Experiment 4: Implementation of Multi-Class Classification using K-Nearest Neighbors (KNN)

1. Database Source

2. Dataset Description :

The experiment utilizes the **Iris Dataset** (Iris.csv).

- **Size:** The dataset contains **150 records** (50 for each species) and **6 columns**.
- **Features:**
  - Id: Unique identifier for each record.
  - SepalLengthCm: Length of the sepal in centimeters (Numerical).
  - SepalWidthCm: Width of the sepal in centimeters (Numerical).
  - PetalLengthCm: Length of the petal in centimeters (Numerical).
  - PetalWidthCm: Width of the petal in centimeters (Numerical).
- **Target Variable:** Species (Categorical: Iris-setosa, Iris-versicolor, Iris-virginica).
- **Characteristics:** This is a classic multi-class classification dataset. It is well-balanced and contains no missing values. The classes "Setosa" are linearly separable from the other two, while "Versicolor" and "Virginica" have some overlap in the feature space.

3. Mathematical Formulation of the Algorithm :

K-Nearest Neighbors is a non-parametric, lazy learning algorithm that classifies a data point based on how its neighbors are classified.

- **Distance Metric:** The most common metric used is the **Euclidean Distance** between two points p and q in n - dimensional space:

$$d(p, q) = \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2}$$

- **Voting Mechanism:** For a new point x , the algorithm identifies the K closest points in the training set. The predicted class $\hat{y}$ is the mode (most frequent class) among these K neighbors:

$$\hat{y} = \mathrm{argmax}_v \sum_{i \in N_k(x)} I(y_i = v)$$

Where $N_k(x)$ is the neighborhood of x and I(•) is an indicator function.

4. Algorithm Limitations :

- **Computationally Expensive:** Since it is a "lazy learner," it does not learn a discriminative function; instead, it performs a search through the entire training set for every prediction, making it slow for large datasets.
- **Curse of Dimensionality:** In high-dimensional spaces, the distance between points becomes less meaningful, causing performance to degrade.
- **Sensitivity to Noise:** Outliers can easily influence the classification if the value of K is too small.
- **Memory Intensive:** Requires storing the entire training dataset in memory to make predictions.

5. Methodology / Workflow :

1. **Data Loading:** Ingest Iris.csv using Pandas.
2. **Exploratory Data Analysis:** Visualize feature distributions using pair plots to observe class separation.
3. **Data Preprocessing:**
   - Drop the Id column as it carries no predictive value.
   - Feature Scaling: Since KNN relies on distance, **Standardization** (StandardScaler) is applied to ensure features with larger magnitudes (like Sepal Length) do not dominate the distance calculation.
4. **Data Splitting:** Partition the data into **Training (70%)** and **Testing (30%)** sets.
5. **Model Training:** Instantiate the KNeighborsClassifier and fit it using the training data.
6. **Evaluation:** Predict the species for the test set and generate a classification report and confusion matrix.

6. Performance Analysis :

- **Evaluation Metrics:**
  - **Accuracy:** Typically achieves **96% - 100%** on the Iris dataset.
  - **Confusion Matrix:** Most errors occur between Iris-versicolor and Iris-virginica due to their similar petal dimensions.
- **Interpretation:** The model is exceptionally robust for Iris-setosa. The high accuracy across all metrics indicates that the morphological features provided are highly discriminative for these species.

7. Hyperparameter Tuning :

The most critical hyperparameter in KNN is the **Number of Neighbors (K)**.

- **Process:** An **Elbow Method** or **Cross-Validation** was used to test K values ranging from 1 to 20.
- **Impact:**
  - **Low K (e.g., $K = 1$):** The model is sensitive to noise and outliers, leading to high variance (Overfitting).
  - **High K (e.g., $K = 20$):** The model becomes too smooth and may ignore local patterns, leading to high bias (Underfitting).
  - **Optimal K:** For this dataset, $K = 5$ or $K = 7$ usually provides the best trade-off, resulting in the highest cross-validation accuracy and a stable decision boundary.

Conclusion: The KNN algorithm successfully classified the Iris species with near-perfect accuracy. This experiment demonstrates the power of distance-based algorithms for well-defined classification tasks and underscores the necessity of feature scaling in such models.