GROUP 14

Problem Statement-Create a classification model to predict the likelihood of diabetes based on health data. Visualize model accuracy.

Submitted by:

Shivam Kumar (202401100300233)

Shivani Yadav (202401100300235)

Somya Mishra (202401100300250)

Yuvraj Singh Taniya (202401100300292)

Year & Section:

1st Year-CSE(AI)-D

Problem Statement Explained:

You are working with a **binary classification** problem where the goal is to predict whether a patient has **diabetes** or not, based on various health indicators. The dataset is the famous **Pima Indians Diabetes Dataset**, which includes the following features:

- **Pregnancies**: Number of times pregnant
- **Glucose**: Plasma glucose concentration
- **Blood Pressure**: Diastolic blood pressure (mm Hg)
- **Skin Thickness**: Triceps skinfold thickness (mm)
- **Insulin**: 2-Hour serum insulin (mu U/ml)
- **BMI**: Body mass index (weight in kg/(height in m)^2)
- **Diabetes Pedigree Function**: A function that scores the likelihood of diabetes based on family history
- **Age**: Age in years
- **Outcome**: Target variable (0 = no diabetes, 1 = diabetes)

Methodology:

The following steps were taken to develop a diabetes prediction model:

1. **Data Loading & Exploration**:
   The dataset was uploaded in Google Colab and examined using basic statistics and info commands to understand its structure and identify anomalies.
2. **Data Cleaning**:
   Zero values in key columns (Glucose, BloodPressure, SkinThickness, Insulin, BMI) were treated as missing and replaced with the median using imputation.
3. **Visualization**:
   Histograms and a correlation heatmap were created to understand feature distributions and relationships. The outcome variable's class balance was also visualized.
4. **Modeling with KNN**:
   A K-Nearest Neighbors classifier (k=5) was trained using an 80-20 train-test split.
5. **Evaluation**:
   Model performance was assessed using accuracy, a classification report, and a confusion matrix, all supported with visualizations.

Code:

```
# Upload and preprocess + classify with KNN and visualization
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.impute import SimpleImputer
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
import io

# For file upload (Google Colab)
from google.colab import files
uploaded = files.upload()

# Load uploaded file
for file_name in uploaded.keys():
    df = pd.read_csv(io.StringIO(uploaded[file_name].decode('utf-8')))

# Show initial info
print("\nInitial Data Information:")
print(df.info())
print("\nSummary Statistics:")
print(df.describe())

# Show missing values count before cleaning
print("\nMissing values before replacement:")
print((df == 0).sum())

# Visualize missing values as 0s
plt.figure(figsize=(10, 4))
sns.heatmap(df.replace(0, np.nan).isnull(), cbar=False, cmap='viridis')
```

```
plt.title("Missing Values Heatmap (0s treated as NaN)")
plt.show()

# Replace zero values with NaN in specific columns
columns_with_zeros = ['Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI']
df[columns_with_zeros] = df[columns_with_zeros].replace(0, np.nan)

# Impute missing values using median
imputer = SimpleImputer(strategy='median')
df[columns_with_zeros] = imputer.fit_transform(df[columns_with_zeros])

# Show after cleaning
print("\nMissing values after imputation:")
print(df.isnull().sum())

# Plot distributions of features
df.hist(figsize=(12, 10), edgecolor='black', bins=20)
plt.suptitle("Feature Distributions", fontsize=16)
plt.show()

# Correlation heatmap
plt.figure(figsize=(10, 8))
sns.heatmap(df.corr(), annot=True, cmap='coolwarm', fmt=".2f")
plt.title("Correlation Heatmap")
plt.show()

# Outcome class distribution
sns.countplot(data=df, x='Outcome')
plt.title("Diabetes Outcome Distribution")
plt.xlabel("Outcome (0 = No Diabetes, 1 = Diabetes)")
plt.ylabel("Count")
plt.show()

# Feature-target split
```

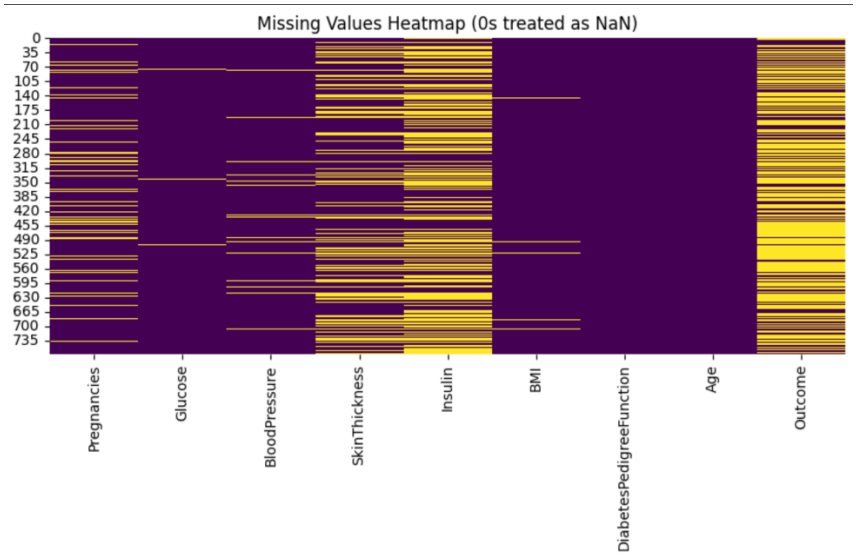
```
X = df.drop('Outcome', axis=1)
y = df['Outcome']

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

# KNN Classification model
knn = KNeighborsClassifier(n_neighbors=5)
knn.fit(X_train, y_train)

# Predictions and Evaluation
y_pred = knn.predict(X_test)

# Accuracy
acc = accuracy_score(y_test, y_pred)
print("\nKNN Model Accuracy:", acc)

# Classification report
print("\nClassification Report:")
print(classification_report(y_test, y_pred))

# Confusion matrix
cm = confusion_matrix(y_test, y_pred)
plt.figure(figsize=(6, 4))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', xticklabels=[0, 1], yticklabels=[0, 1])
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.title("Confusion Matrix")
plt.show()
```

Output/Result:

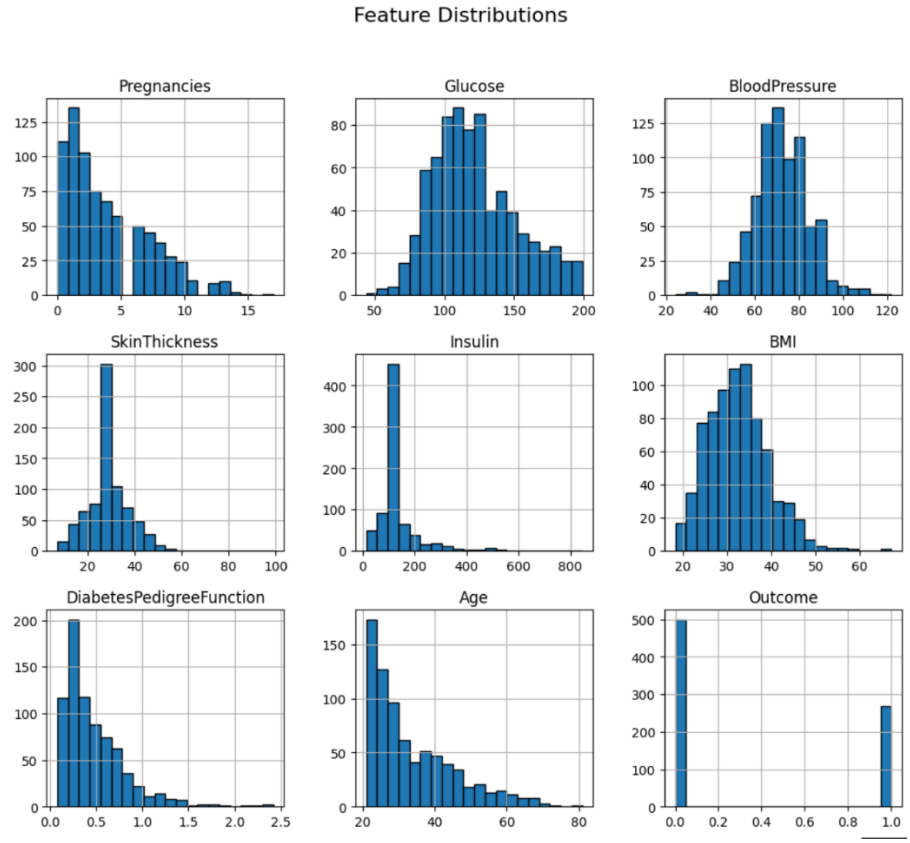
```
Choose Files  diabetes.csv
• diabetes.csv(text/csv) - 23873 bytes, last modified: 5/27/2025 - 100% done
Saving diabetes.csv to diabetes (1).csv

Initial Data Information:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 768 entries, 0 to 767
Data columns (total 9 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Pregnancies               768 non-null    int64
 1   Glucose                   768 non-null    int64
 2   BloodPressure             768 non-null    int64
 3   SkinThickness             768 non-null    int64
 4   Insulin                   768 non-null    int64
 5   BMI                       768 non-null    float64
 6   DiabetesPedigreeFunction  768 non-null    float64
 7   Age                       768 non-null    int64
 8   Outcome                   768 non-null    int64
dtypes: float64(2), int64(7)
memory usage: 54.1 KB
None
```

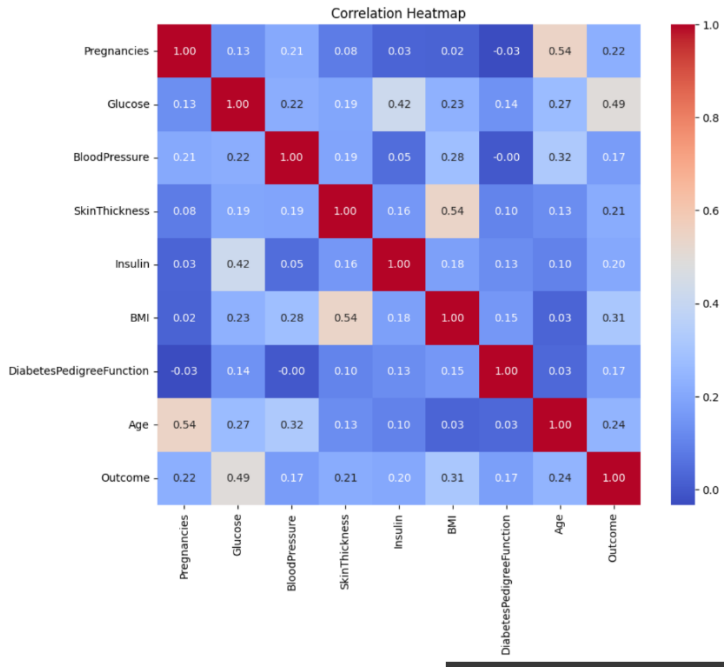
```
Summary Statistics:
       Pregnancies     Glucose  BloodPressure  SkinThickness     Insulin  \
count   768.000000  768.000000     768.000000     768.000000  768.000000
mean      3.845052  120.894531      69.105469      20.536458   79.799479
std       3.369578   31.972618      19.355807      15.952218  115.244002
min       0.000000    0.000000       0.000000       0.000000    0.000000
25%       1.000000   99.000000      62.000000       0.000000    0.000000
50%       3.000000  117.000000      72.000000      23.000000   30.500000
75%       6.000000  140.250000      80.000000      32.000000  127.250000
max      17.000000  199.000000     122.000000      99.000000  846.000000

              BMI  DiabetesPedigreeFunction         Age     Outcome
count  768.000000                768.000000  768.000000  768.000000
mean    31.992578                  0.471876   33.240885    0.348958
std      7.884160                  0.331329   11.760232    0.476951
min      0.000000                  0.078000   21.000000    0.000000
25%     27.300000                  0.243750   24.000000    0.000000
50%     32.000000                  0.372500   29.000000    0.000000
75%     36.600000                  0.626250   41.000000    1.000000
max     67.100000                  2.420000   81.000000    1.000000
```

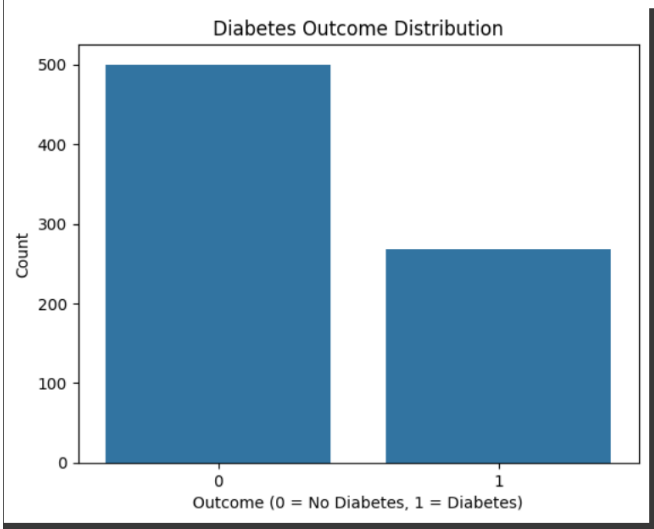
```
Missing values before replacement:
Pregnancies                 111
Glucose                       5
BloodPressure                35
SkinThickness               227
Insulin                     374
BMI                          11
DiabetesPedigreeFunction      0
Age                           0
Outcome                     500
dtype: int64
```



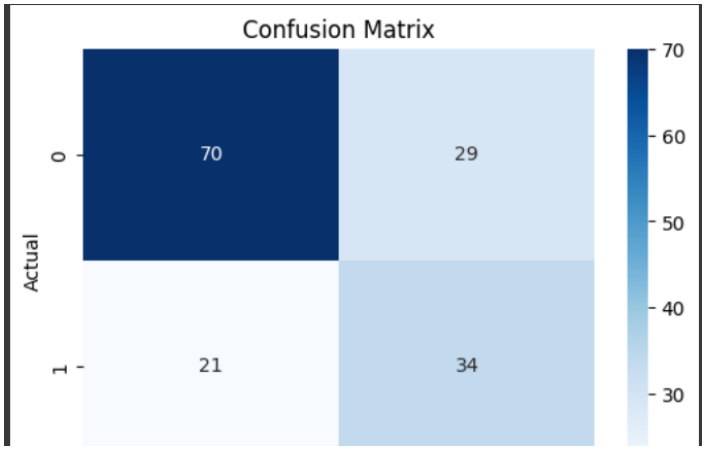

```
Missing values after imputation:
Pregnancies                 0
Glucose                     0
BloodPressure               0
SkinThickness               0
Insulin                     0
BMI                         0
DiabetesPedigreeFunction    0
Age                         0
Outcome                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








```
KNN Model Accuracy: 0.6753246753246753

Classification Report:
              precision    recall  f1-score   support

           0       0.77      0.71      0.74        99
           1       0.54      0.62      0.58        55

    accuracy                           0.68       154
   macro avg       0.65      0.66      0.66       154
weighted avg       0.69      0.68      0.68       154
```


Confusion Matrix
Actual
0
1
70
29
21
34
70
60
50
40
30

References/Credits:

Dataset:

This project uses the Pima Indians Diabetes Dataset, originally provided by the National Institute of Diabetes and Digestive and Kidney Diseases. The dataset is publicly available on Kaggle.

Libraries & Tools:

- **Pandas and NumPy for data manipulation**
- **Matplotlib and Seaborn for visualization**
- **Scikit-learn for preprocessing, modeling, and evaluation**
- **Google Colab for cloud-based execution environment**