

```
In [11]: import pandas as pd
import numpy as np
from sklearn import datasets, svm, metrics
```

```
In [12]: #Digits verisetinin çekilmesi
df = datasets.load_digits()
n_samples = len(datasets.load_digits().images)
n_samples
```

Out[12]: 1797

```
In [13]: data = datasets.load_digits().images.reshape((n_samples, -1))
data
```

```
Out[13]: array([[ 0.,  0.,  5., ...,  0.,  0.,  0.],
 [ 0.,  0.,  0., ..., 10.,  0.,  0.],
 [ 0.,  0.,  0., ..., 16.,  9.,  0.],
 ...,
 [ 0.,  0.,  1., ...,  6.,  0.,  0.],
 [ 0.,  0.,  2., ..., 12.,  0.,  0.],
 [ 0.,  0., 10., ..., 12.,  1.,  0.]])
```

```
In [14]: X = data
Y = df.target
Y
```

Out[14]: array([0, 1, 2, ..., 8, 9, 8])

```
In [15]: from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(X,Y,test_size=0.33)
```

```
In [18]: model = svm.SVC(kernel="poly",degree=3,gamma="auto")
model.fit(x_train,y_train)
```

Out[18]: SVC(gamma='auto', kernel='poly')

```
In [20]: y_predicted = model.predict(x_test)
y_predicted
```

```
Out[20]: array([4, 2, 4, 3, 6, 4, 9, 2, 2, 8, 0, 8, 6, 3, 1, 0, 5, 0, 5, 9, 3, 8,
 0, 8, 1, 1, 7, 1, 2, 0, 7, 9, 5, 2, 4, 5, 0, 7, 3, 2, 0, 7, 5, 2,
 2, 1, 4, 4, 4, 5, 5, 4, 7, 7, 4, 4, 5, 2, 5, 9, 5, 1, 5, 0, 1, 5,
 2, 9, 7, 3, 9, 1, 0, 1, 8, 5, 3, 3, 7, 6, 3, 7, 4, 8, 8, 1, 7, 4,
 8, 0, 9, 3, 7, 1, 8, 3, 2, 9, 3, 3, 1, 9, 1, 6, 5, 6, 7, 1, 6, 9,
 6, 6, 3, 3, 9, 3, 4, 9, 8, 1, 3, 3, 2, 2, 1, 5, 4, 2, 1, 6, 5, 5,
 4, 9, 5, 7, 0, 4, 7, 2, 2, 9, 2, 9, 1, 5, 1, 8, 7, 6, 1, 0, 8, 8,
 2, 5, 1, 4, 8, 3, 2, 9, 1, 2, 6, 0, 7, 9, 5, 3, 9, 1, 8, 1, 9, 7,
 8, 0, 1, 1, 5, 0, 3, 3, 5, 0, 4, 8, 6, 8, 5, 5, 5, 1, 6, 3, 5, 7,
 1, 0, 7, 9, 2, 9, 9, 5, 8, 2, 7, 5, 8, 5, 9, 4, 3, 6, 7, 0, 2, 1,
 2, 7, 1, 2, 8, 4, 7, 7, 0, 6, 4, 8, 9, 3, 2, 4, 8, 3, 0, 9, 4, 0,
 5, 8, 1, 7, 3, 9, 4, 5, 7, 1, 4, 8, 7, 1, 7, 2, 1, 7, 2, 6, 5, 3,
 6, 7, 5, 2, 4, 0, 9, 2, 6, 4, 5, 5, 8, 6, 1, 6, 4, 3, 1, 8, 9, 3,
 9, 9, 4, 3, 9, 6, 2, 9, 5, 7, 2, 9, 6, 6, 6, 3, 7, 7, 2, 2, 2, 6,
 4, 1, 7, 5, 0, 9, 3, 8, 9, 2, 9, 8, 4, 6, 8, 4, 5, 0, 6, 4, 4, 5,
 7, 8, 9, 7, 9, 3, 0, 0, 0, 2, 8, 8, 5, 8, 4, 5, 9, 4, 4, 5, 4, 7,
 5, 8, 9, 1, 3, 5, 7, 2, 9, 2, 8, 2, 3, 5, 8, 9, 1, 8, 1, 6, 2, 1,
 2, 6, 1, 6, 4, 7, 2, 5, 3, 0, 9, 2, 9, 4, 9, 8, 2, 0, 2, 0, 9, 1,
 8, 1, 4, 7, 6, 1, 7, 0, 2, 9, 0, 3, 8, 2, 1, 3, 7, 7, 7, 8, 9, 8,
```

```
3, 3, 9, 2, 4, 4, 4, 7, 6, 0, 2, 3, 3, 6, 2, 7, 2, 6, 9, 1, 7, 4,
8, 7, 0, 0, 3, 1, 6, 2, 0, 3, 9, 8, 4, 3, 6, 5, 4, 8, 2, 3, 0, 5,
2, 4, 4, 8, 4, 1, 3, 8, 8, 7, 9, 7, 5, 3, 8, 4, 8, 9, 0, 2, 3, 0,
9, 9, 4, 0, 1, 3, 0, 6, 2, 7, 6, 9, 7, 7, 8, 5, 0, 4, 1, 0, 0, 6,
1, 6, 3, 3, 8, 4, 5, 8, 5, 2, 6, 2, 1, 0, 6, 1, 0, 2, 1, 2, 3, 4,
9, 5, 3, 2, 6, 3, 7, 6, 3, 3, 3, 4, 5, 8, 2, 9, 0, 4, 8, 6, 0, 1,
6, 9, 2, 1, 4, 0, 3, 9, 7, 3, 0, 6, 3, 4, 6, 7, 0, 7, 5, 3, 2, 1,
1, 9, 3, 1, 6, 2, 3, 9, 9, 0, 3, 3, 5, 8, 5, 9, 4, 0, 2, 2, 5, 0])
```

```
In [21]: r2 = metrics.r2_score(y_test,y_predicted)
r2
```

```
Out[21]: 0.9704139861156544
```

```
In [22]: acc = metrics.accuracy_score(y_test,y_predicted)
acc
```

```
Out[22]: 0.9814814814814815
```

```
In [ ]:
```