Judging Flowers using Classification Models

Jayani Tripathi

Table of Contents:

Overview

The project focuses on applying various classification algorithms to the Iris Dataset. The dataset includes attributes of three species of flowers — *Iris Setosa*, *Iris Versicolor*, and *Iris Virginica*. By leveraging this dataset, the project aims to classify the flower species based on their attributes using multiple machine learning models.

Objectives

1. Deepen familiarity with the SciKit Learn library and its documentation.
2. Gain hands-on experience with classification models such as Logistic Regression, Support Vector Machines (SVM), Neural Networks, and k-Nearest Neighbors (k-NN).
3. Develop insights into the strengths and weaknesses of different classifiers through performance analysis.

Steps and Implementation

1. **Loading the Dataset** The Iris dataset was loaded into the environment using the SciKit Learn library's datasets module. Basic exploratory data analysis (EDA) was conducted to understand the structure and contents of the data.

2. **Data Splitting** The dataset was split into training and testing sets using an 80-20 ratio. The train_test_split method from SciKit Learn ensured a randomized and stratified division of data to maintain class distribution.

3. **Model Training and Evaluation**

    - **Logistic Regression** This model was used to predict probabilities for the flower classes. Its simplicity and interpretability make it a suitable baseline classifier.

    - **Support Vector Machines (SVM)** Both linear and radial basis function (RBF) kernels were implemented. SVM is particularly effective for smaller datasets and works well in high-dimensional spaces.

    - **Neural Network** A multi-layer perceptron (MLP) was trained to classify the flowers. Parameters such as the number of layers, activation functions, and learning rates were tuned for optimal performance.

    - **k-Nearest Neighbors (k-NN)** The model was evaluated with different values of *k* to find the most appropriate number of neighbors for accurate predictions.

4. **Performance Metrics** Each model's performance was measured using metrics like accuracy, precision, recall, and F1-score. Confusion matrices and classification reports were generated to analyze prediction errors and class-level performance.

Conclusion This project provided valuable insights into the application of classification models to a real-world dataset. The Iris Dataset's simplicity allowed for focused exploration of algorithmic performance. SVM emerged as the most effective model, demonstrating the importance of kernel selection. Future work could explore ensemble methods or deep learning models for further improvements.

Note: Please feel free to email me for the code and results of this project!