In [1]:

```
import pandas as pd
import numpy as np
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis as L
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import confusion_matrix, classification_report, accu



iris = load_iris()
df = pd.DataFrame(data=iris.data, columns=iris.feature_names)
df['target'] = iris.target
```

In [3]:

```
scaler = StandardScaler()
X_scaled = scaler.fit_transform(df.iloc[:, :-1])
```

In [9]:

```
X_train, X_test, y_train, y_test = train_test_split(X_scaled, df['target'
```

In [11]:

```
lda = LDA(n_components=2)
X_train_lda = lda.fit_transform(X_train, y_train)
X_test_lda = lda.transform(X_test)
```

In [13]:

```
classifier = LogisticRegression()
classifier.fit(X_train_lda, y_train)
```

Out[13]:

```
▾ LogisticRegression
LogisticRegression()
```

In [15]:

```
y_pred = classifier.predict(X_test_lda)
```

In [17]:

```
conf_matrix = confusion_matrix(y_test, y_pred)
class_report = classification_report(y_test, y_pred)
accuracy = accuracy_score(y_test, y_pred)
```

In [19]:

```
print("Confusion Matrix:\n", conf_matrix)
print("\nClassification Report:\n", class_report)
print("\nAccuracy Score:", accuracy)
```

```
Confusion Matrix:
 [[15  0  0]
 [ 0 15  0]
 [ 0  1 14]]

Classification Report:
               precision    recall  f1-score   support

           0       1.00      1.00      1.00        15
           1       0.94      1.00      0.97        15
           2       1.00      0.93      0.97        15

    accuracy                           0.98        45
   macro avg       0.98      0.98      0.98        45
weighted avg       0.98      0.98      0.98        45


Accuracy Score: 0.9777777777777777
```

In [21]:

```
sample = np.array([[5.1, 3.5, 1.4, 0.2]])
sample_scaled = scaler.transform(sample)
sample_lda = lda.transform(sample_scaled)
sample_pred = classifier.predict(sample_lda)
print("\nPredicted species for sample:", iris.target_names[sample_pred][0
```

```
Predicted species for sample: setosa
```

```
/home/admin1/anaconda3/lib/python3.9/site-packages/sklearn/base.py:464: UserWarning: X does not have valid feature names, but StandardScaler was fitted with feature names
  warnings.warn(
```

In [ ]: