood pressure, Insulin, etc. Here is the brief description of typically explanation of diabetes prediction system.

**Design Thinking:**

Data Collection: The system collects relevant data from individuals, which may include personal information (age, gender), medical history (family history of diabetes, previous diagnoses), and physiological data (blood glucose levels, body mass index, blood pressure, etc.).

**Data Preprocessing:**

Raw data is cleaned, standardized, and prepared for analysis. Missing values are handled, and outliers are identified and addressed.

**Feature Selection:**

Relevant features or variables that contribute to diabetes risk are selected. This step helps reduce noise in the data and focuses on the most significant predictors. Machine Learning

**Model:**

The system employs various machine learning models, such as logistic regression, decision trees, random forests, or neural networks, to analyze the data. These models learn patterns and relationships in the data to make predictions.

**Training:**

The selected machine learning model is trained on a labeled dataset, which includes historical data with known diabetes outcomes. The model learns to recognize patterns that are indicative of diabetes risk.

**Evaluation:**

The system evaluates the performance of the trained model using metrics like accuracy, precision, recall, and F1 score. Cross-validation techniques are often employed to ensure robustness.

**Prediction:**

When a new individual’s data is input into the system, the trained model makes predictions about their likelihood of developing diabetes. This prediction can be binary (yes/no) or probabilistic (probability score).

**Interpretability:**

Many AI-based systems provide insights into the factors contributing to the prediction, allowing healthcare professionals to understand the basis of the prediction. Feedback

**Loop:**

Continuous learning and improvement are facilitated through a feedback loop where new data and outcomes are used to retrain and refine the model.

**Deployment:**

The system can be integrated into healthcare settings, allowing doctors and patients to assess diabetes risk. It can also be used for proactive healthcare management and early intervention.

**Empathize:**

Understand the needs and concerns of potential users, including patients, healthcare professionals, and researchers. Conduct interviews, surveys, and observations to gather insights into the challenges and goals related to diabetes prediction and management.

**Define:**

Define the problem statement and objectives for the AI-based system. This could include identifying the specific population it aims to serve and the desired outcomes. Create user personas and scenarios to represent the different user groups and their needs.

**Ideate:**

Brainstorm potential solutions and features for the diabetes prediction system. Encourage creativity and divergent thinking. Consider different AI algorithms, data sources, and user interfaces that can enhance the system's effectiveness and usability.

**Prototype:**

Develop a prototype or mockup of the AI system to visualize its user interface and functionality.Create low-fidelity and high-fidelity prototypes that can be tested with users for feedback.

**Test:**

Gather feedback from users by conducting usability testing and scenario-based testing of the prototype. Use this feedback to refine the design and functionality of the system.

**Iterate:**

Based on user feedback and insights, make iterative improvements to the system's design and features. Continue testing and refining the prototype until it meets the needs and expectations of users.

**Develop:**

Once the prototype has been validated and refined, move forward with the development of the AI-based diabetes prediction system. Implement the chosen AI algorithms, data pipelines, and user interfaces in a robust and scalable manner.

Test (Again): Conduct rigorous testing of the fully developed system to ensure it performs accurately and reliably in real-world scenarios. Address any technical issues or bugs that arise during testing.

Deploy: Deploy the AI system in a healthcare or clinical setting, ensuring compliance with relevant regulations and data privacy standards.

Phase 2: Innovation

In this section you need to put your design into innovation to solve the problem.

**INNOVATION :**

To learn checking of the device is recommended for DM forecast. For diabetes prediction and monitoring, the recommended structural benefits of effective decision-making technique and helping in good outcome.

- Keeping in view the huge development in the ailment, the recommended prototype goal is to deal with efficiently through cloud computing solutions. Mostly, research is not reviewing the F-score, But some research make a regular estimate of categorizing model with Fscore.

- Outlier detection and missing value imputation methods were considered in the generation of predictive models ML models were appropriately predicted when the ratios of each class were similar. If the class ratio was imbalanced, the algorithm learned to predict most of the classes in a biased manner. Consequently, many studies have been conducted to solve the problem of class imbalance.

- Filtering is a method for selecting variables from properties of statistical data such as mutual information and correlation coefficient without using modeling approaches. Wrapping is a method for selecting the subset with optimal prediction accuracy using only a subset of the variables. The embedded method includes feature selection in its modeling technique.

- The results of feature selection were presented using the filtering method-based Select K Best, embedded method-based least absolute shrinkage and selection operator (LASSO), wrapping method-based Boruta, and permutation feature importance techniques. The Select K Best algorithm selects variables using aunivariate statistical test. The chi-square statistic, which is commonly used for classification tasks, was used in this study.

- The LASSO method is a technique for reducing insignificant variables in the regression model to zero by penalizing the objective function to minimize prediction error (Fonti and Belitser Citation2017). The Boruta algorithm removes variables that are considered less significant than randomized copy variables (Kursa, Jankowski, and Rudnick Citation2010).

- Permutation significance is used to determine the importance of each variable through the degree of increase in error when a particular variable is randomly permuted. Developing an AI-based diabetes prediction system involves several key innovation steps to ensure its accuracy, reliability, and effectiveness in helping individuals manage their health. Here's a high-level overview of the innovation steps for such a system:

1. Data Collection and Preprocessing: - Collect comprehensive and diverse datasets containing health records, medical history, lifestyle factors, genetic information, and biomarker data related to diabetes.

- Clean and preprocess the data to handle missing values, outliers, and inconsistencies. - Ensure data privacy and compliance with relevant regulations, such as HIPAA.

2. Feature Engineering: - Identify relevant features (variables) from the data that can contribute to diabetes prediction, such as age, BMI, family history, dietary habits, and blood glucose levels.

- Transform and engineer features to extract meaningful information, such as creating new features, normalizing data, or handling categorical variables.

3. Model Selection: - Explore various machine learning and AI algorithms suitable for diabetes prediction, such as logistic regression, decision trees, random forests, support vector machines, deep learning models (e.g., neural networks), or ensemble methods.

- Evaluate the performance of different models using metrics like accuracy, precision, recall, F1-score, and area under the ROC curve (AUC).

4. Data Splitting and Validation: - Split the dataset into training, validation, and test sets to assess model performance. - Employ cross-validation techniques to ensure the model's generalization and avoid overfitting.

5. Model Training and Tuning: - Train the selected AI model(s) on the training data, optimizing hyperparameters and model architecture.

- Fine-tune the model using the validation set to achieve the best performance.

6. Interpretability and Explainability: - Develop methods to interpret and explain the AI model's predictions, making them understandable to healthcare professionals and patients.

- Utilize techniques like feature importance scores, SHAP values, or LIME (Local Interpretable Model-agnostic Explanations).

7. Continuous Learning and Updating: - Implement mechanisms for the model to adapt and learn from new data over time.

- Periodically retrain the model with fresh data to improve prediction accuracy and account for changing health conditions.

8. Integration and Deployment: - Integrate the AI model into a user-friendly and secure platform, such as a mobile app or web application. - Ensure seamless communication with electronic health records (EHR) systems and wearable devices.

9. Evaluation and Validation: - Conduct extensive testing and validation of the AI-based diabetes prediction system, involving real-world data and user feedback.

- Assess its clinical utility and impact on patient outcomes and healthcare providers' decisionmaking.

10. Ethical Considerations and Regulatory Compliance: - Address ethical concerns related to data privacy, bias, and fairness in AI predictions. - Ensure compliance with healthcare regulations and standards, such as GDPR, HIPAA, and FDA guidelines.

11. User Education and Engagement: - Develop educational materials and provide user support to help individuals understand and utilize the system effectively.

- Promote user engagement and adherence to health recommendations

12. Scaling and Accessibility: - Plan for scalability to accommodate a growing user base. - Ensure accessibility for a diverse range of users, including those with disabilities.

13. Research and Innovation: - Stay updated on the latest advancements in AI and diabetes research to continually improve the system's performance and features.

14. Collaboration: - Collaborate with healthcare professionals, researchers, and organizations to validate and refine the system's capabilities.

15. Feedback Loop: - Establish a feedback loop for users to report issues, provide feedback, and suggest improvements to enhance the system's effectiveness. Innovations in AI-based diabetes prediction systems can significantly improve the early detection and management of diabetes, ultimately leading to better healthcareoutcomes for individuals with diabetes.

Phase 3: Development Part 1

In this section begin building your project by loading and preprocessing the dataset.

**PROGRAM :**

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns dataset=pd.read\_csv("C:/Users/STUDENT/Desktop/diabe tes.csv")

dataset.head()

dataset.shape

dataset.isnull().values.any()

dataset.info()

dataset.describe()

dataset.isnull().sum()

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

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

sns.heatmap(dataset.corr(), annot = True)

plt.show()

dataset\_new = dataset

dataset\_new[['Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI']] = dataset\_new[['Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI']].replace(0,np.NaN)

dataset\_new.isnull().sum()

dataset\_new['Glucose'].fillna(dataset\_new['Glucose'].me an(),inplace=True)

dataset\_new['BloodPressure'].fillna(dataset\_new['Blood Pressure'].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()

y = dataset\_new['Outcome']

X = dataset\_new.drop('Outcome', axis=1)

from sklearn.model\_selection import train\_test\_splitX\_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

from sklearn.metrics import confusion\_matrix

Cm = confusion\_matrix(Y\_test, Y\_predict)

Cm

sns.heatmap(pd.DataFrame(Cm),annot=True) from sklearn.metrics

import accuracy\_score accuracy =accuracy\_score(Y\_test, Y\_predict)

accuracy

Y\_predict = Model.predict([[1,148,72,35,79.799,33.6,0.627,50]])

print(Y\_predict)

if Y\_predict==1:

print('Diabetes')

else:

print('Non Diabeties')



Phase 4: Development Part 2

In this section continue building the project by performing different activities like feature engineering , model training , evaluation , etc.

**Program Code:**

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

Dataset=pd.read\_csv(‘C:\\Users\\STUDENT\\Desktop\\diabetes.csv’)

Dataset.head()

Dataset.shape

Dataset.isnull().values.any()

Dataset.info()

Dataset.describe()

Dataset.isnull().sum()

Sns.countplot(x = ‘Outcome’,data = dataset)

Sns.pairplot(data = dataset, hue = ‘Outcome’)

Plt.show()

Sns.heatmap(dataset.corr(), annot = True)

Plt.show()

Dataset\_new = dataset

Dataset\_new[[“Glucose”, “BloodPressure”, “SkinThickness”, “Insulin”, “BMI”]] = dataset\_new[[“Glucose”, “BloodPressure”, “SkinThickness”, “Insulin”, “BMI”]].replace(0, np.NaN)

Dataset\_new.isnull().sum()

# Check for Missing Values

Missing\_values = df.isnull().sum()

Print(“Missing Values:”)

Print(missing\_values)

# Handle missing values (if any)

# For example, fill missing values with the mean of the column

Mean\_fill = df.mean()

Df.fillna(mean\_fill, inplace=True)

# Check for Duplicate Rows

Duplicate\_rows = df[df.duplicated()]

Print(“\nDuplicate Rows:”)

Print(duplicate\_rows)

# Handle duplicate rows (if any)

# For example, drop duplicate rows

Df.drop\_duplicates(inplace=True)

# Data Analysis

# Summary Statistics

Summary\_stats = df.describe()

Print(“\nSummary Statistics:”)

Print(summary\_stats)

# Class Distribution (for binary classification problems)

Class\_distribution = df[‘Outcome’].value\_counts()

Print(“\nClass Distribution:”)

Print(class\_distribution)

#Data Visualization

Sns.pairplot(df, hue=’Outcome’)

Plt.show()

#Support Vector Machine (SVM) Modeling

# Separate features and target variable

X = df.drop(‘Outcome’, axis=1)

Y = df[‘Outcome’]

# Split the dataset into a training and testing set

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# Standardize features

Scaler = StandardScaler()

X\_train = scaler.fit\_transform(X\_train)

X\_test = scaler.transform(X\_test)

# Initialize and train the SVM model

Model = SVC(kernel=’linear’, random\_state=42)

Model.fit(X\_train, y\_train)

# Make predictions

Y\_pred = model.predict(X\_test)

# Evaluate the model

Accuracy = accuracy\_score(y\_test, y\_pred)

Print(f’Accuracy: {accuracy:.2f}’)

# Classification report and confusion matrix

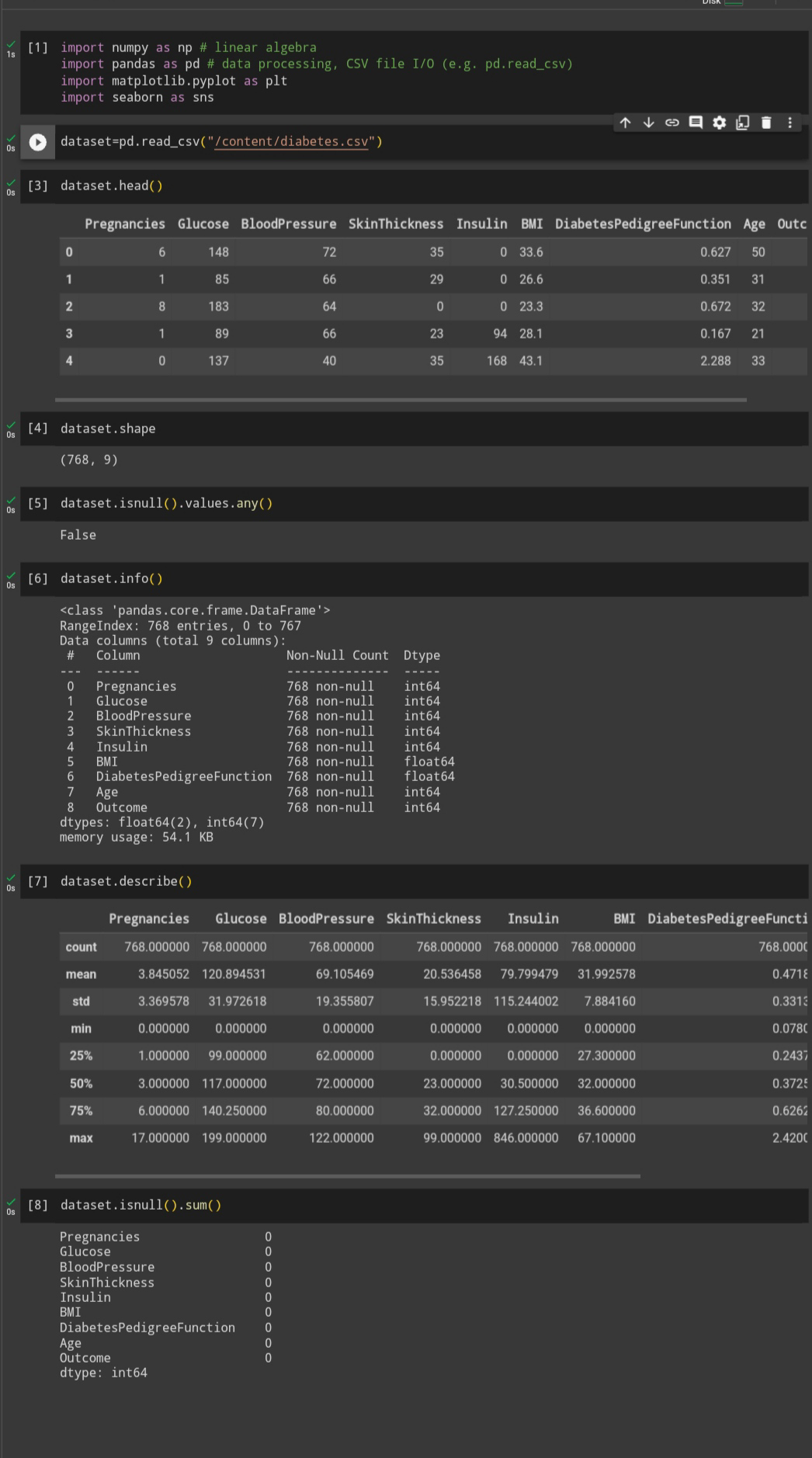
Print(classification\_report(y\_test, y\_pred))

Cm = confusion\_matrix(y\_test, y\_pred)

Sns.heatmap(cm, annot=True, fmt=’d’)

Plt.show()

**OUTPUT:**

****

**Missing Values:**

Pregnancies 0

Glucose 0

BloodPressure 0

SkinThickness 0

Insulin 0

BMI 0

DiabetesPedigreeFunction 0

Age 0

Outcome 0

**Dtype: int64**

**Duplicate Rows:**

Empty DataFrame

Columns: [Pregnancies, Glucose, BloodPressure, SkinThickness, Insulin, BMI, DiabetesPedigreeFunction, Age, Outcome]

**Index: []**

**Summary Statistics:**

Pregnancies Glucose BloodPressure SkinThickness Insulin \

Count 768.000000 768.000000 768.000000 768.000000 768.000000

Mean 3.845052 120.894531 69.105469 20.536458 79.799479

Std 3.369578 31.972618 19.355807 15.952218 115.244002

Min 0.000000 0.000000 0.000000 0.000000 0.000000

25% 1.000000 99.000000 62.000000 0.000000 0.000000

50% 3.000000 117.000000 72.000000 23.000000 30.500000

75% 6.000000 140.250000 80.000000 32.000000 127.250000

Max 17.000000 199.000000 122.000000 99.000000 846.000000

BMI DiabetesPedigreeFunction Age Outcome

Count 768.000000 768.000000 768.000000 768.000000

Mean 31.992578 0.471876 33.240885 0.348958

Std 7.884160 0.331329 11.760232 0.476951

Min 0.000000 0.078000 21.000000 0.000000

25% 27.300000 0.243750 24.000000 0.000000

50% 32.000000 0.372500 29.000000 0.000000

75% 36.600000 0.626250 41.000000 1.000000

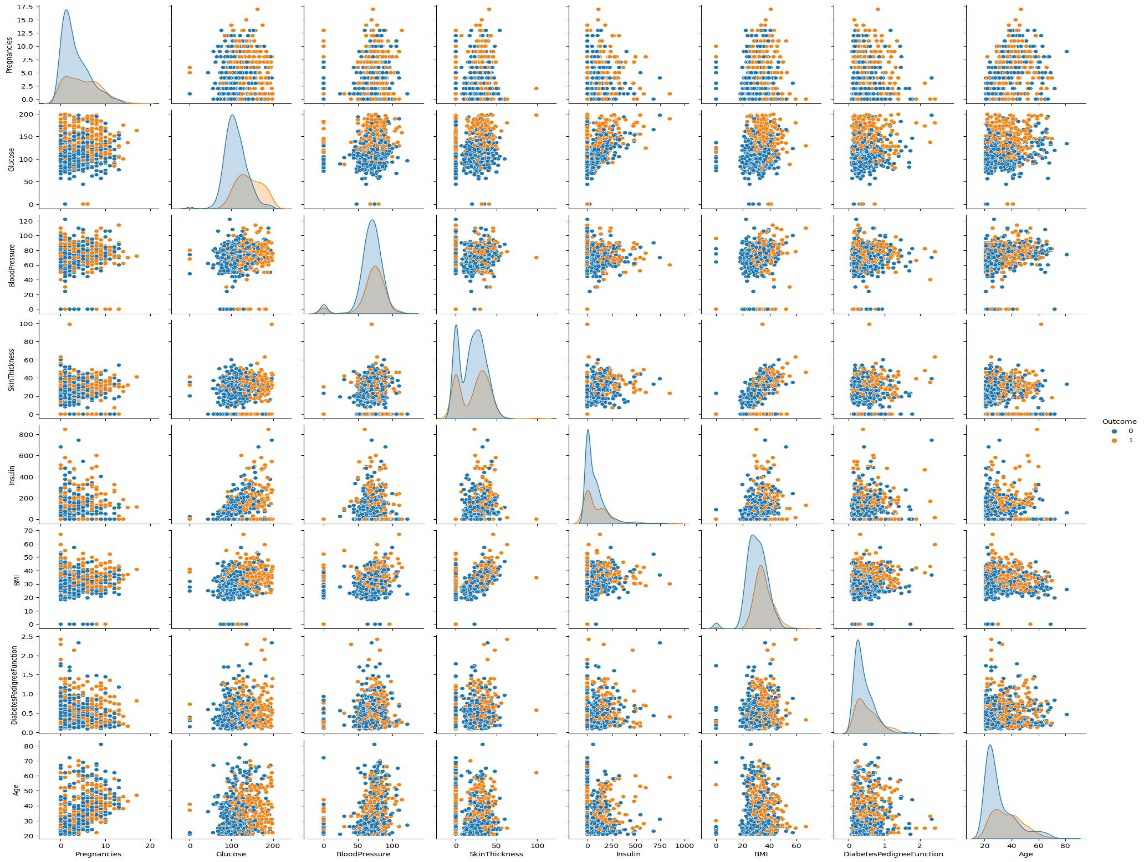
Max 67.100000 2.420000 81.000000 1.000000

**Class Distribution:**

Outcome

1. 500
2. 1 268

**Name: count, dtype: int64**

****

**Accuracy: 0.76**

Precision recall f1-score support

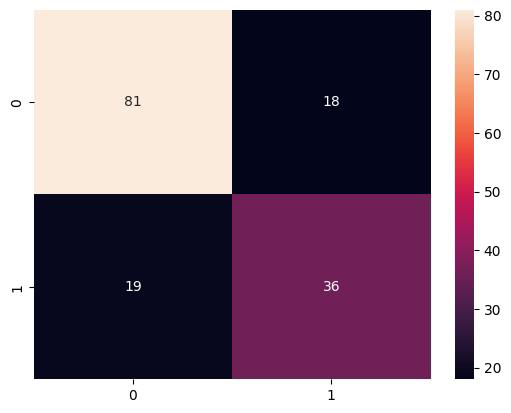
0 0.81 0.82 0.81 99

1 0.67 0.65 0.66 55

Accuracy 0.76 154

Macro avg 0.74 0.74 0.74 154

Weighted avg 0.76 0.76 0.76 154

****

**Features Used in dataset:**

**AI diabetes prediction system databases typically contain a variety of features, including:**

**\* \*\*Demographic features:\*\*** These features include the patient’s age, sex, race, and ethnicity.

**\* \*\*Medical history features**:\*\* These features include the patient’s history of diabetes, prediabetes, and other medical conditions, such as heart disease, stroke, and high blood pressure.

**\* \*\*Lifestyle features:\*\*** These features include the patient’s weight, height, body mass index (BMI), diet, exercise habits, and smoking status.

**\* \*\*Laboratory data**:\*\* This data may include blood glucose levels, cholesterol levels, blood pressure, and other laboratory results.

**In addition to these core features, some AI diabetes prediction system databases may also contain other features, such as:**

**\* \*\*Genetic data:\*\*** This data can be used to identify genetic risk factors for diabetes.

**\* \*\*Wearable device data:\*\*** This data can be collected from wearable devices, such as smartwatches and fitness trackers, and can include information such as heart rate, sleep patterns, and activity levels.

**\* \*\*Social media data:\*\*** This data can be collected from social media platforms, such as Twitter and Facebook, and can be used to identify factors such as social isolation and stress levels, which may be associated with an increased risk of diabetes.

**By combining these different types of data, AI diabetes prediction system databases can be used to develop models that can accurately predict a patient’s risk of developing diabetes.**

**Here are some specific examples of features that might be found in an AI diabetes prediction system database:**

**\* Age**: Age is a strong risk factor for diabetes. The risk of developing diabetes increases with age, especially after the age of 45.

**\*Sex:** Women are more likely to develop diabetes than men.

**\*Race and ethnicity**: African Americans, Hispanics/Latinos, American Indians, and Asian Americans are more likely to develop diabetes than Caucasians.

**\*Family history:** People with a family history of diabetes are more likely to develop the disease themselves.

**\*Body mass index (BMI):**People who are overweight or obese are more likely to develop diabetes.

**\*Blood pressure:**High blood pressure is a risk factor for diabetes.

**\*Cholesterol levels**: High cholesterol levels are a risk factor for diabetes.

**\*Blood sugar levels:**High blood sugar levels are a sign of diabetes.

**Other features that might be found in an AI diabetes prediction system database include:**

**\*Medications:**The types and doses of medications that the patient is taking.

**\*Diet:**The patient’s typical diet, including the types and quantities of food and beverages consumed.

**\*Exercise:**The patient’s exercise habits, including the frequency, intensity, and duration of exercise.

**\*Smoking status:** Whether or not the patient smokes.

**\*Social media data:** The patient’s social media activity, such as the number of friends they have, the types of posts they share, and the people they interact with.

**By combining these different types of data, AI diabetes prediction systems can be used to develop models that can accurately predict a patient’s risk of developing diabetes. This information can then be used to help patients take steps to prevent or manage the disease.**

**CONCLUSION:**

The conclusion about an AI-based Diabetes Prediction System project would depend on the specific goals,outcomes, and performance metrics of the project. Generally, a conclusion might include:

Effectiveness: Assess how well the AI system predicted diabetes. Evaluate metrics like accuracy, sensitivity, specificity, and F1-score.

Data Quality: Consider the quality and quantity of data used for training and testing the AI model. Data plays a crucial role in the system's performance.

User Feedback: Incorporate feedback from healthcare professionals and patients who interacted with the system.

Ethical Considerations: Reflect on the ethical and privacy aspects of using AI in healthcare, including data security and patient consent.

Scalability: Discuss the potential for scaling the system to a larger population or integrating it into healthcare settings.

Future Improvements: Suggest possible enhancements or future directions for the project, such as incorporating more data sources, improving the user interface, or expanding the scope to include related health conditions.