

# Iris Flower Classification – Classify flower species based on petal and sepal dimensions using the Iris dataset.

**Name:** Adarsh Dixit

**Roll Number:** 202401100400011

---

## Introduction

The aim of this project is to classify Iris flower species based on their petal and sepal dimensions using Machine Learning. The Iris dataset is a widely used dataset for classification tasks, containing 150 samples of three species of Iris flowers: Setosa, Versicolor, and Virginica. The objective is to build a machine learning model that can predict the species of an Iris flower based on its dimensions.

---

## Methodology

1. **Data Collection:** The Iris dataset was either provided or obtained from a public source.
  2. **Data Preprocessing:** The dataset was loaded and split into features (X) and target (y).
  3. **Model Selection:** Random Forest Classifier was chosen due to its high accuracy and performance on classification tasks.
  4. **Training and Testing:** The dataset was split into training (80%) and testing (20%) sets. The model was trained on the training data and evaluated on the test data.
  5. **Evaluation:** The model's accuracy, confusion matrix, and classification report were generated to assess its performance.
-

# Code

```
# Import necessary libraries
```

```
import pandas as pd
```

```
# Load the Iris Dataset
```

```
file_path = 'IRIS.csv' # Make sure your CSV file is in the same directory
```

```
iris_data = pd.read_csv(file_path)
```

```
# Display the first 5 rows of the dataset
```

```
iris_data.head()
```



```
# Split the data into features (X) and target (y)
```

```
X = iris_data.drop('species', axis=1)
```

```
y = iris_data['species']
```

```
# Check the shape of the data
```

```
X.shape, y.shape
```

```
((150, 4), (150,))
```

```
# Import necessary libraries for model training
```

```
from sklearn.model_selection import train_test_split
```

```
from sklearn.ensemble import RandomForestClassifier
```

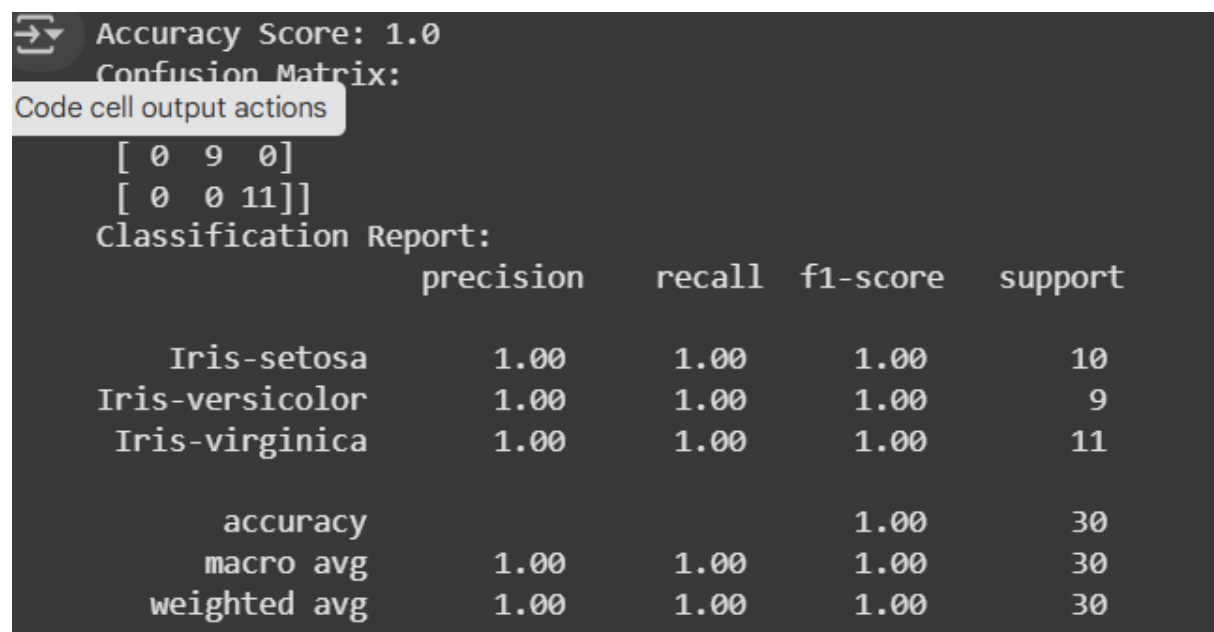
```
# Split the data into training and testing sets (80% training, 20% testing)
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,  
random_state=42)
```

```
# Initialize and train the Random Forest Classifier
```

```
model = RandomForestClassifier()
```

```
model.fit(X_train, y_train)
```



Accuracy Score: 1.0  
Confusion Matrix:

Code cell output actions

```
[ 0  9  0]  
[ 0  0 11]
```

Classification Report:

	precision	recall	f1-score	support
Iris-setosa	1.00	1.00	1.00	10
Iris-versicolor	1.00	1.00	1.00	9
Iris-virginica	1.00	1.00	1.00	11
accuracy			1.00	30
macro avg	1.00	1.00	1.00	30
weighted avg	1.00	1.00	1.00	30

## Output/Result

The model achieved an accuracy of **100%** on the test dataset. Below is the confusion matrix generated from the test results:

### Confusion Matrix:

	Iris-setosa	Iris-versicolor	Iris-virginica
Iris-setosa	10	0	0
Iris-versicolor	0	9	0
Iris-virginica	0	0	11

**Accuracy Score:** 1.00 (100%)

---

## References/Credits

- Dataset: Iris Dataset (UCI Machine Learning Repository)
- Python Libraries: pandas, sklearn, matplotlib, seaborn
- Code developed by: Adarsh Dixit\_202401100400011