```
pip install scikit-learn
```

```
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.10/dist-packages
Requirement already satisfied: numpy>=1.17.3 in /usr/local/lib/python3.10/dist-package
Requirement already satisfied: scipy>=1.3.2 in /usr/local/lib/python3.10/dist-packages
Requirement already satisfied: joblib>=1.1.1 in /usr/local/lib/python3.10/dist-package
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.10/dist-
```

```
import pandas as pd

# Assuming you've saved the dataset as 'iris.csv'
dataset_path = '/Iris.csv'
df = pd.read_csv(dataset_path)

# Display the first few rows of the dataset
print(df.head())
```

```
   Id  SepalLengthCm  SepalWidthCm  PetalLengthCm  PetalWidthCm      Species
0   1            5.1           3.5            1.4           0.2  Iris-setosa
1   2            4.9           3.0            1.4           0.2  Iris-setosa
2   3            4.7           3.2            1.3           0.2  Iris-setosa
3   4            4.6           3.1            1.5           0.2  Iris-setosa
4   5            5.0           3.6            1.4           0.2  Iris-setosa
```

```
X = df.drop('Species', axis=1)  # Features
y = df['Species']  # Target labels
```

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```
from sklearn.svm import SVC

model = SVC(kernel='linear')
```

```
model.fit(X_train, y_train)
```

```
▾        SVC
SVC(kernel='linear')
```

```
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix

y_pred = model.predict(X_test)

# Print evaluation metrics
print(f'Accuracy: {accuracy_score(y_test, y_pred)}')
print(f'Classification Report:\n{classification_report(y_test, y_pred)}')
print(f'Confusion Matrix:\n{confusion_matrix(y_test, y_pred)}')
```

```
Accuracy: 1.0
Classification Report:
                 precision    recall  f1-score   support

    Iris-setosa       1.00      1.00      1.00        10
Iris-versicolor       1.00      1.00      1.00         9
 Iris-virginica       1.00      1.00      1.00        11

       accuracy                           1.00        30
      macro avg       1.00      1.00      1.00        30
   weighted avg       1.00      1.00      1.00        30

Confusion Matrix:
```

```
new_data = [[5.1, 3.5, 1.4, 0.2, 2]]  # Replace with actual measurements
prediction = model.predict(new_data)
print(f'Predicted Species: {prediction}')
```

```
Predicted Species: ['Iris-setosa']
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:439: UserWarning: X does not h
  warnings.warn(
```