**AI-Based Diabetes Prediction System**

**Problem statement:**

The problem statement is to develop an AI based diabetes prediction system . The objective of an AI-Based Diabetes Prediction System is to accurately assess the risk of an individual developing diabetes based on a comprehensive analysis of relevant health data. This includes factors such as medical history, genetic predisposition, lifestyle choices, and key biomarkers. The system aims to provide early and reliable predictions to empower individuals and healthcare professionals with actionable information for preventive measures or targeted interventions.

**Design Thinking:**

**Dataset:**

**\* Attributes:**

The dataset is likely to include various attributes, which could include demographic information (e.g., age, gender), clinical measurements (e.g., blood glucose levels, insulin levels), anthropometric data (e.g., BMI, waist circumference), and potentially lifestyle factors (e.g., diet, exercise).

**\* Target Variable:**

There is likely to be a target variable that indicates whether an individual has diabetes or not. This variable would be used in supervised learning scenarios to train predictive models.

**\* Instances:**

Each row in the CSV file represents an individual or a case. This could be a patient or participant in a study.

**\* Missing Values:**

It's possible that some cells in the dataset might be empty or contain missing values. Handling missing data is an important step in data preprocessing.

**\* Data Types:**

The data types of the attributes may vary. For example, numerical values (integers or decimals) for measurements, categorical values for attributes like gender, and possibly dates or timestamps.

**\* Data Size:**

The dataset could be of varying size, ranging from a small sample to a large-scale compilation of data.

**\* Source:**

The dataset may originate from various sources, such as clinical trials, research studies, healthcare institutions, or population health surveys.

**Data processing;**

Clean and preprocess the data, handle missing values, and convert categorical features into numerical representations.

\* Check for missing values: isnull() function to identify missing values in the dataset and sum() to count them for each column.

\* Handle missing values: Fill missing values with the mean of the respective columns using the fillna() method.

\* Convert categorical features into numerical representations: One hot Encoding is the best way to convert categorical data into binary vectors. This maps the values to integer values.

**Feature Selection:**

Feature selection is a crucial step in building an effective diabetes prediction model. It involves selecting the most relevant attributes (features) from the dataset to use in the modeling process. This helps improve model performance, reduce computational complexity, and enhance interpretability.

**Model Selection:**

The suitable algorithm for predicting the diabetes is **Random Forest .**Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset. Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output. The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.

**Model Training:**

Train the selected model using the preprocessed data. Feed the training data (features and labels) into the model. The model will adjust its internal parameters to learn the underlying patterns in the data. During training, the algorithm chosen tries to minimize a certain loss function, which measures the difference between its predictions and the true labels in the training data.

**Accuracy:**

Accuracy is a proportional measure of the number of correct predictions over all predictions.

Correct predictions are composed of true positives (TP) and true negatives (TN).All predictions are composed of the entirety of positive (P) and negative (N) examples.P is composed of TP and false positives (FP), and N is composed of TN and false negatives (FN).

Thus, we can define accuracy as **ACC =TP + TNTP + TN + FN + TP =TP + TNP + N.**