**INNOVATION OF AI BASED**

**DIABETES PREDICTION SYSTEM**

**AI PHASE-3**

**Problem Statement:**

The problem is to build an AI-powered diabetes prediction system that uses machine learning algorithms to analyze medical data and predict the likelihood of an individual developing diabetes.

**Development part:**

-To load, preprocess the dataset and perform different analysis

**1.Download dataset:**

-Download the dataset from the provided Kaggle link:

<https://www.kaggle.com/datasets/mathchi/diabetes-data-set>

**2.Import necessary Libraries:**

-Use Python and import libraries such as numpy,pandas,matplotlib and scikit-learn for data manipulation and visualization.

***import numpy as np***

***import pandas as pd***

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

***from sklearn.preprocessing import StandardScaler***

***from sklearn.linear\_model import LogisticRegression***

***from sklearn.metrics import accuracy\_score, classification\_report, confusion\_matrix***

**3.Load the dataset:**

-Read the dataset into a pandas dataframe.

***from google.colab import files***

***uploaded = files.upload()***

***import pandas as pd***

***import io***

***df = pd.read\_csv(io.BytesIO(uploaded['diabetes.csv']))***

***print(df)***

**4.Explore the dataset:**

-Check the first few rows of the dataset to understand what we are dealing with.

-Examine the column names and data types.

***print(df.head())***

***print(df.info())***

**5.Spilting the data:**

-Split the dataset into training and testing sets:

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

***X = df.drop('target', axis=1)***

***y = df['target']***

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

**6.Model Training:**

**-**Train a machine learning model to predict diabetes. For example, you can use a linear regression model:

***from sklearn.linear\_model import LinearRegression***

***model = LinearRegression()***

***model.fit(X\_train, y\_train)***

**7.Model Evaluation:**

-Evaluate the model's performance:

***from sklearn.metrics import mean\_squared\_error, r2\_score***

***y\_pred = model.predict(X\_test)***

***mse = mean\_squared\_error(y\_test, y\_pred)***

***r2 = r2\_score(y\_test, y\_pred)***

***print(f"Mean Squared Error: {mse}")***

***print(f"R-squared: {r2}")***

**Code:**

-To predict diabetes in patients using Python, you can use a machine learning model and a dataset. Here’s the code to perform a logistic regression model from the scikit-learn library using the given dataset we have loaded first.

-First, make sure you have scikit-learn installed. You can install it using ‘pip’ if you haven't already:

***Pip install scikit-learn***

-Now let's look at the code:

# Import necessary libraries

***import numpy as np***

***import pandas as pd***

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

***from sklearn.preprocessing import StandardScaler***

***from sklearn.linear\_model import LogisticRegression***

***from sklearn.metrics import accuracy\_score, classification\_report, confusion\_matrix***

# Load the diabetes dataset

***from google.colab import files***

***uploaded = files.upload()***

***import io***

***df = pd.read\_csv(io.BytesIO(uploaded['diabetes.csv']))***

***print(df)***

# Split the data into features (X) and the target (y)

***X = df.drop("Outcome", axis=1)***

***y = df["Outcome"]***

# 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)***

# Standardize the features

***scaler = StandardScaler()***

***X\_train = scaler.fit\_transform(X\_train)***

***X\_test = scaler.transform(X\_test)***

# Create and train a logistic regression model

***model = LogisticRegression()***

***model.fit(X\_train, y\_train)***

# Make predictions on the test set

***y\_pred = model.predict(X\_test)***

# Evaluate the model

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

***confusion = confusion\_matrix(y\_test, y\_pred)***

***classification\_report\_str = classification\_report(y\_test, y\_pred)***

***print(f"Accuracy: {accuracy}")***

***print(f"Confusion Matrix:\n{confusion}")***

***print("Classification Report:")***

***print(classification\_report\_str)***

**Summary:**

A Diabetes Prediction System using AI in Python is an advanced healthcare application that leverages artificial intelligence (AI) techniques to predict the likelihood of an individual developing diabetes. This system is designed to assist healthcare professionals and individuals in early detection and management of diabetes, ultimately leading to better health outcomes.

**Conclusion:**

In conclusion, a Diabetes Prediction System using AI in Python is a valuable tool in the field of healthcare. It empowers healthcare professionals and individuals to make informed decisions, enabling early detection and proactive management of diabetes, ultimately contributing to better public health outcomes.

However, challenges such as data quality, privacy concerns, and the need for model interpretability need to be carefully addressed in the development and deployment of such systems.

In the long run, AI-based diabetes prediction systems have the potential to revolutionize healthcare by shifting the focus from reactive treatment to proactive prevention. These systems are part of a broader trend in AI applications that aim to improve public health and individual well-being, ultimately contributing to a healthier society.