Iris Flower Classification Project

Author: Avni Jain **Date:** 21st March 2025

1. Introduction

1.1 Objective:

The primary objective of this project is to develop a machine learning model that can accurately classify Iris flowers into three species (Iris Setosa, Iris Versicolor, and Iris Virginica) based on their sepal and petal measurements. This project demonstrates the application of various classification algorithms and emphasizes the performance of Neural Networks for this specific task.

1.2 Dataset:

The project utilizes the Iris Flower Dataset, a widely used benchmark dataset in machine learning. This dataset contains 150 instances, with 50 instances for each of the three Iris species. Each instance includes four features: sepal length, sepal width, petal length, and petal width.

1.3 Project Structure:

The project follows these key stages:

1. **Data Loading and Preprocessing:** Importing the dataset, handling missing values, and encoding categorical features.
2. **Exploratory Data Analysis (EDA):** Visualizing data patterns and relationships between features.
3. **Model Selection and Training:** Evaluating different machine learning models and selecting the best-performing one, in this case, the Neural Network, due to its accuracy in this specific scenario.
4. **Model Evaluation:** Assessing the chosen model's performance using various metrics.
5. **Prediction:** Demonstrating the model's ability to classify new, unseen data.

2. Data Preprocessing

2.1 Data Cleaning:

- The dataset was checked for missing values and duplicates. Duplicates were removed to ensure data integrity. No missing values were found.

2.2 Feature Encoding:

- The target variable 'species' was encoded using Label Encoding to transform the categorical species names ('Iris-setosa', 'Iris-versicolor', 'Iris-virginica') into numerical labels (0, 1, 2).

3. Exploratory Data Analysis (EDA)

- Scatter plots were used to visualize the relationships between different features (sepal length vs. sepal width, petal length vs. petal width, etc.) for each species, revealing distinct clusters.
- A correlation heatmap was generated to identify potential correlations between the features.

4. Model Selection and Training

4.1 Models Considered:

Several classification models were evaluated:

- Logistic Regression
- Decision Tree
- Random Forest
- Neural Network

4.2 Model Justification (Neural Network):

The Neural Network model was chosen as the primary model for this project due to its superior performance in terms of accuracy. It was observed to provide the most accurate predictions during evaluation.

4.3 Hyperparameter Tuning:

Hyperparameter optimization was performed to enhance the model's performance. RandomizedSearchCV was used to explore a range of hyperparameter values, and the optimal set of hyperparameters was selected based on cross-validation results.

Optimal Hyperparameters:

- `hidden_layer_sizes`: [Specific value found by RandomizedSearchCV]
- `alpha`: [Specific value found by RandomizedSearchCV]

5. Model Evaluation

5.1 Evaluation Metrics:

The following metrics were used to evaluate the models:

- Precision
- Recall
- Accuracy
- F1-Score

5.2 Model Performance:

[Insert table or text summarizing the evaluation scores for the Neural Network model and other models considered]

Observation:

The Neural Network model outperformed other models in terms of overall accuracy and other key metrics, solidifying its selection as the best model for this task.

6. Prediction

The trained Neural Network model was utilized to classify a new data point:

```
x_rf = np.array([[5.1, 3.5, 1.4, 0.2]])
```

The predicted category label for this data point was found to be: 'Iris-Setosa'

7. Conclusion

This project successfully developed a machine learning model for Iris flower classification. The Neural Network model, after hyperparameter optimization, demonstrated superior performance compared to other classification algorithms, indicating its suitability for this task.

8. Further Improvements

- Explore other advanced neural network architectures, such as convolutional neural networks (CNNs) or recurrent neural networks (RNNs), to potentially further enhance performance.
- Experiment with additional hyperparameter tuning techniques, like Bayesian optimization, to refine the model.
- Consider incorporating feature engineering to potentially create more informative features.