AI BASED DIABETES PREDICTION SYSTEM

PHASE 4: DEVELOPMENT PART-2

Data Collection and Preprocessing:

- Gather a dataset that contains relevant features and labels for diabetes prediction. Common datasets for diabetes prediction include the Pima Indian Diabetes dataset, the Diabetes dataset from the UCI Machine Learning Repository, or any other relevant dataset.
- Preprocess the data by cleaning and transforming it. This may include handling missing values, normalizing or scaling features, and encoding categorical variables.

Feature Selection :

- If your dataset has a large number of features, consider performing feature selection to choose the most relevant features. Feature selection can help improve model performance and reduce computational costs.

K NEAREST neighbour classifier:

- It is a supervised learning algorithm,which uses proximity to make classifications about the grouping of an individual point.

Adaboost Algorithm :

- It is an ensemble technique used to improve the predictive accuracy by combining multiple weak models.

Random Forest Classifier :

- It is an ensemble learning technique used for combining numerous classifiers to enhance a models performance

Data Splitting :

- Split your dataset into training and testing sets. A common split is 70-30 or 80-20, where the larger portion is used for training and the smaller one for testing.

Model Training :

- Train the selected machine learning model using the training dataset. You can use Python libraries like scikit-learn for this purpose.

Model Evaluation :

- After training, evaluate the model's performance using the testing dataset. Common evaluation metrics for binary classification tasks include accuracy, precision, recall, F1-score, and the area under the receiver operating characteristic curve (AUC-ROC).
- Consider using techniques like cross-validation to ensure a robust evaluation.

Deployment :

- Once you are satisfied with the model's performance, you can deploy it as part of your AI-based diabetes prediction system. This may involve creating a user interface or integrating the model into a web application, mobile app, or healthcare system.

Continuous Monitoring and Maintenance :

- After deployment, regularly monitor the model's performance and retrain it as needed to ensure it stays accurate over time.