IRIS Flower Classification using Logistic Regression and Random Forest

Devansh

ITEP-2023

IIT BHUBANESHWAR

Period of Internship: 21ST January 2026 – 17TH February 2026

Report submitted to: IDEAS – Institute of Data Engineering, Analytics and Science Foundation, ISI Kolkata

1. Abstract

This project focuses on implementing supervised machine learning algorithms to solve a multi-class classification problem using the classic Iris dataset. The dataset consists of 150 samples of Iris flowers, categorized into three species based on four morphological features: sepal length, sepal width, petal length, and petal width. I utilized Python libraries such as Pandas for data manipulation and Scikit-learn for model building. To compare different approaches, I selected two distinct algorithms: Logistic Regression (a linear model) and Random Forest Classifier (an ensemble method). The data was pre-processed and split into training and testing sets to ensure accurate evaluation. Both models demonstrated exceptional performance, achieving 100% accuracy on the test set. This report details the methodology, code implementation, and analysis of these findings.

2. Introduction

Relevance and Background: Classification is a fundamental task in machine learning with applications ranging from biology to finance. This project uses the Iris dataset, introduced by Ronald Fisher in 1936, to demonstrate the practical application of supervised learning. The project aims to understand how physical features (dimensions of sepals and petals) can be mathematically modeled to

predict categorical outcomes (species).

Technology and Procedure: The project was executed using Python within the Google Colab environment. Key technologies included:

- **Pandas:** Used for creating DataFrames and handling data manipulation.
- **Scikit-Learn (sklearn):** Used for accessing the dataset, performing an 80/20 train-test split, model training, and metric evaluation.

The procedure involved loading the data, performing exploratory data analysis to understand class distributions, splitting the data, training the two classifiers, and finally evaluating them using quantitative metrics.

Training Topics Covered (First Two Weeks): During the initial phase of the internship, I received training on:

- Python Programming Fundamentals (Variables, Loops, Functions).
- Data Manipulation using Pandas (DataFrames, Indexing, Handling Missing Data).
- Numerical Computing with NumPy.
- Data Visualization principles.
- Introduction to Machine Learning concepts (Supervised vs. Unsupervised Learning).
- Scikit-learn workflow (Fit, Predict, Score).

3. Project Objective

The primary objectives of this project were:

- To successfully load, explore, and pre-process the Iris dataset for machine learning tasks.
- To implement and train a **Logistic Regression** model to understand linear classification boundaries.
- To implement and train a **Random Forest Classifier** to understand ensemble learning.
- To evaluate and compare the performance of both models using metrics such as Accuracy Score, Precision, Recall, and F1-Score.
- To demonstrate how morphological features effectively differentiate flower species.

4. Methodology

Data Collection and Tools: The data was obtained directly from the `sklearn.datasets` library (`load_iris`), and analysis was performed using Python.

Process Flow:

1. **Data Loading:** Imported the raw data and extracted feature names and target labels.
2. **Data Pre-processing:**
    - Converted raw arrays into a structured Pandas DataFrame.
    - Added the target column to align features with labels.
    - Verified data quality using `value_counts()` (confirmed balanced classes) and `DESCR` (confirmed no missing values).
3. **Data Splitting:**
    - The dataset was split into Feature Matrix (X) and Target Vector (y).
    - **Sampling Methodology:** A random split was performed using `train_test_split`.
    - **Split Ratio:** 80% for Training and 20% for Testing.
    - **Reproducibility:** A `random_state` of 42 was used to ensure consistent results.

4. **Model Selection and Validation:**
    - **Logistic Regression:** Selected as a baseline to test for linear separability (`max_iter=200`).
    - **Random Forest:** Selected to test a non-linear, ensemble approach (`n_estimators=100`).
5. **Evaluation:** Models were validated using the held-out test set.

Code Availability: The Python code developed for this analysis is available on GitHub:

- https://github.com/Devanshblip/IDEAS_Internship_Project_2026

5. Data Analysis and Results

Descriptive Analysis: The dataset consists of 150 instances with 4 numeric predictive attributes. The class distribution is perfectly balanced (50 samples per species).

Feature	Count	Description
Sepal Length	150	Numeric (cm)
Sepal Width	150	Numeric (cm)
Petal Length	150	Numeric (cm)
Petal Width	150	Numeric (cm)

Export to Sheets

Inferential Analysis & Model Performance:

Model 1: Logistic Regression

- **Accuracy:** 1.0 (100%)
- The model correctly predicted the species for all 30 flowers in the test set.

Model 2: Random Forest Classifier

- **Accuracy:** 1.0 (100%)
- **Classification Report:**

Class	Precision	Recall	F1-Score	Support

0 (Setosa)	1.00	1.00	1.00	10
1 (Versicolour)	1.00	1.00	1.00	9
2 (Virginica)	1.00	1.00	1.00	11
Overall	**1.00**	**1.00**	**1.00**	**30**

Export to Sheets

Comparative Analysis: Both models achieved identical, perfect scores. This indicates that the test data provided was distinct enough for both linear (Logistic Regression) and non-linear (Random Forest) boundaries to separate the classes without error.

6. Conclusion

After conducting the analysis on the Iris dataset, I drew the following conclusions:

1. **High Separability:** The dataset features (specifically petal dimensions) are highly effective discriminators. The fact that a linear model achieved 100% accuracy suggests the classes are linearly separable in this feature space.
2. **Model Robustness:** The Random Forest classifier also achieved 100% accuracy, confirming that ensemble methods handle this multi-class problem effectively, though they may be computationally heavier than necessary for this specific simple dataset.
3. **Data Quality:** The dataset required no imputation or cleaning, which facilitated efficient model training.

Recommendation for Future Work: While "100% accuracy" is an ideal result, it can sometimes indicate a dataset that is too simple for complex models. Future work should involve:

- Applying **K-Fold Cross-Validation** to ensure the perfect score holds across different data splits.
- Testing these models on a larger, noisier, or real-world dataset where classes overlap more significantly.

7. APPENDICES

Appendix 1: References

1. Course material provided by the institute.
2. Fisher, R.A. "The use of multiple measurements in taxonomic problems", Annual Eugenics, 7, Part II, 179-188 (1936).
3. Scikit-learn Documentation: https://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_iris.html
4. Pandas Documentation: https://pandas.pydata.org/docs/

Appendix 2: GitHub Link

- https://github.com/Devanshblip/IDEAS_Internship_Project_2026