In [6]:
```
import pandas as pd
from sklearn.datasets import load_iris
import numpy as np
```

In [2]:
```
data = load_iris()
iris_df = pd.DataFrame(data.data, columns=data.feature_names)
iris_df['species'] = data.target_names[data.target]
```

In [3]:
```
iris_df.head()
```

Out[3]:

| | sepal length (cm) | sepal width (cm) | petal length (cm) | petal width (cm) | species |
|---|---|---|---|---|---|
| **0** | 5.1 | 3.5 | 1.4 | 0.2 | setosa |
| **1** | 4.9 | 3.0 | 1.4 | 0.2 | setosa |
| **2** | 4.7 | 3.2 | 1.3 | 0.2 | setosa |
| **3** | 4.6 | 3.1 | 1.5 | 0.2 | setosa |
| **4** | 5.0 | 3.6 | 1.4 | 0.2 | setosa |

In [4]:
```
iris_df.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 (cm)  150 non-null    float64
 1   sepal width (cm)   150 non-null    float64
 2   petal length (cm)  150 non-null    float64
 3   petal width (cm)   150 non-null    float64
 4   species            150 non-null    object 
dtypes: float64(4), object(1)
memory usage: 6.0+ KB
```

In [9]:
```
iris_df.describe(include="all")
```

Out[9]:

| | sepal length (cm) | sepal width (cm) | petal length (cm) | petal width (cm) | species |
|---|---|---|---|---|---|
| **count** | 150.000000 | 150.000000 | 150.000000 | 150.000000 | 150 |
| **unique** | NaN | NaN | NaN | NaN | 3 |
| **top** | NaN | NaN | NaN | NaN | setosa |
| **freq** | NaN | NaN | NaN | NaN | 50 |
| **mean** | 5.843333 | 3.057333 | 3.758000 | 1.199333 | NaN |
| **std** | 0.828066 | 0.435866 | 1.765298 | 0.762238 | NaN |
| **min** | 4.300000 | 2.000000 | 1.000000 | 0.100000 | NaN |
| **25%** | 5.100000 | 2.800000 | 1.600000 | 0.300000 | NaN |
| **50%** | 5.800000 | 3.000000 | 4.350000 | 1.300000 | NaN |
| **75%** | 6.400000 | 3.300000 | 5.100000 | 1.800000 | NaN |
| **max** | 7.900000 | 4.400000 | 6.900000 | 2.500000 | NaN |

In [10]:
```
iris_df.isnull().sum()
```

Out[10]:
```
sepal length (cm)    0
sepal width (cm)     0
petal length (cm)    0
petal width (cm)     0
species              0
dtype: int64
```

In [11]:
```
iris_df["species"].value_counts()
```

Out[11]:
```
setosa        50
versicolor    50
virginica     50
Name: species, dtype: int64
```

In [12]:
```
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
```

In [13]:
```
X = iris_df.drop('species', axis=1)
y = iris_df['species']
```

In [14]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=
```

In [15]:
```
svm_model = SVC()
```

In [16]:
```
svm_model.fit(X_train, y_train)
```

Out[16]:
```
SVC()
```

In [17]:
```
y_pred = svm_model.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)
```

```
Accuracy: 1.0
```

In [18]:
```
print("\nClassification Report:")
print(classification_report(y_test, y_pred))
```

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

      setosa       1.00      1.00      1.00        10
  versicolor       1.00      1.00      1.00         9
   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
```

In [19]:
```
print("\nConfusion Matrix:")
print(confusion_matrix(y_test, y_pred))
```

```
Confusion Matrix:
[[10  0  0]
 [ 0  9  0]
 [ 0  0 11]]
```