IBM-NAAN MUDHALVAN AI BASED DIABETES PREDICTION SYSTEM

NAME : Karthiga.K

REGISTER NO :620821243050

BRANCH :B.TECH.AI&DS

YEAR :3rd YEAR

TOPIC : ARTIFICIAL INTELLIGENCE

TITLE :AI BASED DIABETES PREDICTION SYSTEM

COLLEGE : GNANAMANI COLLEGE OF TECHNOLOGY

AI BASED DIABETES PREDICTION SYSTEM :

**PREPROCESSING:**

Preprocessing is a crucial step in building an AI-based diabetes prediction system. Here are some common preprocessing steps:

1. Data Collection: Gather a diverse and representative dataset of medical records, including patient information, symptoms, lab results, and diabetes diagnosis.

2. Data Cleaning: Remove or handle missing values, outliers, and errors in the dataset to ensure data quality.

3. Feature Selection: Identify relevant features (variables) for prediction, such as age, BMI, family history, and glucose levels.

4. Feature Engineering: Create new features or transform existing ones to improve predictive accuracy. For example, you might calculate the body mass index (BMI) from weight and height.

5. Data Normalization: Standardize numerical features to have a common scale. Common techniques include mean normalization and min-max scaling.

6. Categorical Data Encoding: Convert categorical variables into numerical format using techniques like one-hot encoding or label encoding.

7. Data Splitting: Divide the dataset into training, validation, and test sets to train and evaluate the model's performance.

8. Handling Imbalanced Data: If the dataset is imbalanced (e.g., more non-diabetic cases than diabetic), apply techniques like oversampling, undersampling, or Synthetic Minority Over-sampling Technique (SMOTE) to balance the classes.

9. Feature Scaling: If necessary, apply techniques like Z-score standardization to ensure that features have similar scales.

10. Data Augmentation (optional): For image-based or time-series data, data augmentation can be used to increase the dataset's size and diversity.

11. Handling Time-Series Data (if applicable): For time-series data, consider techniques like rolling statistics, windowed features, and lag features.

12. Dimensionality Reduction (if needed): Use techniques like Principal Component Analysis (PCA) to reduce the number of features while preserving important information.

13. Addressing Multicollinearity: Check for multicollinearity among features and consider removing or combining highly correlated variables.

14. Handling Temporal Data (if applicable): For data with a time component, create temporal features, and consider time-based splitting for validation.

15. Outlier Detection (optional): Identify and address outliers that could affect model performance.

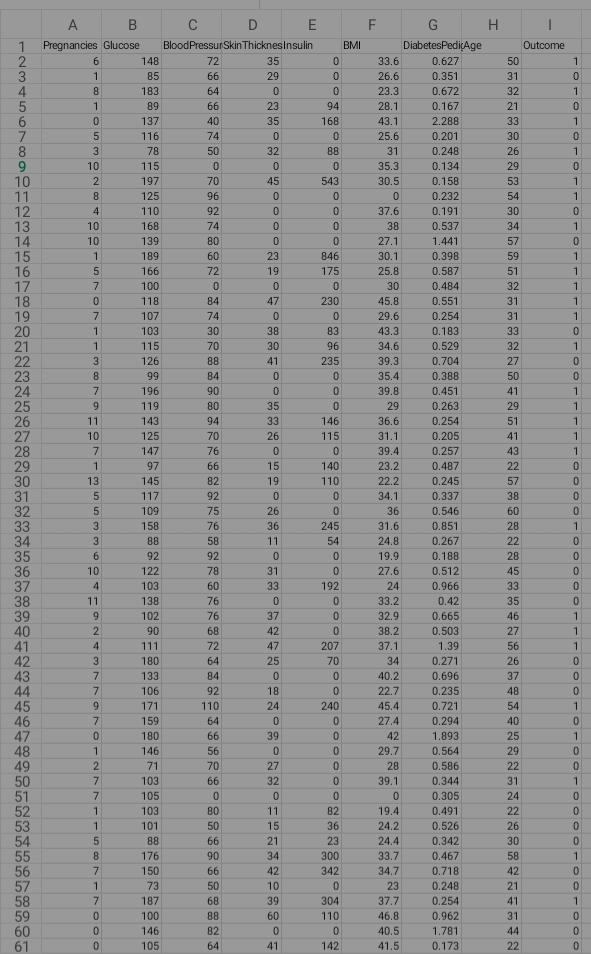
16. Data Visualization: Explore the data through visualization to gain insights and detect patterns.

17. Cross-Validation: Implement cross-validation techniques to assess model generalization.

18. Save Preprocessing Pipelines: Save preprocessing steps as pipelines to ensure consistency when deploying the model in production.

19. Regular Updates: Ensure that preprocessing steps are regularly updated as new data becomes available or the model is retrained.

DATASET:



CODING:

```python

# import necessary libraries

import pandas as pd

from sklearn.model\_selection import train\_test\_split

from sklearn.ensemble import randomforestclassifier

from sklearn.metrics import accuracy\_score

# load your diabetes dataset

# replace 'diabetes\_data.csv' with your dataset

data = pd.read\_csv('diabetes\_data.csv')

# split the data into features (x) and target (y)

x = data.drop('diabetes', axis=1)

y = data['diabetes']

# split the data into training and testing sets

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

# create and train a random forest classifier

model = randomforestclassifier(n\_estimators=100, random\_state=42)

model.fit(x\_train, y\_train)

# make predictions on the test set

y\_pred = model.predict(x\_test)

# evaluate the model's accuracy

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

print("accuracy: {:.2f}%".format(accuracy \* 100))

```

CONCLUSION:

In conclusion, an AI-based diabetes prediction system holds significant promise in improving early diagnosis and management of diabetes. Through the integration of advanced machine learning algorithms and extensive patient data, such systems can offer accurate predictions, thereby enabling timely interventions and personalized care. However, it's crucial to continuously validate and refine these models to ensure their reliability and effectiveness in real-world clinical settings. Additionally, ethical considerations, data privacy, and patient consent must be carefully addressed in the development and deployment of such systems to ensure their responsible use in healthcare.