

```
In [2]: #Step-01 Import the necessary libraries
import numpy as np
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score
import numpy as np
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score
```

```
In [3]: #Step 2: Load the Iris dataset
iris = load_iris()
X = iris.data # Input features
y = iris.target # Target variable
```

```
In [4]: #Step 3: Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```
In [5]: #Step 4: Define and train the K-Nearest Neighbors (KNN) classifier
knn = KNeighborsClassifier(n_neighbors=3) # Create a KNN classifier with 3 neighbors
knn.fit(X_train, y_train) # Train the classifier on the training data
```

```
Out[5]: KNeighborsClassifier(n_neighbors=3)
```

```
In [6]: #Step 5: Make predictions on the test set
y_pred = knn.predict(X_test) # Predict the class labels for the test set
```

```
C:\Users\dnhac\anaconda3\lib\site-packages\sklearn\neighbors\_classification.py:211: FutureWarning: Unlike other reduction functions (e.g. `skew`, `kurtosis`), the default behavior of `mode` typically preserves the axis it acts along. In SciPy 1.11.0, this behavior will change: the default value of `keepdims` will become False, the `axis` over which the statistic is taken will be eliminated, and the value None will no longer be accepted. Set `keepdims` to True or False to avoid this warning.
mode, _ = stats.mode(_y[neigh_ind, k], axis=1)
```

```
In [7]: #Step 6: Calculate the accuracy of the classifier
accuracy = accuracy_score(y_test, y_pred) # Compare the predicted labels with the true labels
print("Accuracy:", accuracy)
```

```
Accuracy: 1.0
```

```
In [ ]:
```