****

**Assessment Report**

on

**“Diagnose Diabetes:** Use patient medical records to classify if an individual has diabetes.**”**

submitted as partial fulfillment for the award of

**BACHELOR OF TECHNOLOGY**

**DEGREE**

SESSION 2024-25

in

**CSE (AI&ML)**

By

Kartik Kumar (202401100400105, CSE(AI&ML)-B)

**Under the supervision of**

“Abhishek Shukla”

**KIET Group of Institutions, Ghaziabad**

**18th May, 2025**

Problem :

Diagnose Diabetes: Use patient medical records to classify if an individual has diabetes.

Solution :

Diabetes is a life-threatening but manageable chronic illness that occurs when the body cannot properly process blood sugar (glucose). Early detection and diagnosis are critical in preventing complications such as heart disease, kidney damage, and nerve issues.

However, diagnosing diabetes based on physical symptoms alone is challenging because its early stages are often asymptomatic. This creates a need for **automated diagnostic tools** that can assist healthcare providers by analyzing **medical records** such as:

* Blood glucose levels
* Body Mass Index (BMI)
* Age
* Insulin levels
* Blood pressure
* Pregnancies

In this project, the goal is to build a **classification model** using machine learning that can predict whether a person is diabetic (1) or not (0) based on these medical attributes. This model learns patterns from historical patient data and uses that knowledge to diagnose new, unseen cases.

Such a predictive tool can:

* Help doctors with faster screening,
* Reduce diagnostic errors,
* And improve patient outcomes by enabling **early intervention**.

Code :

# STEP 1: Install required libraries

!pip install pandas scikit-learn seaborn matplotlib

# STEP 2: Import libraries

import pandas as pd

import numpy as np

import seaborn as sns

import matplotlib.pyplot as plt

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

from sklearn.preprocessing import StandardScaler

from sklearn.ensemble import RandomForestClassifier

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

from google.colab import files

# STEP 3: Upload your CSV file

print("Upload your '2. Diagnose Diabetes.csv' file...")

uploaded = files.upload()

# STEP 4: Load CSV file

filename = "2. Diagnose Diabetes.csv"  # Make sure file name matches

df = pd.read\_csv(filename)

# STEP 5: Display basic info

print("First 5 rows:")

print(df.head())

# STEP 6: Convert to numeric (if needed) and clean

df = df.apply(pd.to\_numeric, errors='coerce')

df.dropna(inplace=True)

# STEP 7: Prepare features and labels

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

y = df['Outcome']

# STEP 8: Scale features

scaler = StandardScaler()

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

# STEP 9: Train-test split

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

# STEP 10: Train model

model = RandomForestClassifier(random\_state=42)

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

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

# STEP 11: Evaluation

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

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

plt.title("Confusion Matrix")

plt.xlabel("Predicted")

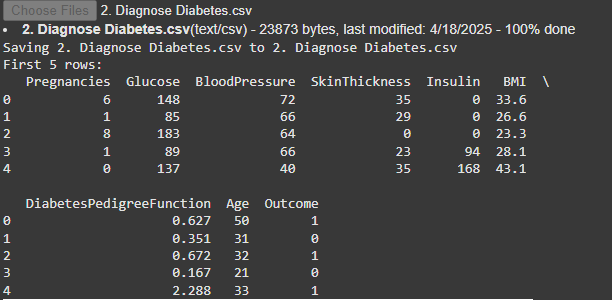
plt.ylabel("Actual")

plt.show()

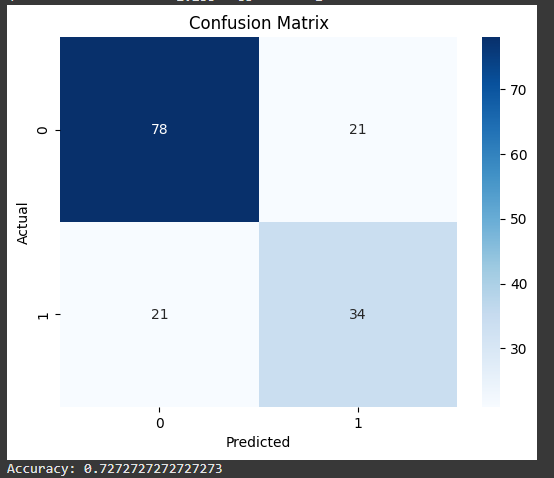
# Metrics

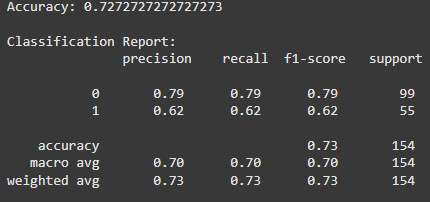
print("Accuracy:", accuracy\_score(y\_test, y\_pred))

print("\nClassification Report:\n", classification\_report(y\_test, y\_pred))



SCREENSHOTS :





Refrence :

Dataset Source: Pima Indians Diabetes Database

Code developed using: Python, scikit-learn, seaborn, matplotlib, Google Colab

Tools: Jupyter/Colab Notebook, Google Colab File Upload feature