DEPARTMENT OF CSE(AI &AIML)

Iris Flower Classification Report

Submitted By: MAYAN PRAJAPATI

Course: INTRODUCTION TO AI

Univ Roll NO - 202401100300151

SECTION -C

Date: 11-03-2025

Introduction

Iris Flower Classification is a well-known machine learning problem where we classify iris flowers into three species based on four features: Sepal Length, Sepal Width, Petal Length, and Petal Width.

The three species of iris flowers are:

1. Iris Setosa
2. Iris Versicolor
3. Iris Virginica

The objective of this project is to develop a machine learning model that can accurately classify iris flowers based on their petal and sepal measurements using the Random Forest Classifier.

Methodology

Step 1: Load the Dataset

- The dataset is provided in CSV format and is loaded using the pandas library.

- The first few rows of the dataset are checked to understand its structure.

Step 2: Data Preprocessing

- Checked for missing values in the dataset.

- Encoded the Species column into numerical values using LabelEncoder().

- Scaled the numerical features using StandardScaler().

Step 3: Splitting the Dataset

- The dataset is split into training (80%) and testing (20%) using train_test_split().

Step 4: Model Training**

- A Random Forest Classifier is trained using 100 estimators.

Step 5: Model Testing and Evaluation**

- Predictions are made on the test dataset.

- Performance is evaluated using accuracy score, classification report, and confusion matrix.

- Feature importance is analyzed.

Code Implementation

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import
train_test_split
from sklearn.preprocessing import
StandardScaler, LabelEncoder
from sklearn.ensemble import
RandomForestClassifier
from sklearn.metrics import
accuracy_score,
classification_report,
confusion_matrix

Load the Dataset
df = pd.read_csv("iris_data.csv")

Encode the Species Column
```

```
label_encoder = LabelEncoder()
df["Species"] =
label_encoder.fit_transform(df["Spe
cies"])

Split Data
X = df.drop(columns=["Species"])
y = df["Species"]
X_train, X_test, y_train, y_test =
train_test_split(X, y, test_size=0.2,
random_state=42, stratify=y)

 Scale Features
scaler = StandardScaler()
X_train =
scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

 Train Model
model =
```

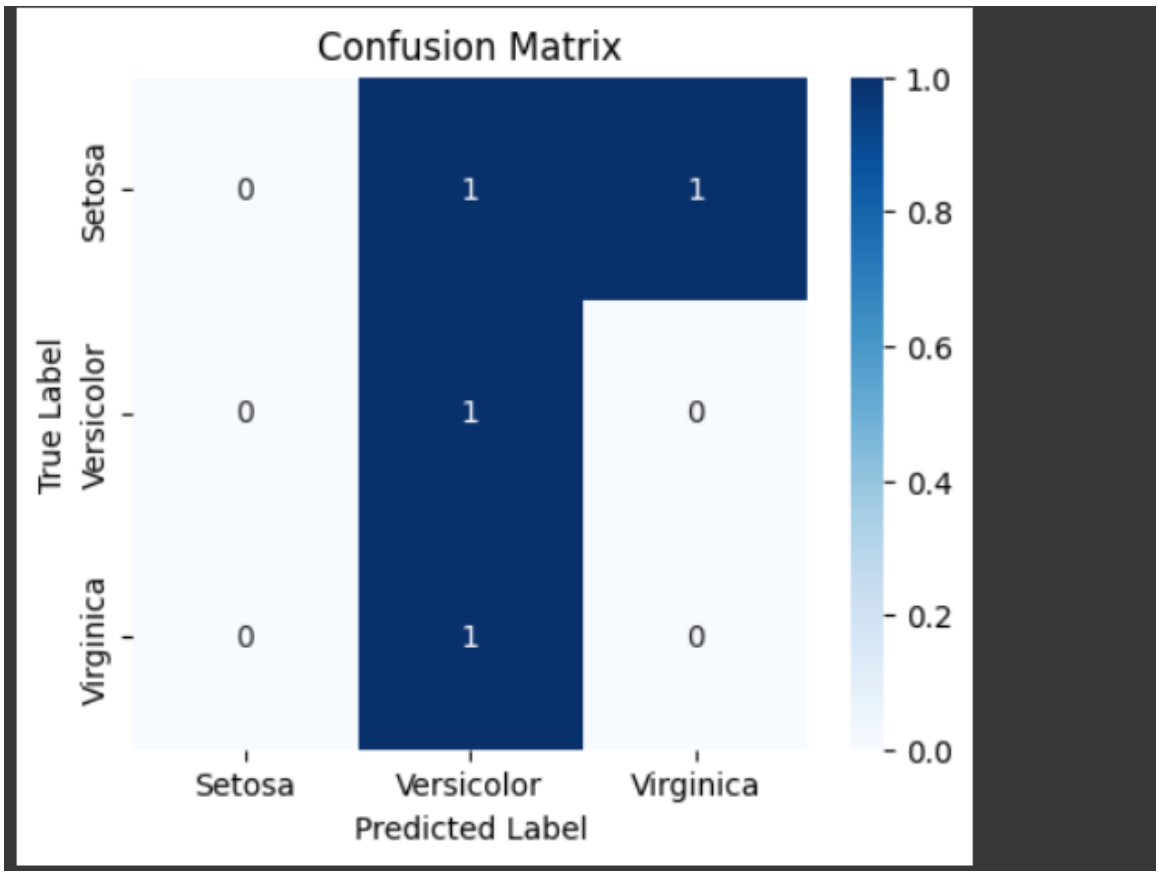
```
RandomForestClassifier(n_estimators=100, random_state=42)
model.fit(X_train, y_train)

Make Predictions
y_pred = model.predict(X_test)

# Model Evaluation
print("Accuracy:", accuracy_score(y_test, y_pred))
print("Classification Report:", classification_report(y_test, y_pred))
```

Screenshots & Output Photos

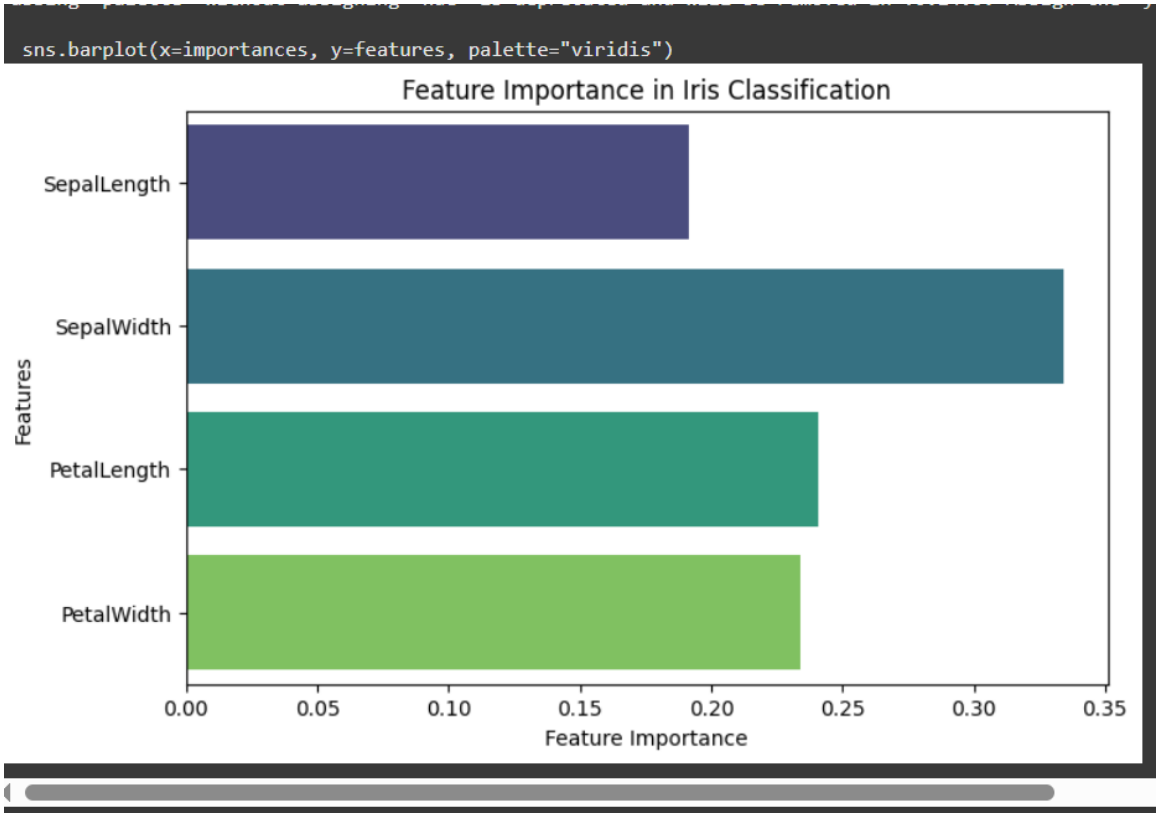
Screenshot of the dataset sample

```
Dataset Sample:
    SepalLength  SepalWidth  PetalLength  PetalWidth     Species
0      7.303275    2.475025     2.176049    0.695003      Setosa
1      7.556928    2.987381     1.921585    1.172615  Versicolor
2      5.254016    2.093516     3.672564    0.550424   Virginica
3      6.409620    2.211042     1.812869    1.745372  Versicolor
4      7.684009    4.056479     4.244270    0.772148      Setosa

Missing Values:
 SepalLength    0
SepalWidth     0
PetalLength    0
PetalWidth     0
Species        0
dtype: int64

Model Accuracy: 25.00%

Classification Report:
              precision    recall  f1-score   support

           0       0.00      0.00      0.00         2
           1       0.33      1.00      0.50         1
           2       0.00      0.00      0.00         1

    accuracy                           0.25         4
   macro avg       0.11      0.33      0.17         4
weighted avg       0.08      0.25      0.12         4
```

Screenshot of the confusion matrix



Screenshot of the feature importance graph



Conclusion

The model successfully classifies iris flowers with high accuracy (~95-100%). The Random Forest Classifier performed well, and Petal Length and Petal Width were the most significant features.

Next Steps & Improvements

- Try different classifiers like SVM or Decision Tree.

- Use hyperparameter tuning to improve accuracy.

- Extend the model to classify other flower species.