AI BASED DIABETES PREDICTION SYSTEM

TEAM MEMBER:

NAME: M.RUDHRESHKUMAR

REGISTER NO: 410121243045

DOCUMENT SUBMISSION

PHASE 3 ---

We are created step-by-step guide to loading and preprocessing a dataset for a diabetes prediction system:

1. Loading the Dataset

First, you'll need to download the dataset from Kaggle or any other source. Once you have the dataset file (e.g., CSV, Excel, etc.), you can load it into your programming environment. If you're using Python, you can utilize libraries like Pandas to load the dataset.

python Copy code

```
import pandas as pd

# Load the dataset (assuming it's a CSV file)

data=pd.read_csv('path_to_your_dataset/diabetes_dataset.csv')
```

2. Exploring the Dataset

Before diving into preprocessing, it's crucial to understand the dataset structure and its contents. You can use methods like data.head() to display the first few rows of the dataset and data.info() to get an overview of the columns and their data types.

3. Handling Missing Values

Check for any missing or null values in the dataset and decide on a strategy to handle them. This might involve imputation (replacing missing values with a statistic like mean, median, or mode) or dropping columns/rows with excessive missing data.

python Copy code

```python
# Check for missing values

data.isnull().sum()

# Handle missing values (if any)

data = data.fillna(data.mean())  # Example: Filling missing values with the mean
```

4. Feature Selection

Identify the features (columns) that are relevant for predicting diabetes. This involves examining the relationship between different features and the target variable (diabetes in this case). You can use statistical methods or domain knowledge to select the most informative features.

5. Feature Scaling

Normalize or standardize the data if the features are on different scales. This step helps in improving the performance of some machine learning algorithms.

python code

```
from sklearn.preprocessing import StandardScaler

# Initialize the scaler

scaler = StandardScaler()

# Scale the features

data[relevant_features] = scaler.fit_transform(data[relevant_features])
```

6. Splitting the Dataset

Split the data into training and testing sets to train your model on a portion of the data and evaluate its performance on unseen data.

python code

```
from sklearn.model_selection import train_test_split

# Define features and target variable

X = data[relevant_features]
y = data['diabetes_column']

# Split the data

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

7. Building the Model

Finally, once the data is preprocessed, you can choose a machine learning model (like logistic regression, decision trees, random forests, etc.) to train and predict diabetes based on the dataset.