www.kiet.edu
Delhi-NCR, Ghaziabad

KIET
GROUP OF INSTITUTIONS
Connecting Life with Learning

Assessment report on

“Diagnose Diabetes”

submitted as partial fulfillment for the award of

BACHELOR OF TECHNOLOGY

DEGREE

SESSION 2024-25

In

Introduction to AI

By

Pragya (202401100400135)

Under the supervision of

“Mr. Abhishek shukla ”

KIET Group of Institutions, Ghaziabad

Affiliated to

Dr. A.P.J. Abdul Kalam Technical University, Lucknow

(Formerly UPTU)

🩺 Problem Statement:

Build a machine learning model to **predict if a person has diabetes** based on medical data using the **Naive Bayes algorithm**.

📊 Dataset Info:

- **768 patients**
- **8 input features** like Glucose, BMI, Age, etc.
- **1 target label**:
  - `0` = No diabetes
  - `1` = Has diabetes

🎯 Goal:

Train a **Naive Bayes classifier** to predict diabetes from patient data.

🤖 Why Naive Bayes?

- Simple and fast
- Works well with small datasets
- Based on probability

✅ Process:

1. Load and clean data
2. Split into training and test sets

3. Train Naive Bayes model

4. Predict outcomes

5. Evaluate with accuracy & confusion matrix

Approach to Solve the Problem

1. **Data Loading**
   The dataset was loaded using pandas to work with it in tabular format.

2. **Data Exploration**
   Basic checks were performed to understand the structure of the dataset, including shape, column names, and missing values.

3. **Feature and Target Separation**

   - Features (X): All input columns such as Glucose, BMI, Age, etc.

   - Target (y): The 'Outcome' column, indicating whether the patient has diabetes (1) or not (0).

4. **Train-Test Split**
   The data was split into training and test sets (80% training, 20% testing) using `train_test_split`.

5. **Model Training**
   A Gaussian Naive Bayes model was trained using the training data.

6. **Prediction**
   The trained model was used to make predictions on the test set.

7. **Model Evaluation**
   The model's performance was evaluated using:

   - Accuracy score

   - Classification report (precision, recall, f1-score)

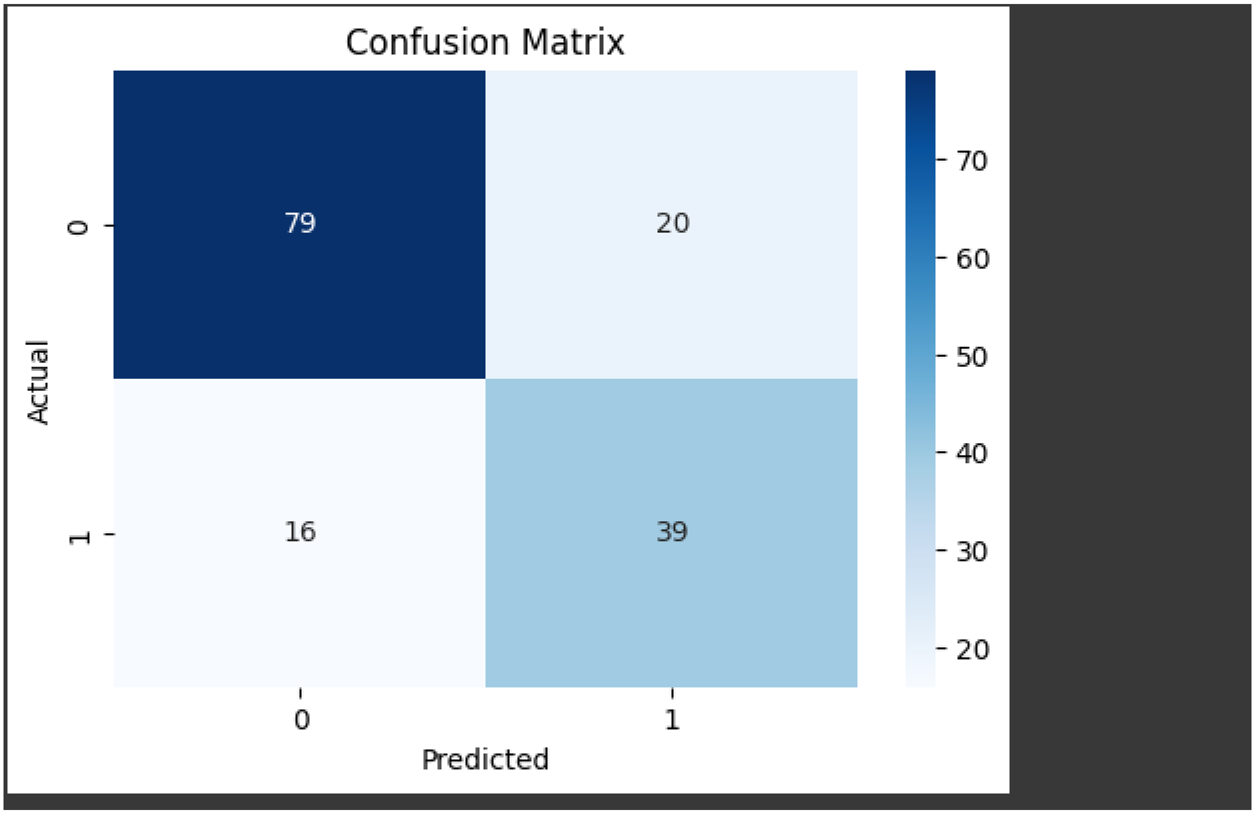
   - Confusion matrix to understand prediction errors

```
from google.colab import files
uploaded = files.upload()

# Step 1: Import necessary libraries
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import classification_report, confusion_matrix,
accuracy_score

# Step 2: Load the dataset
df = pd.read_csv('/content/2. Diagnose Diabetes.csv')

# Step 3: Basic info
print("Dataset shape:", df.shape)
print(df.head())
print("\nMissing values:\n", df.isnull().sum())

# Step 4: Split features and target
X = df.drop('Outcome', axis=1)
y = df['Outcome']

# Step 5: Train-test split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)

# Step 6: Initialize and train Naive Bayes model
model = GaussianNB()
model.fit(X_train, y_train)

# Step 7: Make predictions
y_pred = model.predict(X_test)

# Step 8: Evaluate the model
accuracy = accuracy_score(y_test, y_pred)
print(f"\n✅ Accuracy: {accuracy:.2f}")

print("\n📊 Classification Report:")
```

```python
print(classification_report(y_test, y_pred))

# Step 9: Confusion Matrix
conf_matrix = confusion_matrix(y_test, y_pred)
plt.figure(figsize=(6, 4))
sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='Blues')
plt.title("Confusion Matrix")
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.show()
```

2. Diagnose Diabetes.csv(text/csv) - 23873 bytes, last modified: 4/18/2025 - 100% done

```
Saving 2. Diagnose Diabetes.csv to 2. Diagnose Diabetes.csv
Dataset shape: (768, 9)
   Pregnancies  Glucose  BloodPressure  SkinThickness  Insulin   BMI  \
0            6      148             72             35        0  33.6
1            1       85             66             29        0  26.6
2            8      183             64              0        0  23.3
3            1       89             66             23       94  28.1
4            0      137             40             35      168  43.1

   DiabetesPedigreeFunction  Age  Outcome
0                     0.627   50        1
1                     0.351   31        0
2                     0.672   32        1
3                     0.167   21        0
4                     2.288   33        1

Missing values:
 Pregnancies                 0
Glucose                     0
BloodPressure               0
SkinThickness               0
Insulin                     0
BMI                         0
DiabetesPedigreeFunction    0
Age                         0
Outcome                     0
dtype: int64

✅ Accuracy: 0.77

📊 Classification Report:
              precision    recall  f1-score   support

           0       0.83      0.80      0.81        99
           1       0.66      0.71      0.68        55

    accuracy                           0.77       154
   macro avg       0.75      0.75      0.75       154
weighted avg       0.77      0.77      0.77       154
```



	0	1
0	79	20
1	16	39