

# Task 5: Analysis of Iris Dataset

**Suchismita Maity**

November 21, 2025

## 1 Introduction

The objective of this task is to perform an analysis on the **Iris dataset**. The Iris dataset is a classic dataset in machine learning, consisting of 150 samples from three species of Iris (Iris setosa, Iris virginica, and Iris versicolor). Four features were measured from each sample: the length and the width of the sepals and petals, in centimeters.

This report outlines the Exploratory Data Analysis (EDA) performed and the implementation of a Decision Tree Classifier to predict the species of the flower based on its physical measurements.

## 2 Methodology

The analysis was conducted using Python, leveraging the following libraries:

- **Pandas:** For data manipulation and loading the CSV file.
- **Seaborn & Matplotlib:** For data visualization (Pairplots and Heatmaps).
- **Scikit-Learn:** For splitting the data, training the Decision Tree model, and evaluating its performance.

## 3 Exploratory Data Analysis (EDA)

EDA was performed to understand the distribution of the data and relationships between features.

### 3.1 Data Structure

The dataset contains 150 rows and 6 columns (including ID). There are no missing values in the dataset. The target variable is 'Species'.

### 3.2 Visualizations

We generated a pairplot to visualize the pairwise relationships. The plot revealed that *Iris setosa* is linearly separable from the other two species, while *Iris versicolor* and *Iris virginica* have some overlap but are largely distinct based on Petal Length and Petal Width.

## 4 Model Implementation

A Decision Tree Classifier was chosen for this task due to its interpretability and effectiveness on this specific dataset.

### 4.1 Python Code

Below is the core implementation used to split the data and train the model:

```
1 # Splitting the data
2 X = df.drop('Species', axis=1)
3 y = df['Species']
4 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
5
6 # Training the model
7 dtree = DecisionTreeClassifier()
8 dtree.fit(X_train, y_train)
9
10 # Prediction
11 y_pred = dtree.predict(X_test)
```

Listing 1: Decision Tree Implementation

## 5 Results

The model was evaluated on a test set (20% of the data).

- **Accuracy:** The Decision Tree model achieved an accuracy of approximately **96% - 100%** (depending on the random split).
- **Evaluation Metrics:** The precision, recall, and F1-score were consistently high across all three classes.

## 6 Conclusion

In this task, we successfully analyzed the Iris dataset. The EDA showed clear distinctions between species based on petal dimensions. The Decision Tree classifier proved to be a highly effective model for this classification problem, achieving near-perfect accuracy.