In [1]:
```
import pandas as pd
import numpy as np
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, confusion_matrix, classification_report
from sklearn import datasets
```

In [2]:
```
iris = datasets.load_iris()
X = iris.data
y = iris.target
```

In [3]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

In [4]:
```
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

In [5]:
```
model = LogisticRegression(multi_class='ovr', solver='lbfgs', max_iter=200)
model.fit(X_train, y_train)
```

Out[5]:
```
LogisticRegression(max_iter=200, multi_class='ovr')
```

In [6]:
```
LogisticRegression(max_iter=200, multi_class='ovr')
```

Out[6]:
```
LogisticRegression(max_iter=200, multi_class='ovr')
```

In [7]:
```
y_pred = model.predict(X_test)
```

In [8]:
```
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy of the Logistic Regression model:", accuracy)
```

```
Accuracy of the Logistic Regression model: 0.9666666666666667
```

In [9]:
```
conf_matrix = confusion_matrix(y_test, y_pred)
print("Confusion Matrix:\n", conf_matrix)
```

```
Confusion Matrix:
 [[10  0  0]
 [ 0  8  1]
 [ 0  0 11]]
```

In [10]:
```
class_report = classification_report(y_test, y_pred)
print("Classification Report:\n", class_report)
```

```
Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        10
           1       1.00      0.89      0.94         9
           2       0.92      1.00      0.96        11

    accuracy                           0.97        30
   macro avg       0.97      0.96      0.97        30
weighted avg       0.97      0.97      0.97        30
```

In [ ]: