

Start coding or [generate](#) with AI.

```
# ✅ Step 1: Upload IRIS.csv (Only for Google Colab)
from google.colab import files
uploaded = files.upload()

# ✅ Step 2: Import Libraries
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix

# ✅ Step 3: Load Dataset
df = pd.read_csv("IRIS.csv")
print("📄 First 5 Rows:")
print(df.head())

# ✅ Step 4: Dataset Info & Cleaning
print("\n🔍 Dataset Info:")
print(df.info())
print("\n❓ Missing Values:\n", df.isnull().sum())
print("\n📊 Class Distribution:\n", df['species'].value_counts())

# ✅ Step 5: Visualize Dataset
sns.pairplot(df, hue='species')
plt.title("🌸 Pairplot of Iris Features")
plt.show()

# ✅ Step 6: Prepare Features & Labels
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)

# ✅ Step 7: Feature Scaling
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

# ✅ Step 8: Train Model
model = LogisticRegression()
model.fit(X_train_scaled, y_train)

# ✅ Step 9: Predict & Evaluate
y_pred = model.predict(X_test_scaled)
print("\n✅ Accuracy:", accuracy_score(y_test, y_pred))
print("\n📋 Classification Report:\n", classification_report(y_test, y_pred))

# ✅ Step 10: Confusion Matrix
cm = confusion_matrix(y_test, y_pred)
sns.heatmap(cm, annot=True, cmap='Blues', fmt='g',
            xticklabels=y.unique(), yticklabels=y.unique())
plt.title("📊 Confusion Matrix")
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.show()
```



Choose Files IRIS.csv

• IRIS.csv(text/csv) - 4617 bytes, last modified: 8/4/2025 - 100% done

Saving IRIS.csv to IRIS.csv

First 5 Rows:

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa

Dataset Info:

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 150 entries, 0 to 149

Data columns (total 5 columns):

#	Column	Non-Null Count	Dtype
0	sepal_length	150 non-null	float64
1	sepal_width	150 non-null	float64
2	petal_length	150 non-null	float64
3	petal_width	150 non-null	float64
4	species	150 non-null	object

dtypes: float64(4), object(1)

memory usage: 6.0+ KB

None

Missing Values:

sepal_length 0

sepal_width 0

petal_length 0

petal_width 0

species 0

dtype: int64

Class Distribution:

species

Iris-setosa 50

Iris-versicolor 50

Iris-virginica 50

Name: count, dtype: int64

/usr/local/lib/python3.11/dist-packages/IPython/core/pylabtools.py:151: UserWarning: Glyph 127800 (\N{CHERRY BLOSSOM}) missing from font
fig.canvas.print_figure(bytes_io, **kw)

