گزارش تمرین دوم داده کاوی

سوال ۱

۱-۱:

In [1]:
```
from sklearn.datasets import load_breast_cancer
BreastCancer = load_breast_cancer()
X=BreastCancer.DESCR
print (X)
```

```
.. _breast_cancer_dataset:

Breast cancer wisconsin (diagnostic) dataset
--------------------------------------------

**Data Set Characteristics:**

    :Number of Instances: 569

    :Number of Attributes: 30 numeric, predictive attributes and the class

    :Attribute Information:
        - radius (mean of distances from center to points on the perimeter)
        - texture (standard deviation of gray-scale values)
        - perimeter
        - area
        - smoothness (local variation in radius lengths)
        - compactness (perimeter^2 / area - 1.0)
        - concavity (severity of concave portions of the contour)
        - concave points (number of concave portions of the contour)
```

۱-۲:

In [3]:
```
x=BreastCancer.feature_names
print (x)
```

```
['mean radius' 'mean texture' 'mean perimeter' 'mean area'
 'mean smoothness' 'mean compactness' 'mean concavity'
 'mean concave points' 'mean symmetry' 'mean fractal dimension'
 'radius error' 'texture error' 'perimeter error' 'area error'
 'smoothness error' 'compactness error' 'concavity error'
 'concave points error' 'symmetry error' 'fractal dimension error'
 'worst radius' 'worst texture' 'worst perimeter' 'worst area'
 'worst smoothness' 'worst compactness' 'worst concavity'
 'worst concave points' 'worst symmetry' 'worst fractal dimension']
```

In [4]:
```
x=BreastCancer.data
print (x)
```

```
[[1.799e+01 1.038e+01 1.228e+02 ... 2.654e-01 4.601e-01 1.189e-01]
 [2.057e+01 1.777e+01 1.329e+02 ... 1.860e-01 2.750e-01 8.902e-02]
 [1.969e+01 2.125e+01 1.300e+02 ... 2.430e-01 3.613e-01 8.758e-02]
 ...
 [1.660e+01 2.808e+01 1.083e+02 ... 1.418e-01 2.218e-01 7.820e-02]
 [2.060e+01 2.933e+01 1.401e+02 ... 2.650e-01 4.087e-01 1.240e-01]
 [7.760e+00 2.454e+01 4.792e+01 ... 0.000e+00 2.871e-01 7.039e-02]]
```

In [5]:
```
x=BreastCancer.keys()
print (x)
```

```
['target_names', 'data', 'target', 'DESCR', 'feature_names']
```

۱-۳:

```
In [6]: import pandas as pd
        import numpy as np
        DF= pd.DataFrame(BreastCancer.data, columns=BreastCancer.feature_names)
```

۱-۴:

```
In [7]: DF.describe()
```

Out[7]:

	mean radius	mean texture	mean perimeter	mean area	mean smoothness	mean compactness	mean concavity	mean concave points	mean symmetry	mean fractal dimension	...	worst radius	wors textur
count	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	...	569.000000	569.00000
mean	14.127292	19.289649	91.969033	654.889104	0.096360	0.104341	0.088799	0.048919	0.181162	0.062798	...	16.269190	25.67722
std	3.524049	4.301036	24.298981	351.914129	0.014064	0.052813	0.079720	0.038803	0.027414	0.007060	...	4.833242	6.14625
min	6.981000	9.710000	43.790000	143.500000	0.052630	0.019380	0.000000	0.000000	0.106000	0.049960	...	7.930000	12.02000
25%	11.700000	16.170000	75.170000	420.300000	0.086370	0.064920	0.029560	0.020310	0.161900	0.057700	...	13.010000	21.08000
50%	13.370000	18.840000	86.240000	551.100000	0.095870	0.092630	0.061540	0.033500	0.179200	0.061540	...	14.970000	25.41000
75%	15.780000	21.800000	104.100000	782.700000	0.105300	0.130400	0.130700	0.074000	0.195700	0.066120	...	18.790000	29.72000
max	28.110000	39.280000	188.500000	2501.000000	0.163400	0.345400	0.426800	0.201200	0.304000	0.097440	...	36.040000	49.54000

8 rows × 30 columns

۱-۵:

```
In [8]: DF['target'] = BreastCancer.target
```

۱-۶:

```
In [9]: DF['target'].value_counts()
```

```
Out[9]: 1    357
        0    212
        Name: target, dtype: int64
```

```
In [10]: x=BreastCancer.target_names
         print (x)
```

```
['malignant' 'benign']
```

۱-۷:

```
In [11]: from sklearn.model_selection import train_test_split
         X = df[BreastCancer['feature_names']]
         y = df['target']
         X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=0)
```

۱-۸:

```
In [12]: X_train.shape
Out[12]: (426, 30)

In [13]: X_test.shape
Out[13]: (143, 30)

In [14]: y_train.shape
Out[14]: (426L,)

In [15]: y_test.shape
Out[15]: (143L,)
```

۱-۹:

```
In [32]: from sklearn.neighbors import KNeighborsClassifier
         knn = KNeighborsClassifier(n_neighbors = 6)
         knn.fit(X_train, y_train)
         meanAccuracy = knn.score(X_test, y_test)
         meanAccuracy

Out[32]: 0.9230769230769231
```

۱-۱۰:

```
In [33]: ansTest = knn.predict(X_test)
         ansTest

Out[33]: array([0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0,
                1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 1, 0, 1,
                0, 0, 1, 1, 1, 0, 0, 1, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 0, 1,
                0, 0, 0, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 0, 0, 0, 1, 0, 1, 1, 1,
                0, 0, 1, 0, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0, 0,
                1, 0, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 0, 1,
                1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 0])
```

۱-۱۱:

۱-۱۲:

```
In [62]: from sklearn.preprocessing import MinMaxScaler
         minmax = MinMaxScaler()
         minmax.fit(X)
         normalized_Xtrain = minmax.transform(X_train)
         normalized_Xtest = minmax.transform(X_test)
```

۱-۱۳:

In [64]:
```
knnn = KNeighborsClassifier(n_neighbors = 6)
knnn.fit(normalized_Xtrain,y_train)
```

Out[64]:
```
KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
           metric_params=None, n_jobs=1, n_neighbors=6, p=2,
           weights='uniform')
```

:۱-۱۴

In [67]:
```
meanAccuracy_train = knnn.score(normalized_Xtrain, y_train)
meanAccuracy_test = knnn.score(normalized_Xtest, y_test)
print (meanAccuracy_train)
print (meanAccuracy_test)
```

```
0.9671361502347418
0.965034965034965
```

:۱-۱۵

In [21]:
```
training_accuracy = []
test_accuracy = []

neighbors = range (1, 11)

for i in neighbors:
    knnnn = KNeighborsClassifier(n_neighbors = i)
    knnnn.fit(normalized_Xtrain,y_train)
    training_accuracy.append(knnnn.score(normalized_Xtrain, y_train))
    test_accuracy.append(knnnn.score(normalized_Xtest, y_test))

print(training_accuracy)
print(test_accuracy)
```

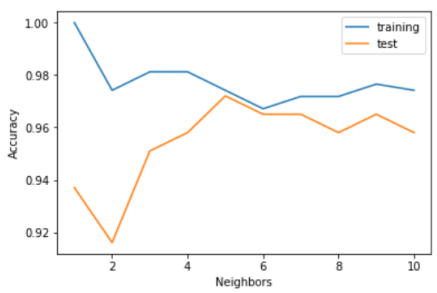
```
[1.0, 0.9741784037558685, 0.9812206572769953, 0.9812206572769953, 0.9741784037558685, 0.9671361502347418, 0.971830985
915493, 0.971830985915493, 0.9765258215962441, 0.9741784037558685]
[0.9370629370629371, 0.916083916083916, 0.951048951048951, 0.958041958041958, 0.972027972027972, 0.965034965034965,
0.965034965034965, 0.958041958041958, 0.965034965034965, 0.958041958041958]
```

:۱-۱۶

In [23]:
```
import matplotlib.pyplot as plt

plt.plot(neighbors, training_accuracy, label='training')
plt.plot(neighbors, test_accuracy, label='test')
plt.ylabel('Accuracy')
plt.xlabel('Neighbors')
plt.legend()
```

Out[23]:
```
<matplotlib.legend.Legend at 0x26a09253470>
```



۱۷-۱: