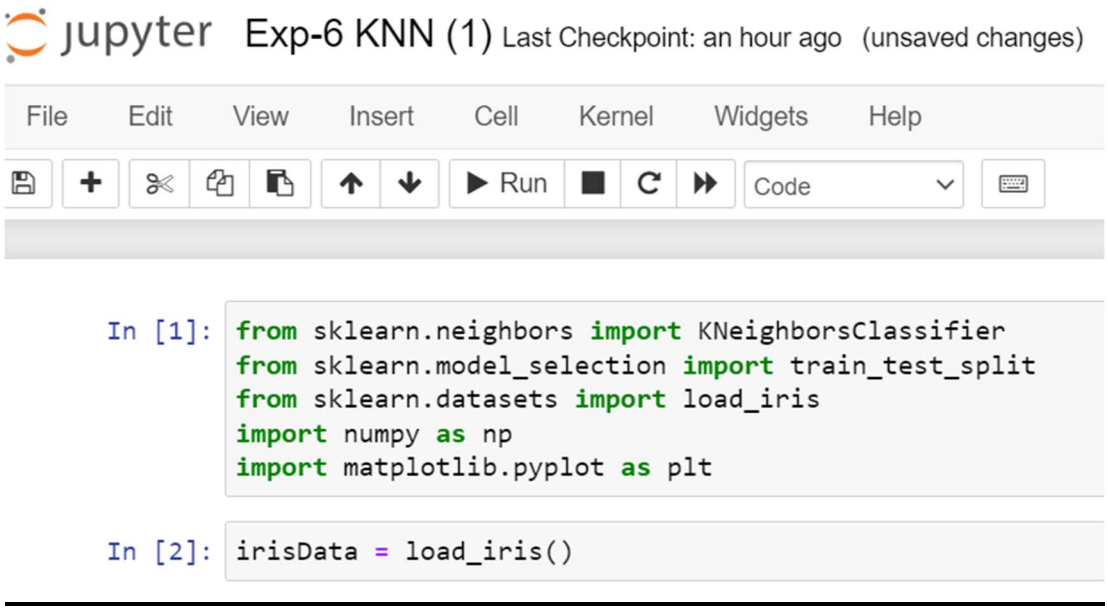
Experiment Title: 2.3	
Student Name: Pritam Kumar Dutta	**UID:** 20BCS3296
Branch: CSE	**Section/Group:** 606-A
Semester: 5	**Date of Performance:** Oct. 11, 2022
Subject Name: Machine Learning Lab	**Subject Code:** 20CSP-317

❖ **Aim/Overview of the practical:** Implement K-Nearest Neighbor on any data set.

❖ **Code & Output:**



In [3]:
```
print(irisData.feature_names)
```

```
['sepal length (cm)', 'sepal width (cm)', 'petal length (cm)', 'petal width (cm)']
```

In [4]:
```
print(irisData.target_names)
```

```
['setosa' 'versicolor' 'virginica']
```

In [5]:
```
# Creating feature and target arrays
X = irisData.data
y = irisData.target
```

In [6]:
```
# Splitting into training and test set
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state=0)
```

In [7]:
```
neighbors = np.arange(1, 15)
print(neighbors)
```

```
[ 1  2  3  4  5  6  7  8  9 10 11 12 13 14]
```

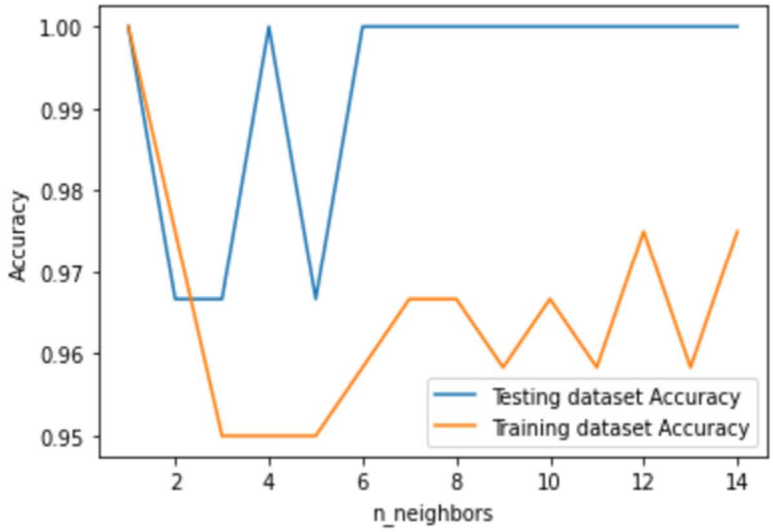
In [8]:
```
train_accuracy = np.empty(len(neighbors))
test_accuracy = np.empty(len(neighbors))
```

In [9]:
```
# Looping over K values
for i, k in enumerate(neighbors):
    knn = KNeighborsClassifier(n_neighbors=k)
    knn.fit(X_train, y_train)

    # Computing training and test data accuracy
    train_accuracy[i] = knn.score(X_train, y_train)
    test_accuracy[i] = knn.score(X_test, y_test)
```

In [10]:
```
# Generating plot
plt.plot(neighbors, test_accuracy, label = 'Testing dataset Accuracy')
plt.plot(neighbors, train_accuracy, label = 'Training dataset Accuracy')

plt.legend()
plt.xlabel('n_neighbors')
plt.ylabel('Accuracy')
plt.show()
```



In [11]:
```
# Using k=5
knn = KNeighborsClassifier(n_neighbors=5)
knn.fit(X_train, y_train)
```

Out[11]:
```
KNeighborsClassifier()
```

In [12]:
```
y_pred = knn.predict(X_test)
```

In [13]:
```
y_pred
```

Out[13]:
```
array([2, 1, 0, 2, 0, 2, 0, 1, 1, 1, 2, 1, 1, 1, 2, 0, 1, 1, 0, 0, 2, 1,
       0, 0, 2, 0, 0, 1, 1, 0])
```

In [14]:
```
from sklearn.metrics import confusion_matrix, classification_report
print("\nConfusion Matrix:-\n",confusion_matrix(y_test,y_pred))
print("\nClassification Report:-\n", classification_report(y_test,y_pred))
```

```
Confusion Matrix:-
 [[11  0  0]
 [ 0 12  1]
 [ 0  0  6]]

Classification Report:-
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        11
           1       1.00      0.92      0.96        13
           2       0.86      1.00      0.92         6

    accuracy                           0.97        30
   macro avg       0.95      0.97      0.96        30
weighted avg       0.97      0.97      0.97        30
```

In [ ]:

❖ Learning outcomes (What I have learnt):

1. We learned about data analysis and data handling in python.
2. We learned about various basic functions and libraries required for data analysis using python.
3. We learned to implement K-Nearest Algorithm on any dataset in python.
4. We learned to verify accuracy of the KNN model.

Evaluation Grid :

Sr. No.	Parameters	Marks Obtained	Maximum Marks
1.	Student Performance (Conduct of experiment) objectives/Outcomes.		12
2.	Viva Voce		10
3.	Submission of Work Sheet (Record)		8
	Total		30