

جواب سوال 12 و 13 و 14

```
# By using all the features in the training dataset with 2 of neighbors
knn = KNeighborsClassifier(n_neighbors = 2)
X_train, X_test, y_train, y_test = tts( x , y , test_size = 0.2, random_state = 0)
knn.fit(X_train,y_train)
pred = knn.predict(X_test)
knn.score(X_test,y_test)
print(confusion_matrix(y_test,pred))
print(classification_report(y_test,pred))
```

```
[[26 3]
 [ 8 21]]
      precision    recall  f1-score   support

    0       0.76      0.90      0.83        29
    1       0.88      0.72      0.79        29

 accuracy          0.81        58
 macro avg      0.82      0.81      0.81        58
weighted avg      0.82      0.81      0.81        58
```

```
# By using all the features in the training dataset with 4 of neighbors
knn = KNeighborsClassifier(n_neighbors = 4)
X_train, X_test, y_train, y_test = tts( x , y , test_size = 0.2, random_state = 0)
knn.fit(X_train,y_train)
pred = knn.predict(X_test)
knn.score(X_test,y_test)
print(confusion_matrix(y_test,pred))
print(classification_report(y_test,pred))
```

```
[[26 3]
 [ 5 24]]
      precision    recall  f1-score   support

    0       0.84      0.90      0.87        29
    1       0.89      0.83      0.86        29

 accuracy          0.86        58
 macro avg      0.86      0.86      0.86        58
weighted avg      0.86      0.86      0.86        58
```

اگر KNN را برابر با 1 قرار دهیم در این صورت 1-نزدیکترین همسایه با خطای مدل 0.827 در کلاس 1 قرار خواهد گرفت.

اگر KNN را برابر با 2 قرار دهیم در این صورت 2-نزدیکترین همسایه با خطای مدل 0.810 در کلاس 0 قرار خواهد گرفت.

اگر KNN را برابر با 3 قرار دهیم در این صورت 3-نزدیکترین همسایه با خطای مدل 0.827 امتیاز قرار گرفتن در هر دو کلاس برابر خواهد بود.

اگر KNN را برابر با 4 قرار دهیم در این صورت 4-نزدیکترین همسایه با خطای مدل 0.862 در کلاس 0 قرار خواهد گرفت.

اگر KNN را برابر با 5 قرار دهیم در این صورت 5-نزدیکترین همسایه با خطای مدل 0.862 در کلاس 1 قرار خواهد گرفت.

اگر KNN را برابر با 6 قرار دهیم در این صورت 6-نزدیکترین همسایه با خطای مدل 0.844 در کلاس 1 قرار خواهد گرفت.

اگر KNN را برابر با 7 قرار دهیم در این صورت 7-نزدیکترین همسایه با خطای مدل 0.879 در کلاس 1 قرار خواهد گرفت.

اگر KNN را برابر با 8 قرار دهیم در این صورت 8-نزدیکترین همسایه با خطای مدل 0.879 در کلاس 1 قرار خواهد گرفت.

اگر KNN را برابر با 9 قرار دهیم در این صورت 9-نزدیکترین همسایه با خطای مدل 0.879 در کلاس 1 قرار خواهد گرفت.

⋮

نتیجه : با زیاد شدن تعداد همسایه ها همچنان در کلاس 1 قرار خواهیم داشت. پس به ازای $K > 5$ در کلاس 1 خواهیم بود و از آنجایی که مقدار خطای مدل به ازای $K > 5$ در $K = 6$ کمترین است می توان نتیجه گرفت K بهینه عدد 6 است.

یعنی $K = 6$ بهینه است.

By using features (trestbps , chol , thalach) in the training dataset with 2 of neighbors

```
knn = KNeighborsClassifier(n_neighbors = 2)
x1 = ds[['trestbps', 'chol', 'thalach']]
X_train, X_test, y_train, y_test = tts(x1, y, test_size = 0.2, random_state = 0)
knn.fit(X_train, y_train)
pred = knn.predict(X_test)
knn.score(X_test, y_test)
print(confusion_matrix(y_test, pred))
print(classification_report(y_test, pred))
```

```
[[21  8]
```

```
[18 11]]
```

```
precision  recall  f1-score  support
```

```
0    0.54    0.72    0.62    29
```

```
1    0.58    0.38    0.46    29
```

```
accuracy                0.55    58
```

```
macro avg    0.56    0.55    0.54    58
```

```
weighted avg    0.56    0.55    0.54    58
```