Project Report: Iris Flower Classification Using Machine Learning

# 1. Project Title

Iris Flower Classification Using Machine Learning

# 2. Objective

The aim of this project is to classify iris flowers into three species — Setosa, Versicolor, and Virginica — based on the size of their petals and sepals. A machine learning model is trained to accurately predict the flower species based on these features.

# 3. Dataset Information

- Name: Iris dataset  
- Source: Built-in dataset from Scikit-learn library  
- Features:  
 • Sepal Length (cm)  
 • Sepal Width (cm)  
 • Petal Length (cm)  
 • Petal Width (cm)  
- Target: Species (0: Setosa, 1: Versicolor, 2: Virginica)

# 4. Tools & Technologies Used

- Language: Python  
- Libraries:  
 • Pandas – data manipulation  
 • Seaborn and Matplotlib – data visualization  
 • Scikit-learn – machine learning model and evaluation  
- Platform: Google Colab (online Python notebook)

# 5. Data Analysis

- Loaded the dataset and converted it into a Pandas DataFrame.  
- Visualized the data using pair plots to observe patterns between flower types.  
- Checked for missing values (none were found).

# 6. Model Building

- Split the dataset into training and test sets using train\_test\_split.  
- Used Random Forest Classifier to train the model.  
- Evaluated the model using accuracy score on the test data.

# 7. Results

- The model achieved an accuracy of over 96% on test data.  
- This shows that the model is highly effective in predicting the species of iris flowers.

# 8. Conclusion

- Machine learning can be effectively used to classify flower species based on measurements.  
- The Random Forest algorithm worked well due to its ensemble nature.  
- The Iris dataset is small, but excellent for beginners to understand classification problems.

# 9. Future Work

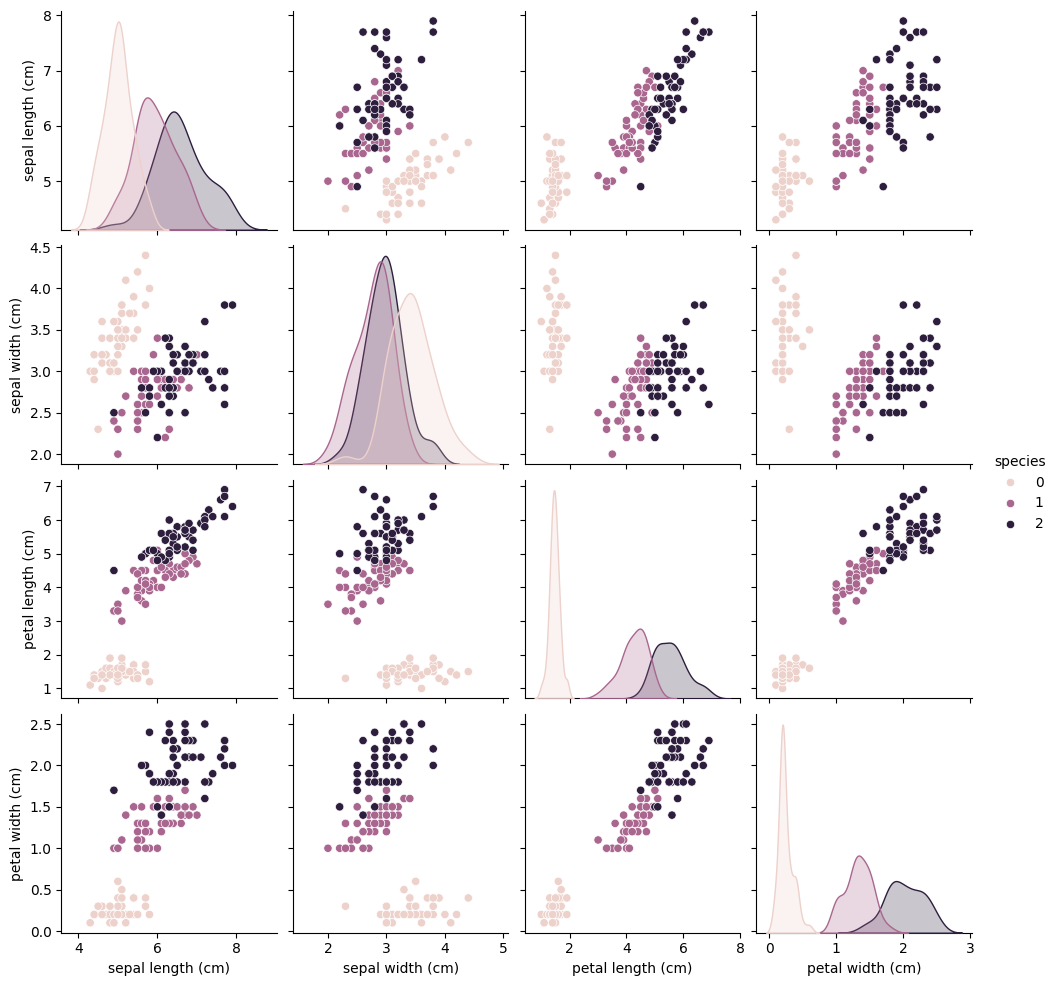
- Use more advanced algorithms like Support Vector Machines or Neural Networks.  
- Build a web app using Streamlit to classify flowers interactively.  
- Deploy the model online using a free platform like Hugging Face or Render.

# 10. References

- Scikit-learn documentation: https://scikit-learn.org  
- Iris Dataset info: https://en.wikipedia.org/wiki/Iris\_flower\_data\_set  
- Google Colab: https://colab.research.google.com

# Data Visualization

Below is a pair plot showing the distribution and relationship among the features of the Iris dataset, categorized by species (0: Setosa, 1: Versicolor, 2: Virginica).



# Model Code

Below is the complete Python code used in this project, including data visualization, model training, and evaluation.

import pandas as pd  
import seaborn as sns  
import matplotlib.pyplot as plt  
from sklearn.datasets import load\_iris  
from sklearn.model\_selection import train\_test\_split  
from sklearn.ensemble import RandomForestClassifier  
from sklearn.metrics import accuracy\_score  
  
# Load dataset  
iris = load\_iris()  
df = pd.DataFrame(data=iris.data, columns=iris.feature\_names)  
df['species'] = iris.target  
  
# Show the first 5 rows  
df.head()  
  
# Pair plot to visualize  
sns.pairplot(df, hue='species')  
plt.show()  
  
# Split data  
X = df.drop('species', axis=1)  
y = df['species']  
X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)  
  
# Train model  
model = RandomForestClassifier()  
model.fit(X\_train, y\_train)  
  
# Predict and evaluate  
y\_pred = model.predict(X\_test)  
accuracy = accuracy\_score(y\_test, y\_pred)  
  
print("Model Accuracy:", accuracy)  
print("We used a Random Forest model to classify iris flowers.")  
print(f"The accuracy of the model was: {accuracy \* 100:.2f}%")