digits data classificatioin using svc

importing libraries

In [2]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
from sklearn.datasets import load_digits
datset = load_digits()
```

In [3]:
```
dd = datset.data
dd
```

Out[3]:
```
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 [4]:
```
datset.data[0]
```

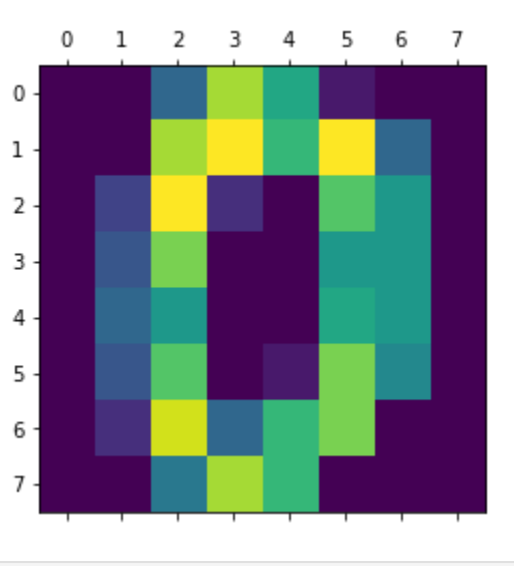
Out[4]:
```
array([ 0.,  0.,  5., 13.,  9.,  1.,  0.,  0.,  0.,  0., 13., 15., 10.,
       15.,  5.,  0.,  0.,  3., 15.,  2.,  0., 11.,  8.,  0.,  0.,  4.,
       12.,  0.,  0.,  8.,  8.,  0.,  0.,  5.,  8.,  0.,  0.,  9.,  8.,
        0.,  0.,  4., 11.,  0.,  1., 12.,  7.,  0.,  0.,  2., 14.,  5.,
       10., 12.,  0.,  0.,  0.,  0.,  6., 13., 10.,  0.,  0.,  0.])
```

In [5]:
```
datset.target
```

Out[5]:
```
array([0, 1, 2, ..., 8, 9, 8])
```

In [6]:
```
plt.cool
plt.matshow(datset.images[0])
```

Out[6]:
```
<matplotlib.image.AxesImage at 0x1a36907ae80>
```



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

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

In [8]:
```
df = pd.DataFrame(dd, columns=[datset.feature_names])
```

In [9]:
```
df['targt'] = datset.target
```

In [10]:
```
df
```

Out[10]:

	pixel_0_0	pixel_0_1	pixel_0_2	pixel_0_3	pixel_0_4	pixel_0_5	pixel_0_6	pixel_0_7	pixel_1_0	pixel_1_1	...	pixel_6_7	pixel_7_0	pixel_7_1	pixel_7_2	pixel_7_3	pixel_7_4	pixel_7_5	pixel_7_6	pixel_7_7	targt
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	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	0.0	1
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	0.0	2
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	0.0	3
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	0.0	4
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1792	0.0	0.0	4.0	10.0	13.0	6.0	0.0	0.0	0.0	1.0	...	0.0	0.0	0.0	2.0	14.0	15.0	9.0	0.0	0.0	9
1793	0.0	0.0	6.0	16.0	13.0	11.0	1.0	0.0	0.0	0.0	...	0.0	0.0	0.0	6.0	16.0	14.0	6.0	0.0	0.0	0
1794	0.0	0.0	1.0	11.0	15.0	1.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	2.0	9.0	13.0	6.0	0.0	0.0	8
1795	0.0	0.0	2.0	10.0	7.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	5.0	12.0	16.0	12.0	0.0	0.0	9
1796	0.0	0.0	10.0	14.0	8.0	1.0	0.0	0.0	0.0	2.0	...	0.0	0.0	1.0	8.0	12.0	14.0	12.0	1.0	0.0	8

1797 rows × 65 columns

In [11]:
```
X = df.drop(['targt'],axis=1)
X
```

Out[11]:

	pixel_0_0	pixel_0_1	pixel_0_2	pixel_0_3	pixel_0_4	pixel_0_5	pixel_0_6	pixel_0_7	pixel_1_0	pixel_1_1	...	pixel_6_6	pixel_6_7	pixel_7_0	pixel_7_1	pixel_7_2	pixel_7_3	pixel_7_4	pixel_7_5	pixel_7_6	pixel_7_7
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	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	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	0.0
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	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	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1792	0.0	0.0	4.0	10.0	13.0	6.0	0.0	0.0	0.0	1.0	...	4.0	0.0	0.0	0.0	2.0	14.0	15.0	9.0	0.0	0.0
1793	0.0	0.0	6.0	16.0	13.0	11.0	1.0	0.0	0.0	0.0	...	1.0	0.0	0.0	0.0	6.0	16.0	14.0	6.0	0.0	0.0
1794	0.0	0.0	1.0	11.0	15.0	1.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	2.0	9.0	13.0	6.0	0.0	0.0
1795	0.0	0.0	2.0	10.0	7.0	0.0	0.0	0.0	0.0	0.0	...	2.0	0.0	0.0	0.0	5.0	12.0	16.0	12.0	0.0	0.0
1796	0.0	0.0	10.0	14.0	8.0	1.0	0.0	0.0	0.0	2.0	...	8.0	0.0	0.0	1.0	8.0	12.0	14.0	12.0	1.0	0.0

1797 rows × 64 columns

In [12]:
```
y = df.targt
y
```

Out[12]:

	targt
0	0
1	1
2	2
3	3
4	4
...	...
1792	9
1793	0
1794	8
1795	9
1796	8

1797 rows × 1 columns

splitting the Data for training and testing

In [13]:
```
from sklearn.model_selection import train_test_split
```

In [14]:
```
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.2,random_state=38)
```

In [15]:
```
X_train
```

Out[15]:

	pixel_0_0	pixel_0_1	pixel_0_2	pixel_0_3	pixel_0_4	pixel_0_5	pixel_0_6	pixel_0_7	pixel_1_0	pixel_1_1	...	pixel_6_6	pixel_6_7	pixel_7_0	pixel_7_1	pixel_7_2	pixel_7_3	pixel_7_4	pixel_7_5	pixel_7_6	pixel_7_7
1540	0.0	0.0	6.0	16.0	16.0	11.0	0.0	0.0	0.0	1.0	...	0.0	0.0	0.0	0.0	9.0	16.0	10.0	1.0	0.0	0.0
281	0.0	0.0	11.0	7.0	12.0	15.0	1.0	0.0	0.0	1.0	...	0.0	0.0	0.0	0.0	12.0	16.0	10.0	1.0	0.0	0.0
507	0.0	0.0	0.0	0.0	13.0	3.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	15.0	3.0	0.0	0.0
1228	0.0	0.0	10.0	12.0	16.0	16.0	8.0	0.0	0.0	4.0	...	0.0	0.0	0.0	0.0	13.0	16.0	11.0	0.0	0.0	0.0
1619	0.0	0.0	0.0	10.0	16.0	7.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	1.0	11.0	15.0	6.0	0.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
900	0.0	0.0	0.0	4.0	15.0	6.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	6.0	16.0	2.0	0.0	0.0
316	0.0	0.0	7.0	14.0	16.0	11.0	0.0	0.0	0.0	2.0	...	2.0	0.0	0.0	0.0	6.0	16.0	12.0	5.0	0.0	0.0
1491	0.0	0.0	2.0	9.0	13.0	12.0	2.0	0.0	0.0	1.0	...	2.0	0.0	0.0	0.0	0.0	13.0	15.0	6.0	0.0	0.0
53	0.0	0.0	4.0	8.0	16.0	5.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	6.0	16.0	12.0	1.0	0.0	0.0
1441	0.0	0.0	6.0	14.0	1.0	0.0	0.0	0.0	0.0	0.0	...	9.0	0.0	0.0	0.0	5.0	13.0	16.0	12.0	1.0	0.0

1437 rows × 64 columns

In [16]:
```
y_train
```

Out[16]:

	targt
1540	9
281	5
507	4
1228	5
1619	8
...	...
900	4
316	3
1491	8
53	8
1441	6

1437 rows × 1 columns

In [17]:
```
X_test
```

Out[17]:

	pixel_0_0	pixel_0_1	pixel_0_2	pixel_0_3	pixel_0_4	pixel_0_5	pixel_0_6	pixel_0_7	pixel_1_0	pixel_1_1	...	pixel_6_6	pixel_6_7	pixel_7_0	pixel_7_1	pixel_7_2	pixel_7_3	pixel_7_4	pixel_7_5	pixel_7_6	pixel_7_7
1196	0.0	1.0	7.0	14.0	10.0	0.0	0.0	0.0	0.0	10.0	...	5.0	0.0	0.0	0.0	4.0	14.0	14.0	12.0	2.0	0.0
214	0.0	0.0	10.0	16.0	9.0	0.0	0.0	0.0	0.0	4.0	...	0.0	0.0	0.0	0.0	11.0	16.0	16.0	16.0	2.0	0.0
979	0.0	0.0	13.0	16.0	11.0	0.0	0.0	0.0	0.0	2.0	...	11.0	5.0	0.0	0.0	9.0	12.0	13.0	16.0	16.0	11.0
1080	0.0	0.0	9.0	16.0	6.0	0.0	0.0	0.0	0.0	3.0	...	0.0	0.0	0.0	0.0	12.0	14.0	3.0	0.0	0.0	0.0
615	0.0	0.0	5.0	16.0	14.0	8.0	0.0	0.0	0.0	0.0	...	3.0	0.0	0.0	0.0	6.0	16.0	16.0	16.0	3.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
711	0.0	0.0	2.0	14.0	1.0	0.0	0.0	0.0	0.0	0.0	...	13.0	0.0	0.0	0.0	3.0	11.0	15.0	14.0	1.0	0.0
262	0.0	0.0	3.0	13.0	6.0	0.0	0.0	0.0	0.0	0.0	...	11.0	0.0	0.0	0.0	3.0	13.0	15.0	8.0	0.0	0.0
950	0.0	1.0	10.0	16.0	15.0	1.0	0.0	0.0	0.0	3.0	...	6.0	0.0	0.0	0.0	12.0	16.0	16.0	7.0	0.0	0.0
1487	0.0	0.0	7.0	16.0	15.0	1.0	0.0	0.0	0.0	5.0	...	3.0	0.0	0.0	0.0	8.0	16.0	13.0	5.0	0.0	0.0
1092	0.0	0.0	1.0	14.0	6.0	0.0	0.0	0.0	0.0	0.0	...	16.0	3.0	0.0	0.0	1.0	12.0	13.0	16.0	9.0	1.0

360 rows × 64 columns

In [18]:
```
y_test
```

Out[18]:

	targt
1196	9
214	2
979	2
1080	8
615	1
...	...
711	6
262	6
950	3
1487	0
1092	6

360 rows × 1 columns

importing the required model

In [19]:
```
from sklearn.svm import SVC
```

In [20]:
```
svc = SVC()
svc.fit(X_train,y_train)
```

```
C:\Users\91910\anaconda3\lib\site-packages\sklearn\utils\validation.py:1688: FutureWarning: Feature names only support names that are all strings. Got feature names with dtypes: ['tuple']. An error will be raised in 1.2.
  warnings.warn(
C:\Users\91910\anaconda3\lib\site-packages\sklearn\utils\validation.py:993: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().
  y = column_or_1d(y, warn=True)
```

Out[20]:
```
SVC()
```

In [32]:
```
y_pred = svc.predict(X_test)
y_pred
```

```
C:\Users\91910\anaconda3\lib\site-packages\sklearn\utils\validation.py:1688: FutureWarning: Feature names only support names that are all strings. Got feature names with dtypes: ['tuple']. An error will be raised in 1.2.
  warnings.warn(
```

Out[32]:
```
array([9, 2, 2, 8, 1, 8, 4, 8, 9, 5, 0, 0, 6, 6, 9, 7, 5, 9, 3, 5, 6, 9,
       1, 2, 8, 4, 9, 4, 0, 6, 7, 1, 0, 6, 1, 5, 1, 5, 4, 6, 1, 9, 9, 6,
       8, 4, 5, 5, 0, 2, 4, 1, 7, 4, 2, 7, 2, 2, 1, 5, 4, 1, 8, 0, 2, 7,
       7, 3, 3, 9, 0, 1, 5, 0, 4, 1, 9, 9, 5, 3, 8, 8, 4, 4, 2, 7, 5, 1,
       7, 6, 6, 3, 1, 8, 9, 0, 0, 9, 3, 0, 3, 0, 4, 8, 4, 4, 4, 0, 5, 6,
       5, 6, 6, 0, 1, 3, 0, 2, 4, 4, 2, 1, 1, 4, 8, 4, 6, 8, 0, 4, 1, 3,
       3, 1, 6, 8, 3, 8, 9, 0, 4, 1, 8, 6, 3, 9, 3, 9, 2, 0, 2, 4, 8, 1,
       6, 9, 4, 3, 5, 6, 0, 3, 6, 4, 6, 7, 7, 4, 9, 8, 0, 8, 6, 0, 4, 0,
       0, 0, 3, 0, 7, 7, 2, 2, 5, 2, 8, 6, 6, 5, 4, 0, 0, 4, 3, 2, 5, 3,
       9, 5, 7, 1, 9, 2, 8, 6, 3, 2, 8, 1, 8, 6, 0, 5, 5, 9, 7, 6, 3, 7,
       8, 1, 1, 2, 5, 7, 2, 1, 7, 7, 4, 1, 3, 1, 3, 7, 3, 7, 1, 3, 4, 8,
       0, 7, 5, 3, 5, 5, 1, 1, 2, 4, 9, 3, 3, 6, 6, 5, 5, 1, 4, 8, 8, 2,
       4, 5, 0, 7, 8, 1, 0, 8, 4, 7, 7, 4, 6, 9, 9, 4, 1, 1, 7, 3, 0, 3,
       9, 2, 4, 6, 4, 8, 5, 0, 9, 2, 8, 8, 1, 7, 1, 6, 6, 3, 0, 6, 8, 4,
       5, 5, 8, 6, 4, 6, 2, 9, 6, 4, 6, 6, 3, 7, 1, 2, 6, 2, 3, 7, 1, 8,
       3, 1, 2, 6, 3, 0, 9, 0, 2, 7, 7, 4, 0, 8, 7, 3, 7, 7, 4, 9, 4, 3,
       1, 4, 5, 6, 6, 3, 0, 6])
```

In [22]:
```
datset.target[67]
```

Out[22]:
```
6
```

In [23]:
```
svc.predict([datset.data[67]])
```

Out[23]:
```
array([6])
```

In [24]:
```
svc.score(X_test,y_test)
```

```
C:\Users\91910\anaconda3\lib\site-packages\sklearn\utils\validation.py:1688: FutureWarning: Feature names only support names that are all strings. Got feature names with dtypes: ['tuple']. An error will be raised in 1.2.
  warnings.warn(
```

Out[24]:
```
0.9888888888888889
```

In [ ]:

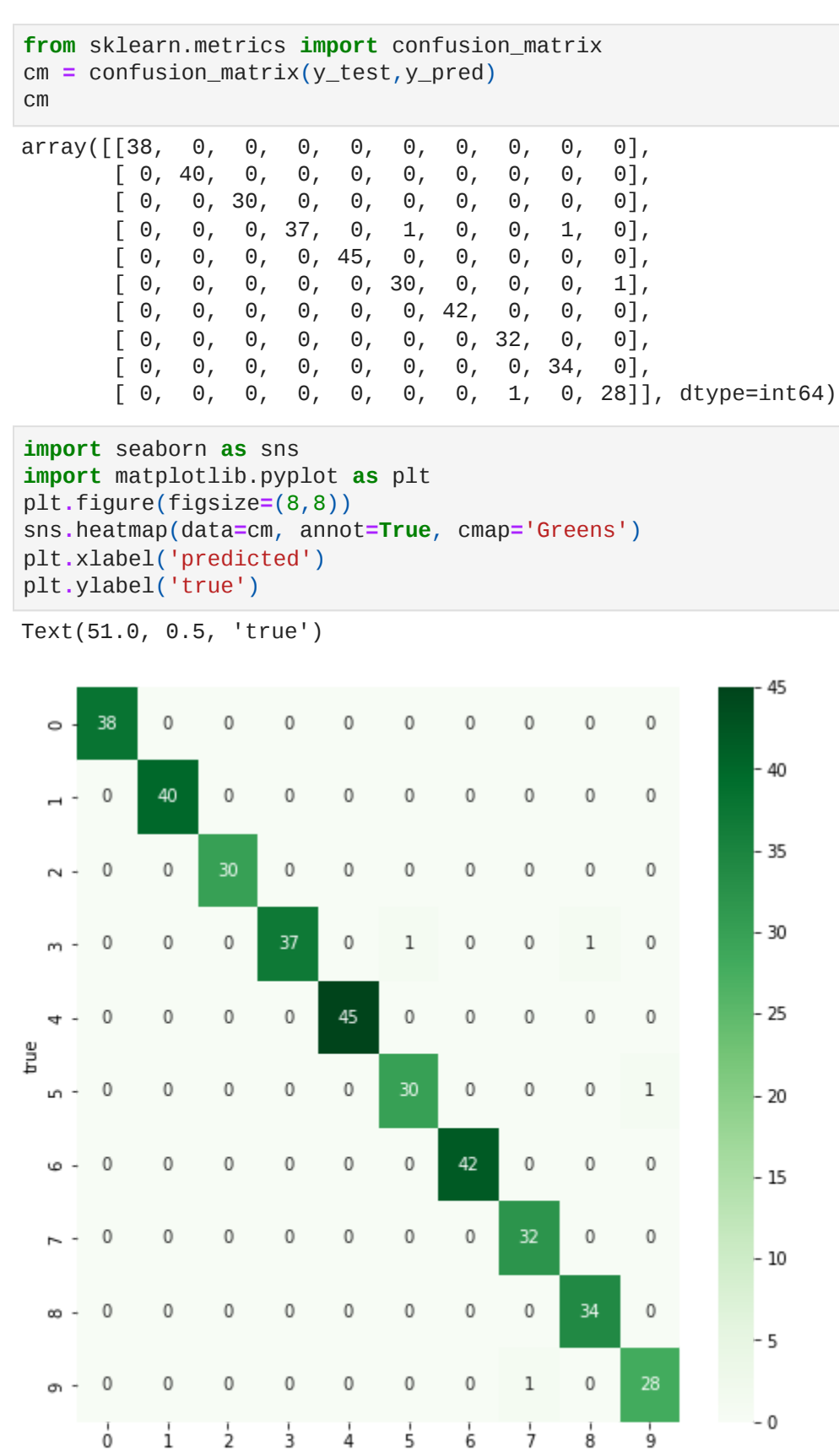
confusion matrix

In [33]:
```
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test,y_pred)
cm
```

Out[33]:
```
array([[38,  0,  0,  0,  0,  0,  0,  0,  0,  0],
       [ 0, 40,  0,  0,  0,  0,  0,  0,  0,  0],
       [ 0,  0, 30,  0,  0,  0,  0,  0,  0,  0],
       [ 0,  0,  0, 37,  0,  1,  0,  0,  1,  0],
       [ 0,  0,  0,  0, 45,  0,  0,  0,  0,  0],
       [ 0,  0,  0,  0,  0, 30,  0,  0,  0,  1],
       [ 0,  0,  0,  0,  0,  0, 42,  0,  0,  0],
       [ 0,  0,  0,  0,  0,  0,  0, 32,  0,  0],
       [ 0,  0,  0,  0,  0,  0,  0,  0, 34,  0],
       [ 0,  0,  0,  0,  0,  0,  0,  1,  0, 28]], dtype=int64)
```

In [37]:
```
import seaborn as sns
import matplotlib.pyplot as plt
plt.figure(figsize=(8,8))
sns.heatmap(data=cm, annot=True, cmap='Greens')
plt.xlabel('predicted')
plt.ylabel('true')
```

Out[37]:
```
Text(51.0, 0.5, 'true')
```



using parameters

regularization(c)

In [25]:
```
model_C = SVC(C=1)
model_C.fit(X_train, y_train)
model_C.score(X_test, y_test)
```

```
C:\Users\91910\anaconda3\lib\site-packages\sklearn\utils\validation.py:1688: FutureWarning: Feature names only support names that are all strings. Got feature names with dtypes: ['tuple']. An error will be raised in 1.2.
  warnings.warn(
C:\Users\91910\anaconda3\lib\site-packages\sklearn\utils\validation.py:993: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().
  y = column_or_1d(y, warn=True)
C:\Users\91910\anaconda3\lib\site-packages\sklearn\utils\validation.py:1688: FutureWarning: Feature names only support names that are all strings. Got feature names with dtypes: ['tuple']. An error will be raised in 1.2.
  warnings.warn(
```

Out[25]:
```
0.9888888888888889
```

In [26]:
```
model_C = SVC(C=10)
model_C.fit(X_train, y_train)
model_C.score(X_test, y_test)
```

```
C:\Users\91910\anaconda3\lib\site-packages\sklearn\utils\validation.py:1688: FutureWarning: Feature names only support names that are all strings. Got feature names with dtypes: ['tuple']. An error will be raised in 1.2.
  warnings.warn(
C:\Users\91910\anaconda3\lib\site-packages\sklearn\utils\validation.py:993: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().
  y = column_or_1d(y, warn=True)
C:\Users\91910\anaconda3\lib\site-packages\sklearn\utils\validation.py:1688: FutureWarning: Feature names only support names that are all strings. Got feature names with dtypes: ['tuple']. An error will be raised in 1.2.
  warnings.warn(
```

Out[26]:
```
0.9944444444444445
```

gamma

In [27]:
```
model_g = SVC(gamma=10)
model_g.fit(X_train, y_train)
model_g.score(X_test, y_test)
```

```
C:\Users\91910\anaconda3\lib\site-packages\sklearn\utils\validation.py:1688: FutureWarning: Feature names only support names that are all strings. Got feature names with dtypes: ['tuple']. An error will be raised in 1.2.
  warnings.warn(
C:\Users\91910\anaconda3\lib\site-packages\sklearn\utils\validation.py:993: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().
  y = column_or_1d(y, warn=True)
C:\Users\91910\anaconda3\lib\site-packages\sklearn\utils\validation.py:1688: FutureWarning: Feature names only support names that are all strings. Got feature names with dtypes: ['tuple']. An error will be raised in 1.2.
  warnings.warn(
```

Out[27]:
```
0.08055555555555556
```

In [28]:
```
model_g = SVC(gamma=1)
model_g.fit(X_train, y_train)
model_g.score(X_test, y_test)
```

```
C:\Users\91910\anaconda3\lib\site-packages\sklearn\utils\validation.py:1688: FutureWarning: Feature names only support names that are all strings. Got feature names with dtypes: ['tuple']. An error will be raised in 1.2.
  warnings.warn(
C:\Users\91910\anaconda3\lib\site-packages\sklearn\utils\validation.py:993: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().
  y = column_or_1d(y, warn=True)
C:\Users\91910\anaconda3\lib\site-packages\sklearn\utils\validation.py:1688: FutureWarning: Feature names only support names that are all strings. Got feature names with dtypes: ['tuple']. An error will be raised in 1.2.
  warnings.warn(
```

Out[28]:
```
0.16111111111111112
```

using rbf kernal

In [29]:
```
rbf_model = SVC(kernel='rbf')
rbf_model.fit(X_train,y_train)
rbf_model.score(X_test,y_test)
```

```
C:\Users\91910\anaconda3\lib\site-packages\sklearn\utils\validation.py:1688: FutureWarning: Feature names only support names that are all strings. Got feature names with dtypes: ['tuple']. An error will be raised in 1.2.
  warnings.warn(
C:\Users\91910\anaconda3\lib\site-packages\sklearn\utils\validation.py:993: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().
  y = column_or_1d(y, warn=True)
C:\Users\91910\anaconda3\lib\site-packages\sklearn\utils\validation.py:1688: FutureWarning: Feature names only support names that are all strings. Got feature names with dtypes: ['tuple']. An error will be raised in 1.2.
  warnings.warn(
```

Out[29]:
```
0.9888888888888889
```

In [30]:
```
rbf_model = SVC(kernel='linear')
rbf_model.fit(X_train,y_train)
rbf_model.score(X_test,y_test)
```

```
C:\Users\91910\anaconda3\lib\site-packages\sklearn\utils\validation.py:1688: FutureWarning: Feature names only support names that are all strings. Got feature names with dtypes: ['tuple']. An error will be raised in 1.2.
  warnings.warn(
C:\Users\91910\anaconda3\lib\site-packages\sklearn\utils\validation.py:993: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().
  y = column_or_1d(y, warn=True)
C:\Users\91910\anaconda3\lib\site-packages\sklearn\utils\validation.py:1688: FutureWarning: Feature names only support names that are all strings. Got feature names with dtypes: ['tuple']. An error will be raised in 1.2.
  warnings.warn(
```

Out[30]:
```
0.9805555555555555
```

In [ ]:

In [ ]: