

# Exercise

Train svm classifier using sklearn digits dataset (i.e. from sklearn.datasets import load\_digits) and then:

1. Use 80% of samples as training data size.
2. Measure accuracy of your model using different kernels such as rbf and linear.
3. Tune your model further using regularization and gamma parameters and try to come up with highest accuracy score.

```
In [1]: import pandas as pd
        from sklearn.datasets import load_digits
        digits = load_digits()
```

```
In [2]: digits.target
```

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

```
In [3]: dir(digits)
```

```
Out[3]: ['DESCR', 'data', 'feature_names', 'frame', 'images', 'target', 'target_names']
```

```
In [4]: digits.target_names
```

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

```
In [5]: df = pd.DataFrame(digits.data, digits.target)
        df.head()
```

```
Out[5]:
```

	0	1	2	3	4	5	6	7	8	9	...	54	55	56	57	58	59	60	61	62
0	0.0	0.0	5.0	13.0	9.0	1.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	6.0	13.0	10.0	0.0	0.0
1	0.0	0.0	0.0	12.0	13.0	5.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	11.0	16.0	10.0	0.0
2	0.0	0.0	0.0	4.0	15.0	12.0	0.0	0.0	0.0	0.0	...	5.0	0.0	0.0	0.0	0.0	3.0	11.0	16.0	9.0
3	0.0	0.0	7.0	15.0	13.0	1.0	0.0	0.0	0.0	8.0	...	9.0	0.0	0.0	0.0	7.0	13.0	13.0	9.0	0.0
4	0.0	0.0	0.0	1.0	11.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	2.0	16.0	4.0	0.0

5 rows × 64 columns



```
In [6]: df['target'] = digits.target
df.head(10)
```

Out[6]:

	0	1	2	3	4	5	6	7	8	9	...	55	56	57	58	59	60	61	62
0	0.0	0.0	5.0	13.0	9.0	1.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	6.0	13.0	10.0	0.0	0.0
1	0.0	0.0	0.0	12.0	13.0	5.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	11.0	16.0	10.0	0.0
2	0.0	0.0	0.0	4.0	15.0	12.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	3.0	11.0	16.0	9.0
3	0.0	0.0	7.0	15.0	13.0	1.0	0.0	0.0	0.0	8.0	...	0.0	0.0	0.0	7.0	13.0	13.0	9.0	0.0
4	0.0	0.0	0.0	1.0	11.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	2.0	16.0	4.0	0.0
5	0.0	0.0	12.0	10.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	9.0	16.0	16.0	10.0	0.0
6	0.0	0.0	0.0	12.0	13.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	1.0	9.0	15.0	11.0	3.0
7	0.0	0.0	7.0	8.0	13.0	16.0	15.0	1.0	0.0	0.0	...	0.0	0.0	0.0	13.0	5.0	0.0	0.0	0.0
8	0.0	0.0	9.0	14.0	8.0	1.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	11.0	16.0	15.0	11.0	1.0
9	0.0	0.0	11.0	12.0	0.0	0.0	0.0	0.0	0.0	2.0	...	0.0	0.0	0.0	9.0	12.0	13.0	3.0	0.0

10 rows × 65 columns

```
In [7]: from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(df.drop('target',axis='columns
```

### Using RBF kernel

```
In [8]: from sklearn.svm import SVC
rbf_model = SVC(kernel='rbf')
```

```
In [9]: len(X_train)
```

Out[9]: 1437

```
In [10]: len(X_test)
```

Out[10]: 360

```
In [11]: rbf_model.fit(X_train, y_train)
```

Out[11]: SVC()

```
In [12]: rbf_model.score(X_test,y_test)
```

Out[12]: 0.9972222222222222

### Using Linear kernel

```
In [13]: linear_model = SVC(kernel='linear')  
linear_model.fit(X_train,y_train)
```

```
Out[13]: SVC(kernel='linear')
```

```
In [14]: linear_model.score(X_test,y_test)
```

```
Out[14]: 0.9861111111111112
```

Date	Author
2021-09-19	<a href="#">Ehsan Zia</a>