

```

1  # Import necessary modules
2  from sklearn.neighbors import KNeighborsClassifier
3  from sklearn.model_selection import train_test_split
4  from sklearn.datasets import load_iris
5
6  # Loading data
7  irisData = load_iris()
8
9  # Create feature and target arrays
10 X = irisData.data
11 y = irisData.target
12
13 # Split into training and test set
14 X_train, X_test, y_train, y_test = train_test_split(
15     X, y, test_size = 0.2, random_state=42)
16
17 knn = KNeighborsClassifier(n_neighbors=7)
18
19 knn.fit(X_train, y_train)
20
21 # Calculate the accuracy of the model
22 print(knn.score(X_test, y_test))

```

0.9666666666666667

```

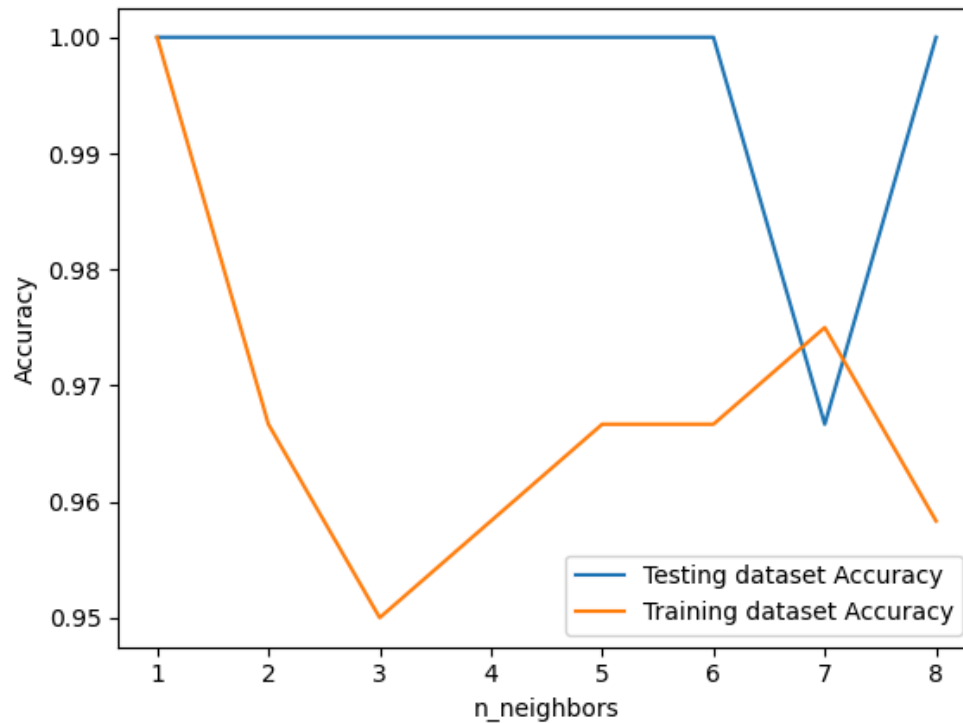
1  # Import necessary modules
2  from sklearn.neighbors import KNeighborsClassifier
3  from sklearn.model_selection import train_test_split
4  from sklearn.datasets import load_iris
5  import numpy as np
6  import matplotlib.pyplot as plt
7
8  irisData = load_iris()
9
10 # Create feature and target arrays
11 X = irisData.data
12 y = irisData.target
13
14 # Split into training and test set
15 X_train, X_test, y_train, y_test = train_test_split(
16     X, y, test_size = 0.2, random_state=42)
17
18 neighbors = np.arange(1, 9)
19 train_accuracy = np.empty(len(neighbors))
20 test_accuracy = np.empty(len(neighbors))
21
22 # Loop over K values
23 for i, k in enumerate(neighbors):
24     knn = KNeighborsClassifier(n_neighbors=k)
25     knn.fit(X_train, y_train)
26
27     # Compute training and test data accuracy
28     train_accuracy[i] = knn.score(X_train, y_train)
29     test_accuracy[i] = knn.score(X_test, y_test)
30
31 # Generate plot
32 plt.plot(neighbors, test_accuracy, label = 'Testing dataset Accuracy')
33 plt.plot(neighbors, train_accuracy, label = 'Training dataset Accuracy')

```

```

34
35 plt.legend()
36 plt.xlabel('n_neighbors')
37 plt.ylabel('Accuracy')
38 plt.show()

```



```

1 # Import necessary modules
2 from sklearn.neighbors import KNeighborsClassifier
3 from sklearn.model_selection import train_test_split
4 from sklearn.datasets import load_iris
5
6 # Loading data
7 irisData = load_iris()
8
9 # Create feature and target arrays
10 X = irisData.data
11 y = irisData.target
12
13 # Split into training and test set
14 X_train, X_test, y_train, y_test = train_test_split(
15     X, y, test_size = 0.2, random_state=42)
16
17 knn = KNeighborsClassifier(n_neighbors=7)
18
19 knn.fit(X_train, y_train)
20
21 # Predict on dataset which model has not seen before
22 print(knn.predict(X_test))

```

[1 0 2 1 1 0 1 2 2 1 2 0 0 0 0 1 2 1 1 2 0 2 0 2 2 2 2 2 0 0]

