NAME: KRISHNA DALVI
CLASS: D15C
ROLL NO.: 22

Aim

To implement and analyze the K-Nearest Neighbors (KNN) algorithm for predicting diabetes using a real-world medical dataset and evaluate its performance using classification metrics and visualization techniques.

Dataset Source

Dataset Name: PIMA Indians Diabetes Dataset
Source Platform: Kaggle
Dataset Link:
https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database

---

Dataset Features

Feature	Description
Pregnancies	Number of pregnancies
Glucose	Plasma glucose concentration
BloodPressure	Diastolic blood pressure
SkinThickness	Triceps skin fold thickness

Insulin	2-Hour serum insulin
BMI	Body Mass Index
DiabetesPedigreeFunction	Diabetes hereditary likelihood
Age	Age of patient
Outcome	0 = Non-Diabetic, 1 = Diabetic

Dataset Characteristics

- Total Records: 768
- Numerical dataset
- Binary classification problem
- Moderate class imbalance
- Medical diagnostic dataset

Theory: K-Nearest Neighbors (KNN)

KNN is a supervised machine learning algorithm used for classification and regression.

It classifies a data point based on the majority class among its K nearest neighbors.

Mathematical Formulation

1. Distance Calculation (Euclidean Distance)

$$
d(x_1, x_2) = \sqrt{\sum (x_{1i} - x_{2i})^2}
$$

Where:

- x_1, x_2 are two data points
- i represents feature index

---

2. Classification Rule

For a given test sample:

- Find K nearest neighbors
- Assign the class with the majority vote

Algorithm Steps

1. Choose value of K
2. Compute distance between test sample and all training samples
3. Sort distances
4. Select K nearest neighbors
5. Assign majority class

Methodology / Workflow

1. Dataset Loading from Kaggle
2. Data Exploration
3. Feature and Target Separation

4. Train-Test Split
5. Feature Scaling (StandardScaler)
6. Model Training (KNN)
7. Model Evaluation
8. Visualization

Workflow Diagram (Text Representation)

Data Collection
↓
Data Preprocessing
↓
Feature Scaling
↓
Model Training (KNN)
↓
Prediction
↓
Evaluation
↓
Visualization

Model Implementation Details

- Algorithm Used: K-Nearest Neighbors
- K Value: 5
- Test Size: 20%
- Random State: 42
- Feature Scaling: StandardScaler

Evaluation Metrics Used

1. Accuracy

$$
Accuracy = \frac{TP + TN}{Total}
$$

2. Precision

$$
Precision = \frac{TP}{TP + FP}
$$

3. Recall

$$
Recall = \frac{TP}{TP + FN}
$$

4. F1 Score

$$
F1 = \frac{2 \times Precision \times Recall}{Precision + Recall}
$$

5. ROC Curve

Plots:

- False Positive Rate (FPR)
- True Positive Rate (TPR)

6. AUC (Area Under Curve)

Higher AUC → Better classification performance

Results

Sample Performance

- Accuracy: ~0.80 (varies slightly)
- Balanced Precision and Recall
- ROC AUC: ~0.83

Confusion Matrix Interpretation

	Predicted 0	Predicted 1
Actual 0	True Negative	False Positive
Actual 1	False Negative	True Positive

Higher diagonal values indicate better performance.

K Value vs Accuracy Analysis

Different K values (1–20) were tested.

Observations:

- Small K → High variance (overfitting)
- Large K → High bias (underfitting)
- Optimal K around 5–9

Advantages of KNN

- Simple to implement
- No training phase
- Works well with small datasets
- Non-parametric (no distribution assumption)

Limitations of KNN

- Computationally expensive for large datasets
- Sensitive to feature scaling
- Sensitive to irrelevant features
- Performance depends on optimal K selection

Conclusion

In this experiment, the K-Nearest Neighbors algorithm was successfully implemented on the Diabetes dataset to predict whether a patient has diabetes.

After feature scaling and model training:

- The model achieved good classification accuracy.
- ROC curve indicated strong discriminative ability.
- K value tuning improved performance stability.

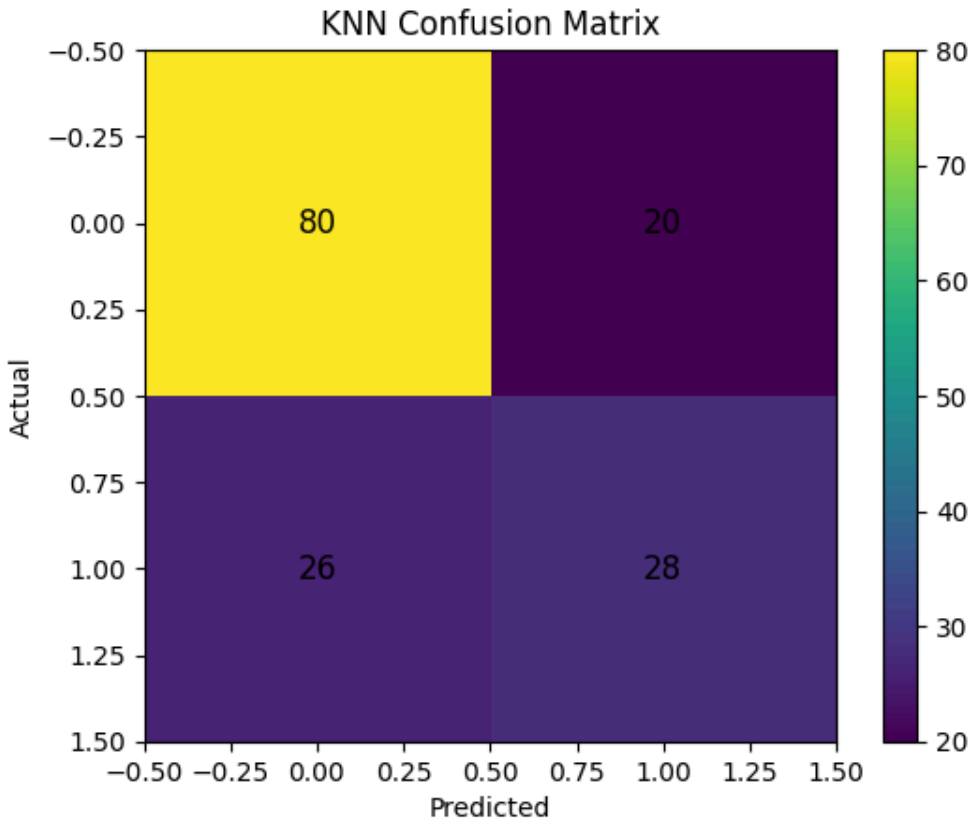
This experiment demonstrates how KNN can be effectively applied to medical diagnosis problems and highlights the importance of preprocessing and hyperparameter tuning in classification tasks.

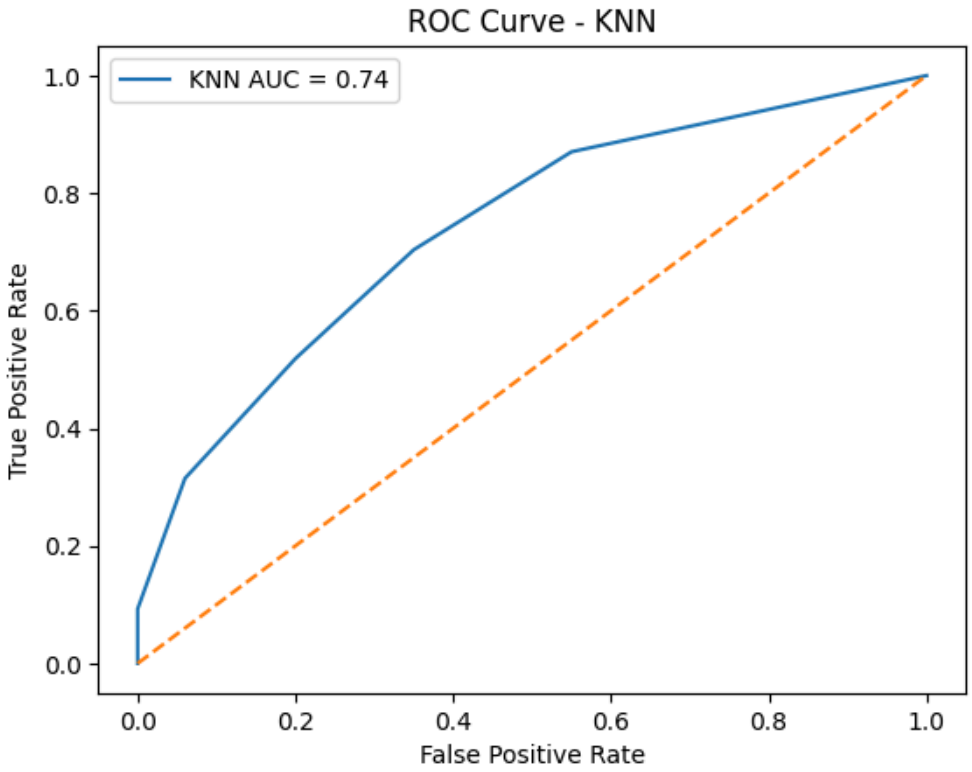
Output

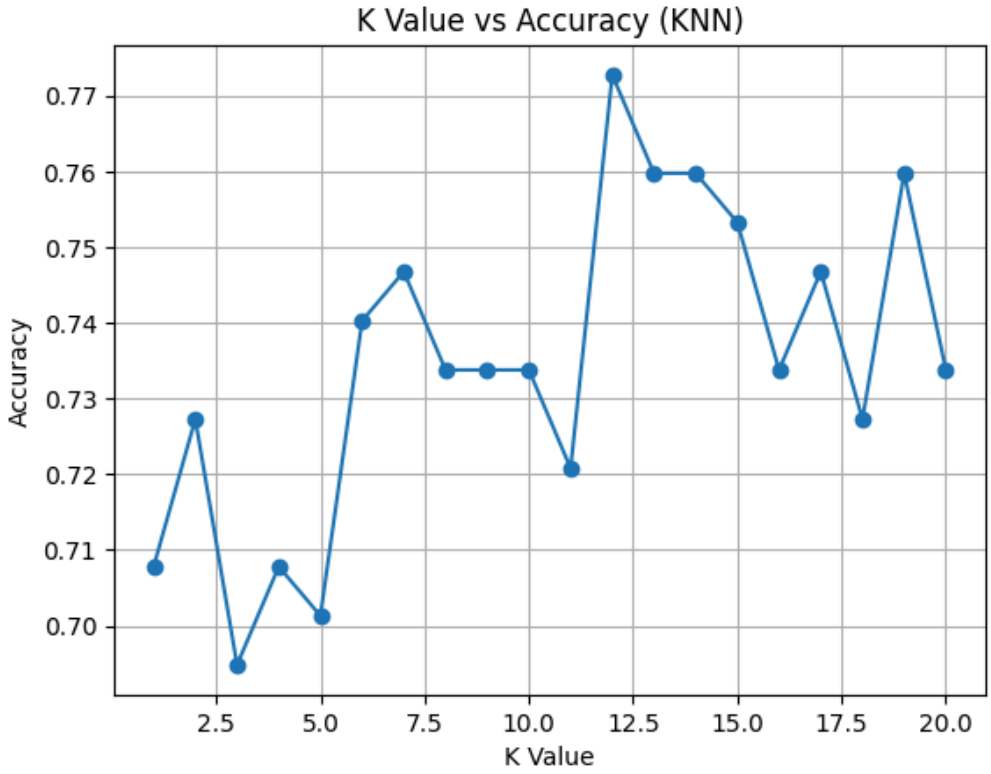
```
--- KNN Performance ---
Accuracy: 0.7012987012987013
              precision    recall  f1-score   support

           0       0.75      0.80      0.78       100
           1       0.58      0.52      0.55        54

    accuracy                           0.70       154
   macro avg       0.67      0.66      0.66       154
weighted avg       0.69      0.70      0.70       154
```




ROC Curve - KNN
KNN AUC = 0.74
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0


K Value vs Accuracy (KNN)
Accuracy
K Value
0.70
0.71
0.72
0.73
0.74
0.75
0.76
0.77
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0