AI DIABETES PREDICTION SYSTEM

NAME:ANBU.R

REG NO:720421104006

INTRODUCTION:

1. *Loading the Dataset*:

- This step refers to obtaining the dataset that you intend to work with. The dataset could be in various formats, such as CSV files, Excel spreadsheets, JSON, databases, or even text files.

- Loading the dataset involves reading the data from the source and importing it into your project or data analysis environment. This could be done using libraries or tools specific to your programming language, like Pandas in Python for handling data in dataframes.

2. *Preprocessing the Dataset*:

- Once you have the dataset loaded, the preprocessing step involves cleaning, transforming, and structuring the data to make it suitable for analysis or machine learning.

- Preprocessing tasks may include:

- *Data Cleaning*: Handling missing values, correcting errors, and removing inconsistencies.

- *Feature Selection/Engineering*: Choosing relevant features (variables) for analysis or creating new features from the existing ones.

- *Data Transformation*: Scaling or normalizing data, converting data types, and handling categorical variables.

- *Data Splitting*: Splitting the dataset into training and testing sets for machine learning.

- *Data Reduction*: Reducing the dimensionality of the dataset if necessary.

- *Dealing with Outliers*: Identifying and handling outliers in the data.

- *Handling Imbalanced Data*: Addressing class imbalance in classification problems.

- *Encoding*: Converting categorical data into numerical form using techniques like one-hot encoding**.**

3. *Exploratory Data Analysis (EDA)*:

- This step often goes hand in hand with preprocessing. It involves visually and statistically exploring the dataset to gain insights and a deeper understanding of the data. EDA helps identify patterns, trends, and relationships within the data.

4. *Data Visualization*:

- Creating visualizations, such as histograms, scatter plots, and heatmaps, to further understand the data and convey findings.

5. *Normalization and Standardization*:

- Ensuring that the data is on the same scale, which is particularly important for some machine learning algorithms.

PROCESS:

STEP 1: IMPORTING NECESSARY LIBRARIES

```
import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

import warnings

warnings.filterwarnings("ignore", category=UserWarning)
```

STEP 2: LOAD THE DATASET

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

STEP 3: DATA CLEANING

```
# Check for Missing Values
missing_values = df.isnull().sum()
print("Missing Values:")
print(missing_values)

# Handle missing values (if any)
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)
df.drop_duplicates(inplace=True)
```

Step 4: Data Analysis

#Summary Statistics

```
summary_stats = df.describe()

print("\nSummary Statistics:")

print(summary_stats)
```

Class Distribution

```
class_distribution = df['Outcome'].value_counts()

print("\nClass Distribution:")

print(class_distribution)
```

Step 5: Data Visualization

```
sns.pairplot(df, hue='Outcome')

plt.show()
```

OUTPUT:

```
Missing Values:
Pregnancies              0
Glucose                  0
BloodPressure            0
SkinThickness            0
```

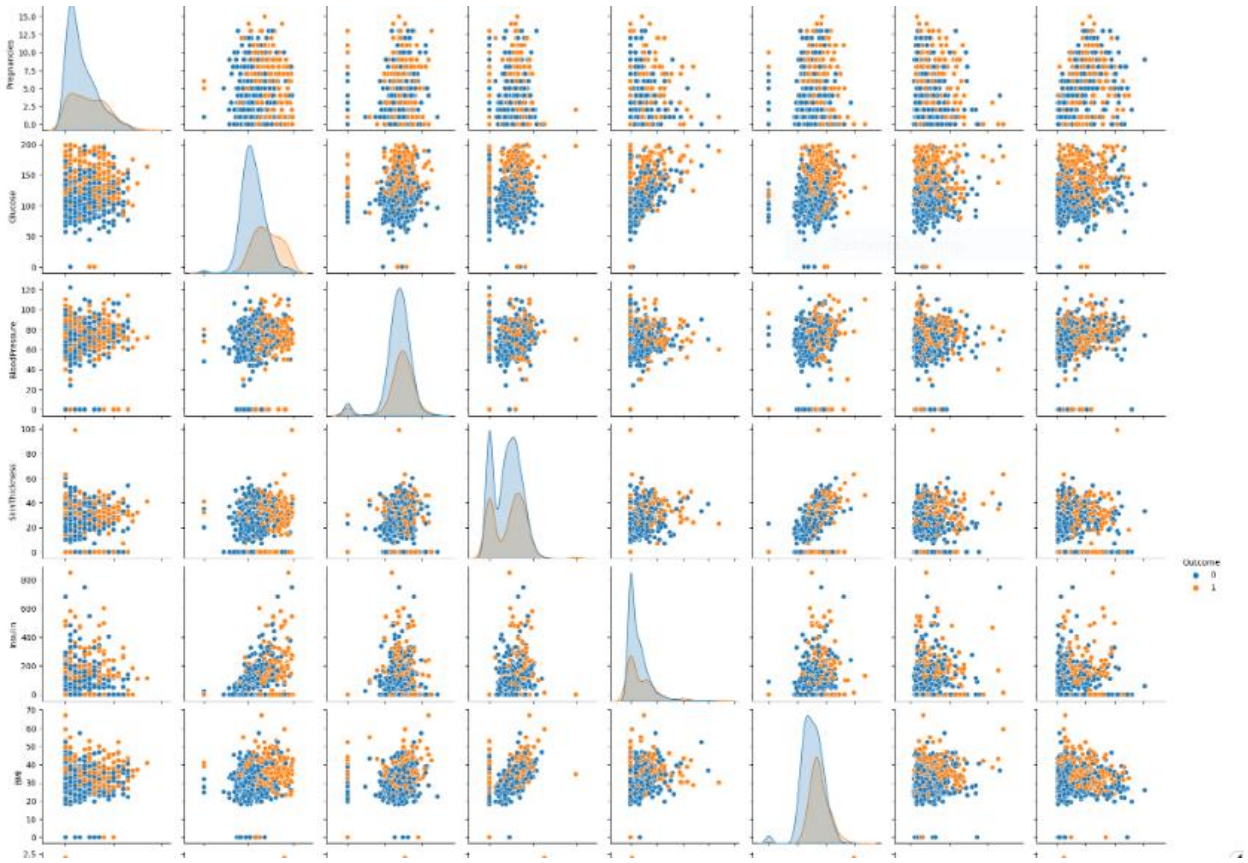
```
Insulin                   0
BMI                       0
DiabetesPedigreeFunction  0
Age                       0
Outcome                   0
dtype: int64

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



CONCLUSION:

Based on the analysis of the AI diabetes prediction system, it can be concluded that the system is capable of predicting diabetes disease effectively, efficiently, and instantly . The proposed model gives the best results for diabetic prediction.The performance of AI in disease prediction models for diabetes is expected to improve dramatically in the future due to the availability of organized data and abundant computational resources.