**Assignment 6 : Performance Metrics on KNN**

**Made By : Arnav Bansal**

**Group : CSE G1  
SID : 23103046**

Q1: Implement KNN for classification on Sonar dataset (Binary-class classification dataset).

Use the dataset division as 70% for training and 30% for testing.

After prediction on 30% testing data.

Calculate Accuracy, Precision, Recall, F1-Score, FPR, FNR and AUC of the model.

Run it for 5 times and then calculate the above performance metrics for each time.

Finally calculate the average accuracy of the model.

Explanation:

Library Imports:

* The code imports key libraries like pandas, numpy for data manipulation, train\_test\_split from sklearn to split the dataset, StandardScaler to normalize the data, and several metrics like accuracy\_score, precision\_score, recall\_score, etc., to evaluate model performance.

Loading the Dataset:

* The dataset data.csv is loaded using pandas.read\_csv(). The features are stored in X, which excludes the last column, and the target variable (class labels) is stored in y (last column).

Mapping Class Labels:

* The target labels are strings ('Rock' and 'Mine'). These are mapped to binary values: 'Rock' becomes 0, and 'Mine' becomes 1. This is necessary for binary classification with the KNN algorithm.

Performance Metric Storage:

* An empty list result is created to store the evaluation metrics for each of the 5 runs of the KNN model. It will store metrics like accuracy, precision, recall, F1-score, FPR, FNR, and AUC.

5 Runs of Model Evaluation:

* The model is trained and evaluated 5 times by looping through different random splits of the dataset (random\_state=i), ensuring that the train-test split is slightly different in each iteration.

Train-Test Split (70%/30%):

* In each iteration, the dataset is split into 70% for training and 30% for testing using train\_test\_split().

Feature Scaling:

* Feature scaling is applied using StandardScaler. This step ensures that all features have a mean of 0 and a standard deviation of 1, which is important for KNN since it calculates distances between points.

KNN Model Training:

* A K-Nearest Neighbors (KNN) classifier with 5 neighbors (n\_neighbors=5) is instantiated and trained using the scaled training data (X\_train\_scaled).

Prediction and Metrics Calculation:

* After training, the model predicts labels on the test set (X\_test\_scaled). Various metrics are calculated:
  + Accuracy: Proportion of correct predictions.
  + Precision: Correct positive predictions out of total predicted positives.
  + Recall: Correct positive predictions out of actual positives.
  + F1-Score: Harmonic mean of precision and recall.
  + AUC: Area under the ROC curve, using predicted probabilities.
  + FPR and FNR: False positive rate and false negative rate are calculated from the confusion matrix.

Results Aggregation and Average Accuracy:

* The calculated metrics for each run are stored in the results list. After 5 runs, the results are converted into a DataFrame and displayed. The average accuracy over the 5 runs is computed and printed to give a summary of model performance across different splits.

