```
import pandas as pd
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import confusion_matrix
import matplotlib.pyplot as plt
from sklearn.datasets import load_digits


digits = load_digits()


dir(digits)
```

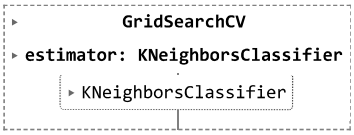
```
['DESCR', 'data', 'feature_names', 'frame', 'images', 'target', 'target_names']
```

```
df = pd.DataFrame(digits.data, columns=digits.feature_names)
df['target'] = digits.target
df.head()
```

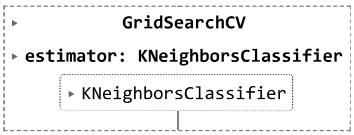
	pixel_0_0	pixel_0_1	pixel_0_2	pixel_0_3	pixel_0_4	pixel_0_5	pixel_0_6	pixel_0_
0	0.0	0.0	5.0	13.0	9.0	1.0	0.0	0
1	0.0	0.0	0.0	12.0	13.0	5.0	0.0	0
2	0.0	0.0	0.0	4.0	15.0	12.0	0.0	0
3	0.0	0.0	7.0	15.0	13.0	1.0	0.0	0
4	0.0	0.0	0.0	1.0	11.0	0.0	0.0	0

5 rows × 65 columns

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(df.drop('target',axis='columns'), df.target, test_size=0.3, random_state=2)


clf = GridSearchCV(KNeighborsClassifier(),{
    'n_neighbors': [3,4,5,6,7,8,9,10]
}, cv=5, return_train_score=False)

clf.fit(X_train, y_train)
```




```
GridSearchCV(cv=5, estimator=KNeighborsClassifier(),
             param_grid={'n_neighbors': [3, 4, 5, 6, 7, 8, 9, 10]})
```




```
pd.DataFrame(clf.cv_results_)
```

	mean_fit_time	std_fit_time	mean_score_time	std_score_time	param_n_neighbors	params	split0_test_score	split1_test_score
0	0.012791	0.009441	0.178308	0.107719	3	{'n_neighbors': 3}	0.992063	0.984127
1	0.017386	0.020072	0.093857	0.030691	4	{'n_neighbors': 4}	0.992063	0.980159
2	0.003609	0.000826	0.021674	0.004181	5	{'n_neighbors': 5}	0.988095	0.984127

```
knn = KNeighborsClassifier(n_neighbors=5)
knn.fit(X_train, y_train)
```

```
KNeighborsClassifier()
```

```
knn.score(X_test, y_test)
```

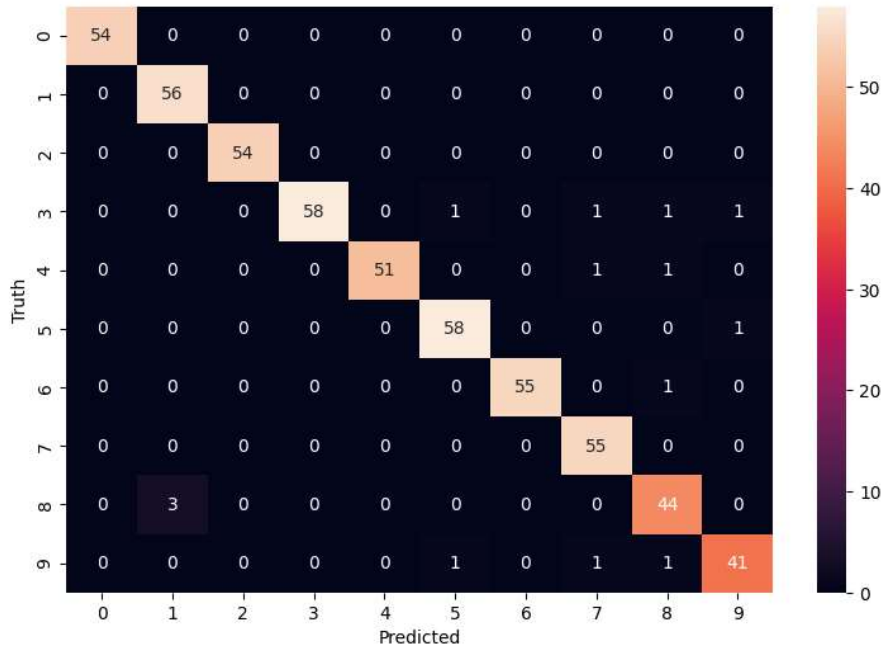
```
0.9740740740740741
```

```
from sklearn.metrics import confusion_matrix
y_pred = knn.predict(X_test)

cm = confusion_matrix(y_test, y_pred)
cm
```

```
array([[54,  0,  0,  0,  0,  0,  0,  0,  0,  0],
       [ 0, 56,  0,  0,  0,  0,  0,  0,  0,  0],
       [ 0,  0, 54,  0,  0,  0,  0,  0,  0,  0],
       [ 0,  0,  0, 58,  0,  1,  0,  1,  1,  1],
       [ 0,  0,  0,  0, 51,  0,  0,  1,  1,  0],
       [ 0,  0,  0,  0,  0, 58,  0,  0,  0,  1],
       [ 0,  0,  0,  0,  0,  0, 55,  0,  1,  0],
       [ 0,  0,  0,  0,  0,  0,  0, 55,  0,  0],
       [ 0,  3,  0,  0,  0,  0,  0,  0, 44,  0],
       [ 0,  0,  0,  0,  0,  1,  0,  1,  1, 41]])
```

```
import seaborn as sn
from matplotlib import pyplot as plt

plt.figure(figsize=(9,6))
sn.heatmap(cm, annot=True)
plt.xlabel("Predicted")
plt.ylabel("Truth")
```

```
Text(83.22222222222221, 0.5, 'Truth')
```



```
from sklearn.metrics import classification_report
print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        54
           1       0.95      1.00      0.97        56
           2       1.00      1.00      1.00        54
           3       1.00      0.94      0.97        62
           4       1.00      0.96      0.98        53
           5       0.97      0.98      0.97        59
           6       1.00      0.98      0.99        56
           7       0.95      1.00      0.97        55
           8       0.92      0.94      0.93        47
           9       0.95      0.93      0.94        44

    accuracy                           0.97       540
   macro avg       0.97      0.97      0.97       540
weighted avg       0.97      0.97      0.97       540
```