

Name : Tavhare Ruchita Sharad

class : BE AI&DS

Roll No.: 61

Subject: Computer Laboratory-I(Machine Learning)

Title: Use different voting mechanism and Apply AdaBoost (AdaptiveBoosting), Gradient Tree Boosting (GBM), XGBoost classification on Iris dataset and compare the performance of three models using different evaluation measures.

```
In [4]: import pandas as pd
from sklearn.datasets import load_digits
digits = load_digits()
dir(digits)
```

```
Out[4]: ['DESCR', 'data', 'feature_names', 'frame', 'images', 'target', 'target_names']
```

```
In [6]: %matplotlib inline
import matplotlib.pyplot as plt
plt.gray()
for i in range(4):
    plt.matshow(digits.images[i])
```

```
In [7]: df = pd.DataFrame(digits.data)
df.head()
```

```
Out[7]:   0   1   2   3   4   5   6   7   8   9   ...   54   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  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  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  ...  5.0  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  ...  9.0  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  0.0  2.0
```

5 rows × 64 columns

```
In [10]: df['target'] = digits.target
df[0:12]
```

Out[10]:

	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	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
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
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
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
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
10	0.0	0.0	1.0	9.0	15.0	11.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	1.0	10.0
11	0.0	0.0	0.0	0.0	14.0	13.0	1.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	1.0

12 rows × 65 columns

In [12]:

```
X = df.drop('target', axis='columns')
y = df.target
```

In [14]:

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.2)
```

In [16]:

```
from sklearn.ensemble import RandomForestClassifier
model = RandomForestClassifier(n_estimators=20)
model.fit(X_train, y_train)
RandomForestClassifier(n_estimators=20)
model.score(X_test, y_test)
```

Out[16]: 0.95

In [18]:

```
y_predicted = model.predict(X_test)
```

In [20]:

```
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_predicted)
cm
```

```
Out[20]: array([[36,  0,  0,  0,  1,  0,  0,  0,  0,  0,  0],  
                 [ 0, 36,  0,  0,  0,  0,  0,  0,  0,  0,  0],  
                 [ 0,  0, 40,  0,  0,  0,  0,  0,  1,  0],  
                 [ 0,  0,  0, 33,  0,  0,  0,  0,  3,  1],  
                 [ 0,  0,  0,  0, 31,  0,  0,  1,  0,  0],  
                 [ 0,  0,  0,  1,  0, 40,  0,  0,  0,  2],  
                 [ 0,  0,  0,  0,  2,  0, 31,  0,  0,  0],  
                 [ 0,  0,  0,  0,  0,  0,  0, 25,  0,  0],  
                 [ 0,  1,  0,  0,  0,  0,  1,  0, 37,  1],  
                 [ 0,  1,  0,  2,  0,  0,  0,  0,  0, 33]], dtype=int64)
```

```
In [22]: %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[22]: Text(95.72222222222221, 0.5, 'Truth')
```