## AI Based Diabetes Prediction System

## PHASE 4(Development Part 2):

## Dataset link:  https://www.kaggle.com/datasets/mathchi/diabetes-data-set

## Tools used:

## Pandas

## NumPy

## Matplotlib

## Seaborn

## Scikit-Learn

## Mechine learning Algorithm:

## Logistic Regression

## Training the Model:

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

## Code:

# Import necessary libraries

import pandas as pd

import numpy as np

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 your dataset

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

# Separate features (X) and target variable (y)

X = data.drop('Outcome', axis=1)

y = data['Outcome']

# Split the dataset into training and testing sets (80% training, 20% testing)

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

# Standardize features (optional but often helpful)

scaler = StandardScaler()

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

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

# Create a Logistic Regression model

model = LogisticRegression(random\_state=42)

# Train the model

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)

print(f'Accuracy: {accuracy:.2f}')

# Print a classification report and confusion matrix

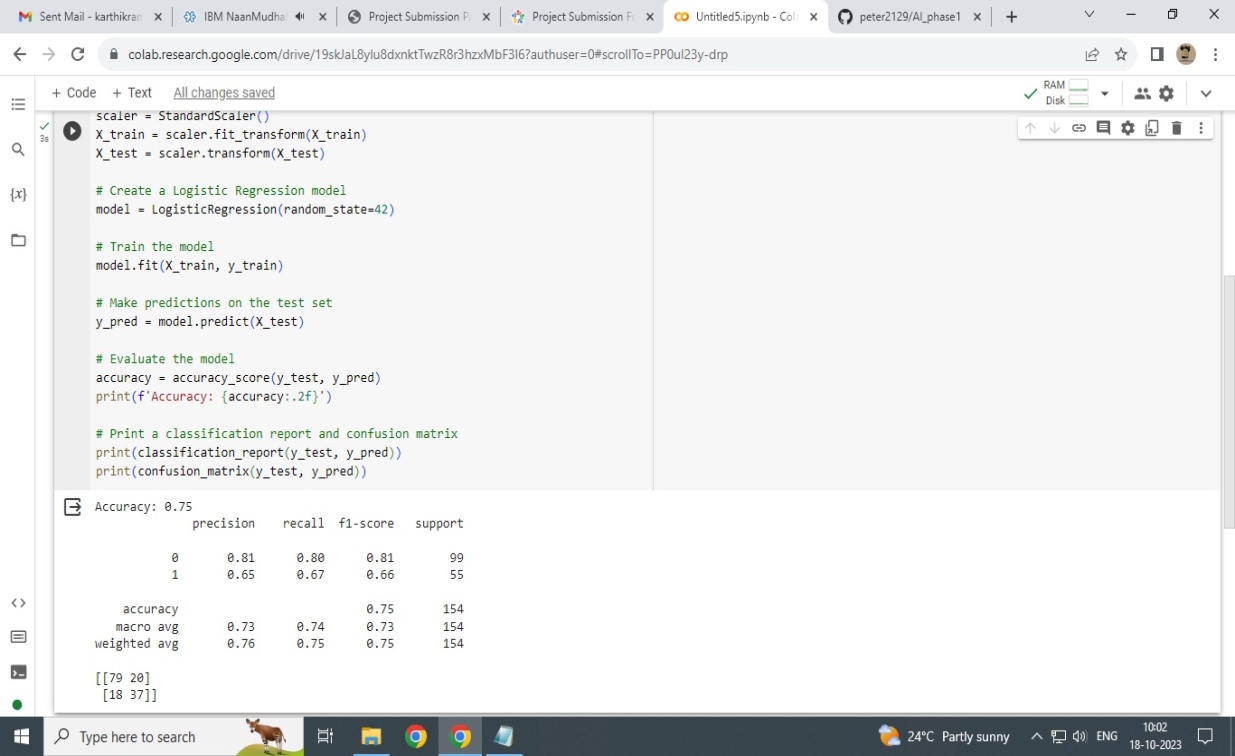
print(classification\_report(y\_test, y\_pred))

print(confusion\_matrix(y\_test, y\_pred))

## Sample Output:

## 

## 

****

**Visualization:**

import seaborn as sns

import matplotlib.pyplot as plt

# Generate the confusion matrix

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

# Create a heatmap

plt.figure(figsize=(6, 4))

sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', cbar=False,

            xticklabels=['No Diabetes', 'Diabetes'],

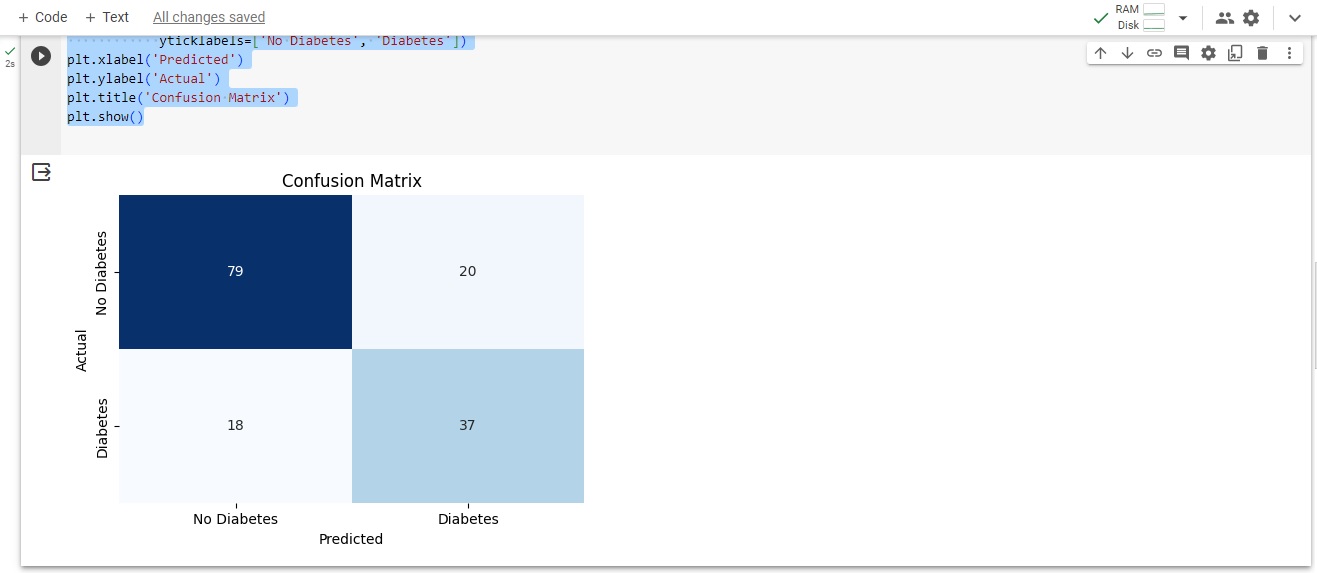
            yticklabels=['No Diabetes', 'Diabetes'])

plt.xlabel('Predicted')

plt.ylabel('Actual')

plt.title('Confusion Matrix')

plt.show()

****