Experiment 9: Implement Random Forest Classifier by using Digit Dataset

In [1]:

```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.datasets import load_digits
digits = load_digits()
```

In [3]:

```
dir(digits)
```

Out[3]:

```
['DESCR', 'data', 'feature_names', 'frame', 'images', 'target', 'target_nam
es']
```

In [6]:

```
# digits.data
digits.target_names
```

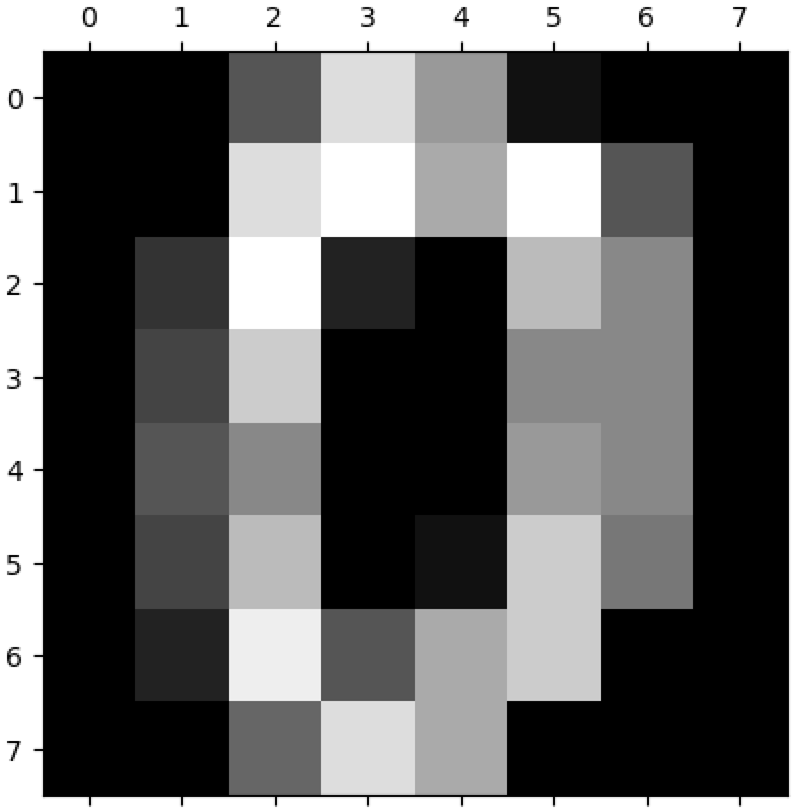
Out[6]:

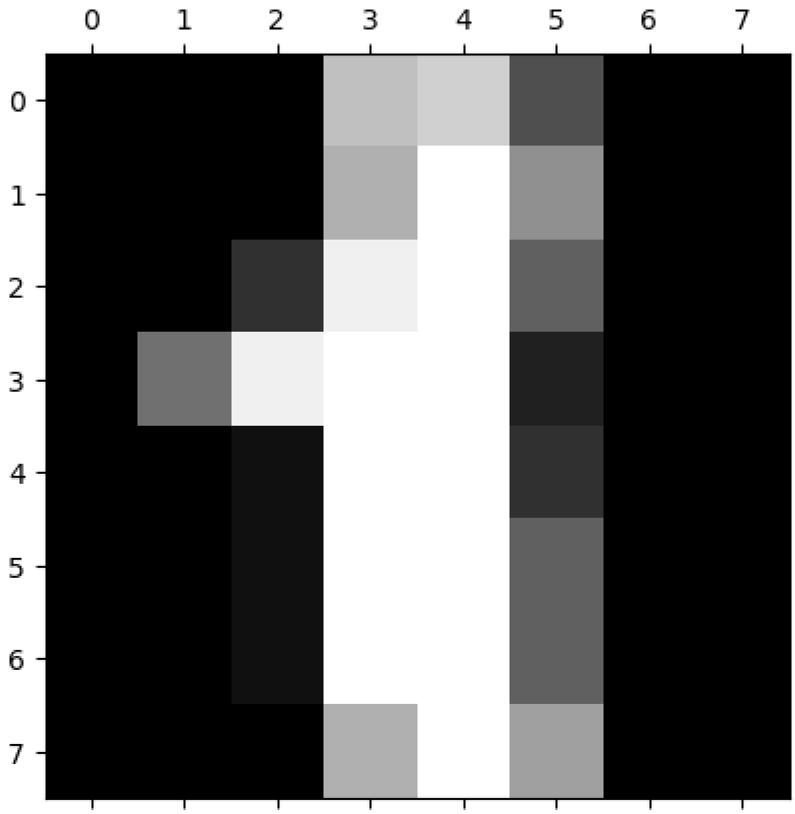
```
array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
```

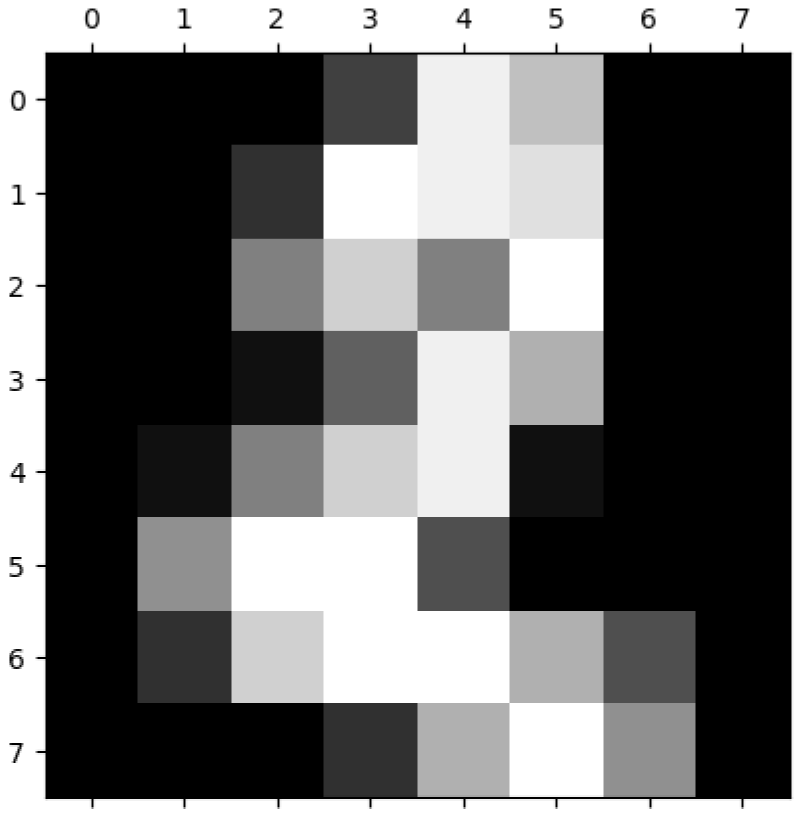
In [8]:

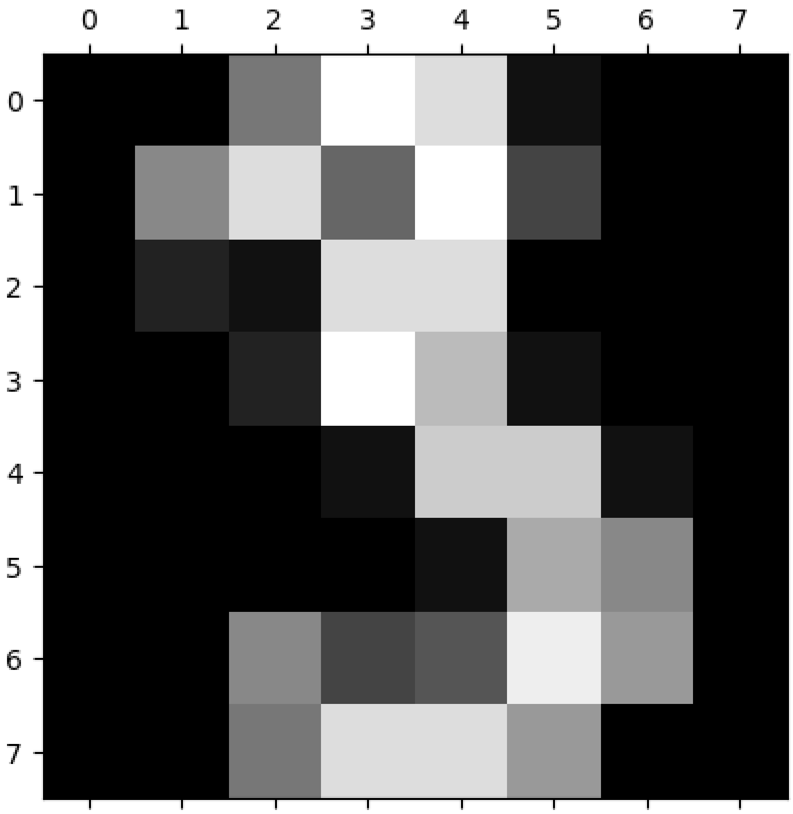
```
%matplotlib inline
plt.gray()
for i in range(4):
    plt.matshow(digits.images[i])
```

```
<Figure size 640x480 with 0 Axes>
```




0 1 2 3 4 5 6 7
0 1 2 3 4 5 6 7


0 1 2 3 4 5 6 7
0 1 2 3 4 5 6 7



In [12]:
```
df = pd.DataFrame(digits.data)
df.head()
```

Out[12]:

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

5 rows × 64 columns

In [14]:
```
df['target'] = digits.target
df.head()
```

Out[14]:

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

5 rows × 65 columns

In [16]:
```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(df.drop(['target'], axis
```

In [18]:
```
len(x_train)
# len(x_test)
```

Out[18]:
```
1437
```

In [20]:
```
from sklearn.ensemble import RandomForestClassifier
model = RandomForestClassifier(n_estimators=200, criterion='entropy')
model.fit(x_train, y_train)
```

Out[20]:

▾ RandomForestClassifier

```
RandomForestClassifier(criterion='entropy', n_estimators=200)
```

In [24]:
```
model.score(x_test, y_test)
```

Out[24]:
```
0.9805555555555555
```

In [26]:
```
y_pred = model.predict(x_test)
```

In [28]:
```
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
```

In [30]:
```
cm
```

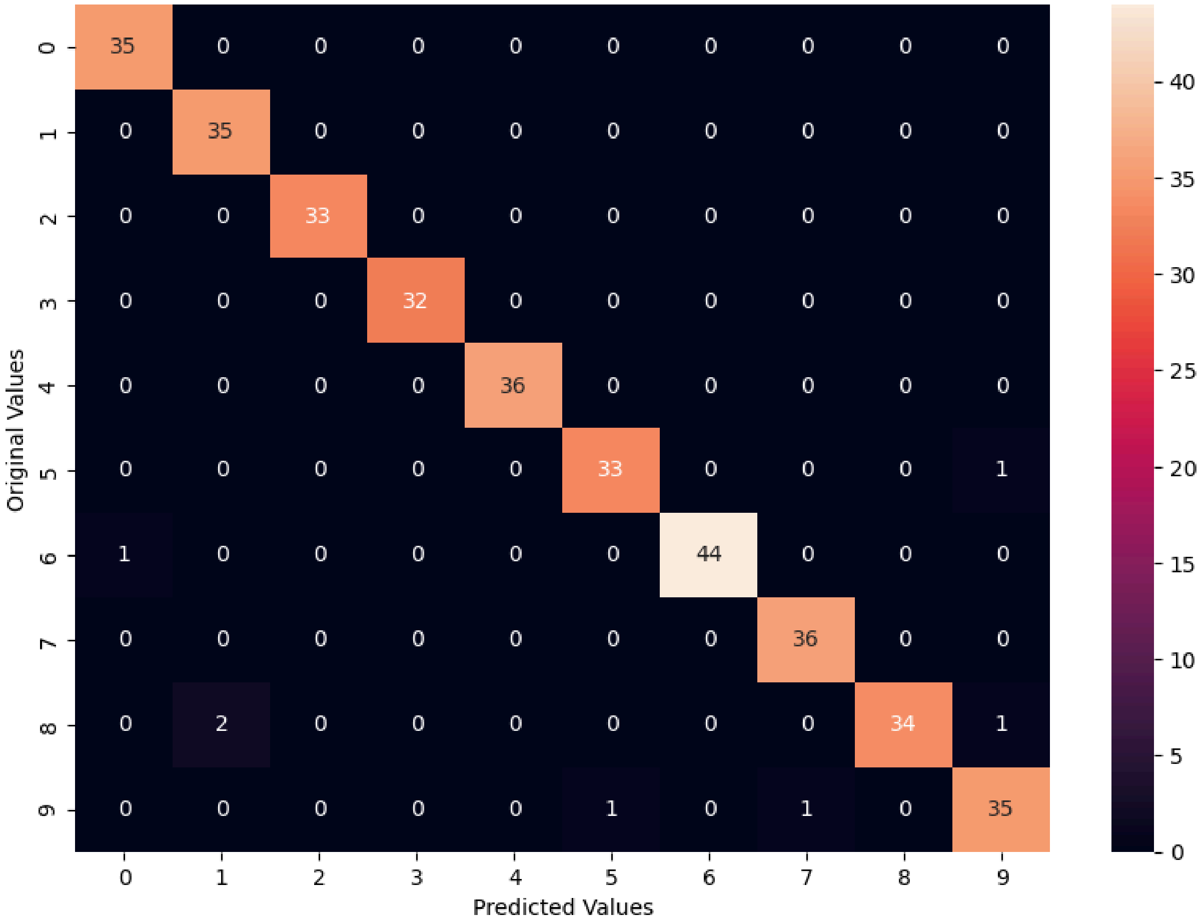
Out[30]:
```
array([[35,  0,  0,  0,  0,  0,  0,  0,  0,  0],
       [ 0, 35,  0,  0,  0,  0,  0,  0,  0,  0],
       [ 0,  0, 33,  0,  0,  0,  0,  0,  0,  0],
       [ 0,  0,  0, 32,  0,  0,  0,  0,  0,  0],
       [ 0,  0,  0,  0, 36,  0,  0,  0,  0,  0],
       [ 0,  0,  0,  0,  0, 33,  0,  0,  0,  1],
       [ 1,  0,  0,  0,  0,  0, 44,  0,  0,  0],
       [ 0,  0,  0,  0,  0,  0,  0, 36,  0,  0],
       [ 0,  2,  0,  0,  0,  0,  0,  0, 34,  1],
       [ 0,  0,  0,  0,  0,  1,  0,  1,  0, 35]], dtype=int64)
```

In [38]:

```
import seaborn as sns
plt.figure(figsize=(10,7))
sns.heatmap(cm, annot=True)
plt.xlabel('Predicted Values')
plt.ylabel('Original Values')
```

Out[38]:

```
Text(95.72222222222221, 0.5, 'Original Values')
```


Original Values
Predicted Values
0: 35 0 0 0 0 0 0 0 0 0
1: 0 35 0 0 0 0 0 0 0 0
2: 0 0 33 0 0 0 0 0 0 0
3: 0 0 0 32 0 0 0 0 0 0
4: 0 0 0 0 36 0 0 0 0 0
5: 0 0 0 0 0 33 0 0 0 1
6: 1 0 0 0 0 0 44 0 0 0
7: 0 0 0 0 0 0 0 36 0 0
8: 0 2 0 0 0 0 0 0 34 1
9: 0 0 0 0 0 1 0 1 0 35