

# Random Forest Python Tutorial

Digits dataset from sklearn

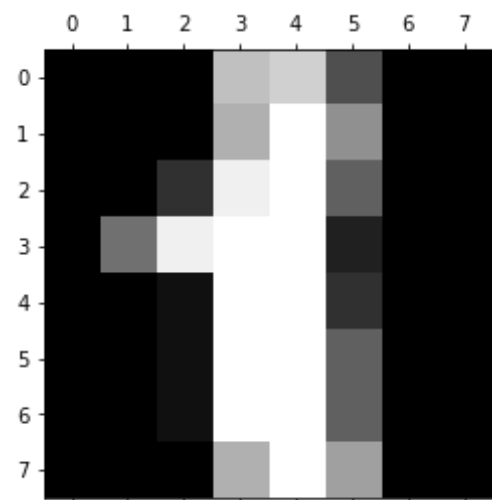
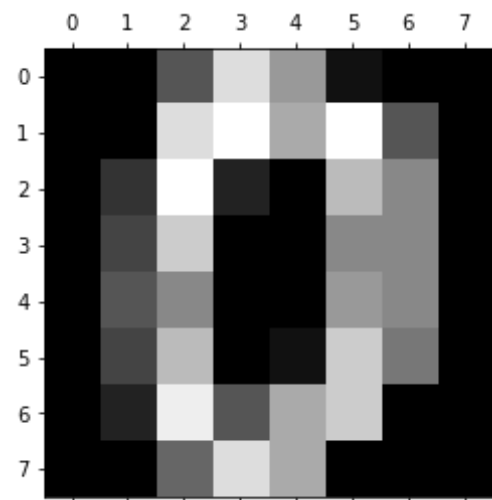
```
In [6]: import pandas as pd
        from sklearn.datasets import load_digits
        digits = load_digits()
```

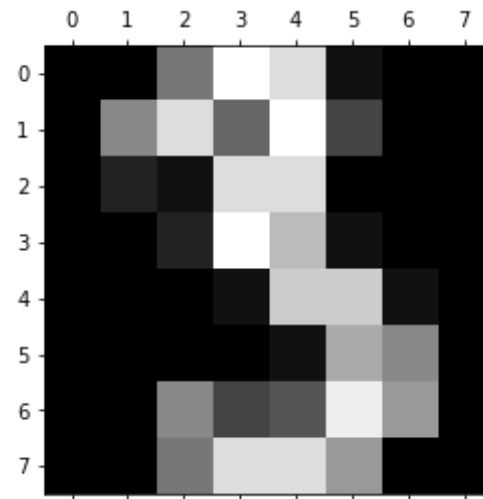
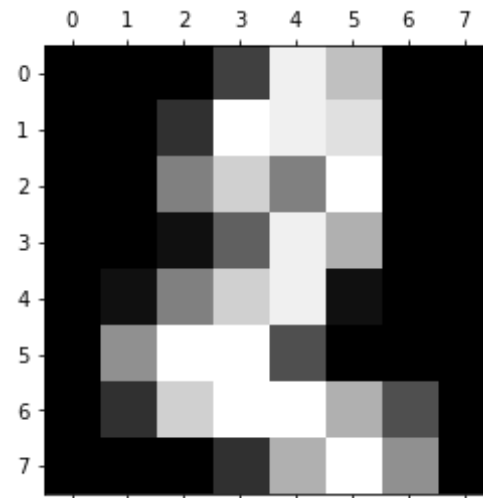
```
In [7]: dir(digits)
```

```
Out[7]: ['DESCR', 'data', 'images', 'target', 'target_names']
```

```
In [9]: %matplotlib inline
        import matplotlib.pyplot as plt
        plt.gray()
        for i in range(4):
            plt.matshow(digits.images[i])
```

<Figure size 432x288 with 0 Axes>





```
In [10]: digits.data[:5]
```

```
Out[10]: array([[ 0.,  0.,  5., 13.,  9.,  1.,  0.,  0.,  0.,  0., 13., 15.,  1
 0.,
          15.,  5.,  0.,  0.,  3., 15.,  2.,  0., 11.,  8.,  0.,  0.,
 4.,
          12.,  0.,  0.,  8.,  8.,  0.,  0.,  5.,  8.,  0.,  0.,  9.,
```

```

8.,      0.,  0.,  4., 11.,  0.,  1., 12.,  7.,  0.,  0.,  2., 14.,
5.,      10., 12.,  0.,  0.,  0.,  0.,  6., 13., 10.,  0.,  0.,  0.],
[ 0.,  0.,  0., 12., 13.,  5.,  0.,  0.,  0.,  0.,  0.,  0., 11., 1
6.,      9.,  0.,  0.,  0.,  0.,  3., 15., 16.,  6.,  0.,  0.,  0.,
7.,      15., 16., 16.,  2.,  0.,  0.,  0.,  0.,  1., 16., 16.,  3.,
0.,      0.,  0.,  0.,  1., 16., 16.,  6.,  0.,  0.,  0.,  0.,  1., 1
6.,      16.,  6.,  0.,  0.,  0.,  0.,  0., 11., 16., 10.,  0.,  0.],
[ 0.,  0.,  0.,  4., 15., 12.,  0.,  0.,  0.,  0.,  3., 16., 1
5.,      14.,  0.,  0.,  0.,  0.,  8., 13.,  8., 16.,  0.,  0.,  0.,
0.,      1.,  6., 15., 11.,  0.,  0.,  0.,  1.,  8., 13., 15.,  1.,
0.,      0.,  0.,  9., 16., 16.,  5.,  0.,  0.,  0.,  0.,  3., 13., 1
6.,      16., 11.,  5.,  0.,  0.,  0.,  0.,  3., 11., 16.,  9.,  0.],
[ 0.,  0.,  7., 15., 13.,  1.,  0.,  0.,  0.,  0.,  8., 13.,  6., 1
5.,      4.,  0.,  0.,  0.,  2.,  1., 13., 13.,  0.,  0.,  0.,  0.,
0.,      2., 15., 11.,  1.,  0.,  0.,  0.,  0.,  0.,  1., 12., 12.,
1.,      0.,  0.,  0.,  0.,  0.,  1., 10.,  8.,  0.,  0.,  0.,  8.,
4.,      5., 14.,  9.,  0.,  0.,  0.,  7., 13., 13.,  9.,  0.,  0.],
[ 0.,  0.,  0.,  1., 11.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  7.,
8.,      0.,  0.,  0.,  0.,  0.,  1., 13.,  6.,  2.,  2.,  0.,  0.,
0.,      7., 15.,  0.,  9.,  8.,  0.,  0.,  5., 16., 10.,  0., 16.,
6.,      0.,  0.,  4., 15., 16., 13., 16.,  1.,  0.,  0.,  0.,  0.,

```

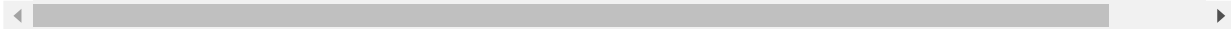
```
3.,
      15., 10., 0., 0., 0., 0., 0., 2., 16., 4., 0., 0.]])
```

```
In [11]: df = pd.DataFrame(digits.data)
df.head()
```

```
Out[11]:
```

	0	1	2	3	4	5	6	7	8	9	...	54	55	56	57	58	59	60	61
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
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
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
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
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

5 rows × 64 columns

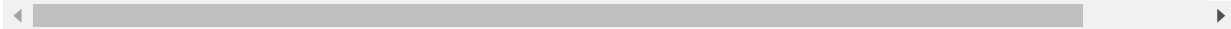


```
In [12]: df['target']=digits.target
df.head()
```

```
Out[12]:
```

	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 rows × 65 columns



```
In [32]: from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test = train_test_split(df.drop(['target'],axi
```

```
s='columns'),digits.target,test_size=0.2)
```

```
In [33]: len(X_train)
```

```
Out[33]: 1437
```

```
In [34]: len(X_test)
```

```
Out[34]: 360
```

```
In [35]: from sklearn.ensemble import RandomForestClassifier
model = RandomForestClassifier(n_estimators=40)
model.fit(X_train, y_train)
```

```
Out[35]: RandomForestClassifier(bootstrap=True, ccp_alpha=0.0, class_weight=None,
                               criterion='gini', max_depth=None, max_features='auto',
                               max_leaf_nodes=None, max_samples=None,
                               min_impurity_decrease=0.0, min_impurity_split=None,
                               min_samples_leaf=1, min_samples_split=2,
                               min_weight_fraction_leaf=0.0, n_estimators=40,
                               n_jobs=None, oob_score=False, random_state=None,
                               verbose=0, warm_start=False)
```

```
In [36]: model.score(X_test,y_test)
```

```
Out[36]: 0.9583333333333334
```

```
In [37]: y_predicted = model.predict(X_test)
```

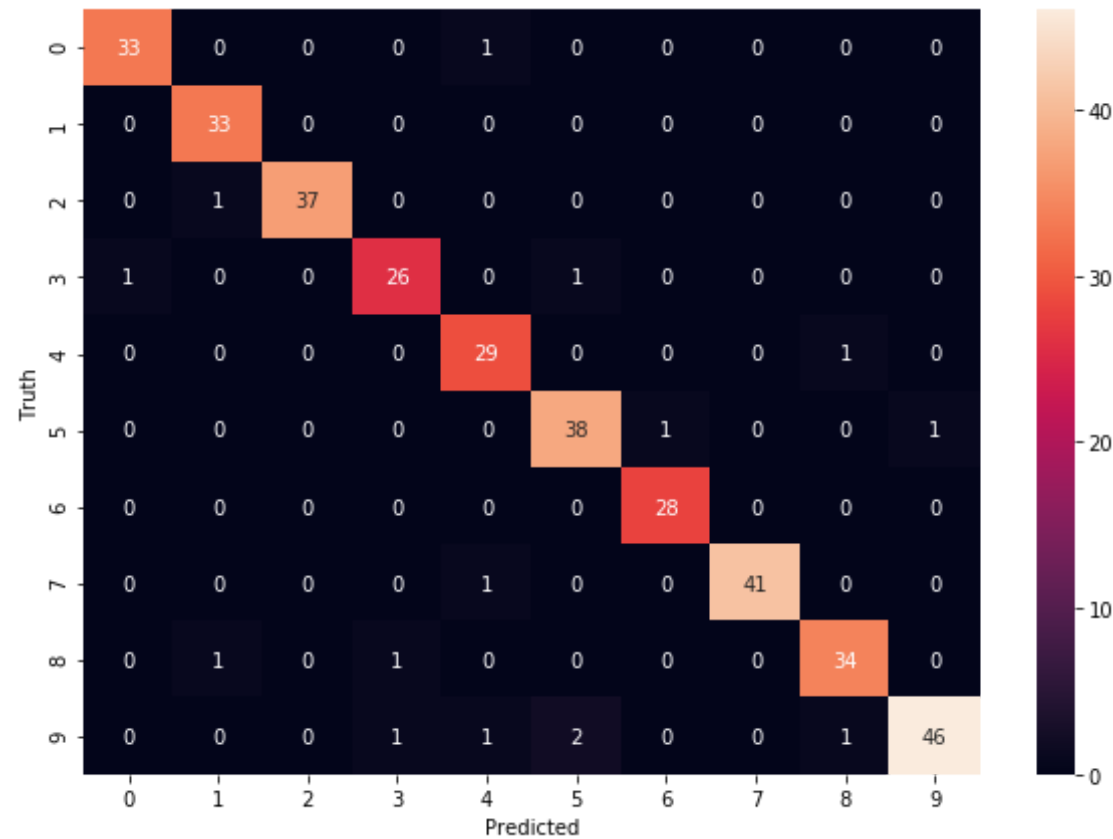
### Confusion Matrix

```
In [42]: from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_predicted)
cm
```


```
Out[42]: array([[33,  0,  0,  0,  1,  0,  0,  0,  0,  0],
               [ 0, 33,  0,  0,  0,  0,  0,  0,  0,  0],
               [ 0,  1, 37,  0,  0,  0,  0,  0,  0,  0],
               [ 1,  0,  0, 26,  0,  1,  0,  0,  0,  0],
               [ 0,  0,  0,  0, 29,  0,  0,  0,  1,  0],
               [ 0,  0,  0,  0,  0, 38,  1,  0,  0,  1],
               [ 0,  0,  0,  0,  0,  0, 28,  0,  0,  0],
               [ 0,  0,  0,  0,  1,  0,  0, 41,  0,  0],
               [ 0,  1,  0,  1,  0,  0,  0,  0, 34,  0],
               [ 0,  0,  0,  1,  1,  2,  0,  0,  1, 46]], dtype=int64)
```

```
In [41]: %matplotlib inline
import matplotlib.pyplot as plt
import seaborn as sn
plt.figure(figsize=(10,7))
sn.heatmap(cm, annot=True)
plt.xlabel('Predicted')
plt.ylabel('Truth')
```

```
Out[41]: Text(69.0, 0.5, 'Truth')
```



### Exercise

 Use famous iris flower dataset from sklearn.datasets to predict flower species using random forest classifier.

1. Measure prediction score using default `n_estimators` (10)
2. Now fine tune your model by changing number of trees in your classifier and tell me what best score you can get using how many trees