**TOPIC NAME:IRIS FLOWERING CLASSIFICATION**

**NAME:Rahul Kumar Gupta**

**Branch:CSE(AI)**

**SECTION:C**

**ROLL NO.:202401100300191**

# Introduction to Iris Flower Classification

The Iris flower classification problem is one of the most classic and well-known examples in the field of machine learning and data science. It involves classifying iris flowers into different species based on their sepal and petal measurements. The dataset used for this problem is called the Iris dataset, which was introduced by the British statistician and biologist Ronald Fisher in 1936.

## Importance of the Iris Dataset:

- The Iris dataset is often used as a beginner-friendly dataset for learning classification algorithms because of its simplicity and small size.
- It is a **multiclass classification** problem, meaning there are more than two classes to predict.
- The dataset is well-balanced, with 50 samples for each of the three species.

## Applications:

- **Botany**: Classifying plant species based on physical characteristics.
- **Machine Learning Education**: Teaching fundamental concepts of supervised learning, classification, and evaluation metrics.
- **Data Science**: Demonstrating data preprocessing, visualization, and model training.

**Methodology for Iris Flower Classification**

The methodology for solving the **Iris flower classification** problem involves a systematic approach to analyze the dataset, preprocess the data, build a classification model, and evaluate its performance. Below is a step-by-step methodology:

**1. Problem Understanding**

- **Objective**: Classify iris flowers into one of three species (**Setosa**, **Versicolor**, or **Virginica**) based on their sepal and petal measurements.
- **Dataset**: The Iris dataset contains 150 samples, with 50 samples for each species. Each sample has four features: **Sepal Length**, **Sepal Width**, **Petal Length**, and **Petal Width**.

**2. Data Collection**

- Collect the dataset, either from a public repository (e.g., UCI Machine Learning Repository) or generate synthetic data if necessary.
- Ensure the dataset is clean and properly formatted.

**3. Data Preprocessing**

- **Data Cleaning**:
    - Check for missing values and handle them (e.g., imputation or removal).
    - Remove duplicates if any.
- **Feature Scaling**:
    - Normalize or standardize the features to ensure all features are on the same scale (e.g., using StandardScaler or MinMaxScaler).

# CODE TYPED

```
# Define the dataset
data = [
    [7.303274553, 2.475025253, 2.176048615, 0.695002979, 'Setosa'],
    [7.556927655, 2.987381009, 1.921584622, 1.172614805, 'Versicolor'],
    [5.254016374, 2.093516024, 3.672563876, 0.550423563, 'Virginica'],
    [6.409620272, 2.211041581, 1.812868616, 1.745372348, 'Versicolor'],
    [7.684009282, 4.056478761, 4.244270339, 0.772147812, 'Setosa'],
    [6.422517726, 4.032260738, 1.859477867, 1.559467631, 'Virginica'],
    [7.841052714, 4.047467666, 1.686771972, 0.455375349, 'Setosa'],
    [5.369546145, 4.001910728, 2.034979754, 1.514997694, 'Virginica'],
    [7.485555011, 2.1578975, 6.384229363, 2.42567448, 'Setosa'],
    [4.781220299, 3.71564794, 3.718518068, 0.967480314, 'Virginica'],
    [6.163901635, 2.208071681, 1.653575827, 1.818660006, 'Virginica'],
    [7.271711491, 2.103083754, 5.983380102, 0.499022342, 'Versicolor'],
    [4.668608383, 2.359761973, 5.406349138, 1.467767093, 'Versicolor'],
    [7.880377471, 2.032034463, 4.92393886, 1.254959596, 'Virginica'],
    [5.161007082, 2.10882377, 2.248808105, 0.10821174, 'Setosa'],
    [7.866695605, 3.148141876, 2.835834842, 0.989854885, 'Virginica'],
    [5.408765531, 3.95800881, 6.424216666, 2.340494975, 'Setosa'],
    [5.850162368, 2.474045174, 3.43553309, 1.55365103, 'Versicolor'],
    [7.82776353, 2.372291678, 5.968919023, 1.446512336, 'Versicolor'],
    [6.650334851, 2.665678693, 3.866596082, 1.834436922, 'Setosa']
]
```

```
# Define a function to classify the species based on rules
def classify_iris(sepal_length, sepal_width, petal_length, petal_width):
    if petal_width < 0.6:
        return 'Setosa'
    elif petal_length >= 5.0 and petal_width < 1.5:
        return 'Versicolor'
    elif petal_length >= 3.0 and petal_width >= 1.5:
        return 'Virginica'
    else:
        return 'Unknown'

# Test the function on the dataset
correct = 0
total = len(data)

for row in data:
    sepal_length, sepal_width, petal_length, petal_width, true_species = row
    predicted_species = classify_iris(sepal_length, sepal_width, petal_length, petal_width)
    if predicted_species == true_species:
        correct += 1
    print(f"True: {true_species}, Predicted: {predicted_species}")

# Calculate accuracy
accuracy = (correct / total) * 100
print(f"\nAccuracy: {accuracy:.2f}%")
```

## SCREENSHOTS OF OUTPUT: