

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
from sklearn.datasets import load_digits
```

```
digit=load_digits()
```

```
digit
```

```
{'data': 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.]])
, 'target': array([0, 1, 2, ..., 8, 9, 8])
, 'frame': None
, 'feature_names': ['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_1_2',
 'pixel_1_3',
 'pixel_1_4',
 'pixel_1_5',
 'pixel_1_6',
 'pixel_1_7',
 'pixel_2_0',
 'pixel_2_1',
 'pixel_2_2',
 'pixel_2_3',
 'pixel_2_4',
 'pixel_2_5',
 'pixel_2_6',
 'pixel_2_7',
 'pixel_3_0',
 'pixel_3_1',
 'pixel_3_2',
 'pixel_3_3',
 'pixel_3_4',
 'pixel_3_5',
 'pixel_3_6',
 'pixel_3_7',
 'pixel_4_0',
 'pixel_4_1',
 'pixel_4_2',
 'pixel_4_3',
 'pixel_4_4',
 'pixel_4_5',
 'pixel_4_6',
 'pixel_4_7',
 'pixel_5_0',
 'pixel_5_1',
 'pixel_5_2',
 'pixel_5_3',
 'pixel_5_4',
 'pixel_5_5',
 'pixel_5_6',
 'pixel_5_7',
 'pixel_6_0',
```

```
plt.imshow(digit.images[1], cmap='gray')
```

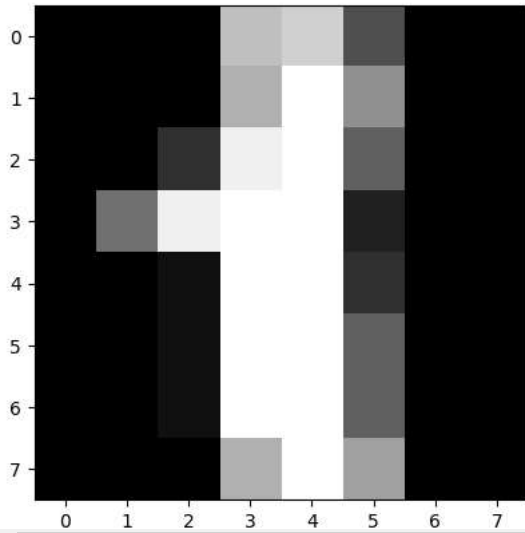


McAfee | WebAdvisor




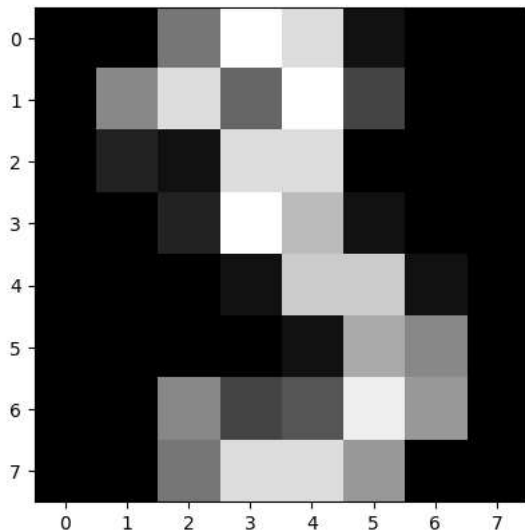
Your download's being scanned.
We'll let you know if there's an issue.

 <matplotlib.image.AxesImage at 0x7e3ff3c40070>




```
plt.imshow(digit.images[3], cmap='gray')
```

 <matplotlib.image.AxesImage at 0x7e3ff3daf5e0>



```
data1=pd.DataFrame(digit.data)
data1
```



	0	1	2	3	4	5	6	7	8	9	...	54	55	56	57	58	59	60	61	62	63
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

1797 rows x 64 columns

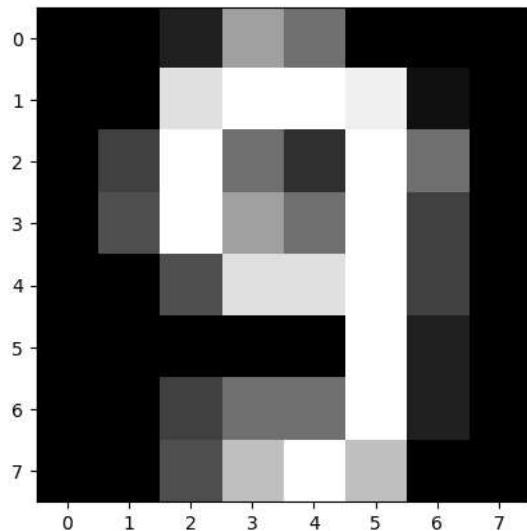


McAfee | WebAdvisor

Your download's being scanned.
We'll let you know if there's an issue.

```
plt.imshow(digit.images[1795], cmap='gray')
```

```
<matplotlib.image.AxesImage at 0x7e3ff0df3310>
```



```
x=digit.data
y=pd.DataFrame(digit.target)
```

```
y.value_counts()
```

```
count
```

```
0
3    183
1    182
5    182
4    181
6    181
9    180
7    179
0    178
2    177
8    174
```

```
from sklearn.preprocessing import StandardScaler
```

```
sc=StandardScaler()
```

```
x_std=sc.fit_transform(x)
```

```
x1=x_std.T
x1
```

```
array([[ 0.         ,  0.         ,  0.         , ...,  0.         ,
         0.         ,  0.         ],
       [-0.33501649, -0.33501649, -0.33501649, ..., -0.33501649,
        -0.33501649, -0.33501649],
       [-0.04308102, -1.09493684, -1.09493684, ..., -0.88456568,
        -0.67419451,  1.00877481],
       ...,
       [-1.14664746,  0.54856067,  1.56568555, ..., -0.12952258,
         0.8876023 ,  0.8876023 ],
       [-0.5056698 , -0.5056698 ,  1.6951369 , ..., -0.5056698 ,
        -0.5056698 , -0.26113572],
```



McAfee | WebAdvisor



Your download's being scanned.
We'll let you know if there's an issue.

```
cov_mat=np.cov(x1)
print(cov_mat)
```

```
print(cov_mat.shape)
```

```
eig_value,eig_vecs=np.linalg.eig(cov_mat)
```

eig vecs

```
tot=sum(eig_value)
var_exp=[(i/tot)*100 for i in eig_value]
var_exp
```

[12.033916097734904,
 9.561054403097883,
 8.444414892624566,
 6.498407907524167,
 4.860154875966403,
 4.214119869271944,
 3.9420828035674003,
 3.3893809246383264,
 2.9982201101625232,
 2.9320025512522068,
 2.781805463550326,
 2.5770550925819995,
 2.2753033157642486,
 2.227179739514353,
 2.1652294318492435,
 1.9141666064421259.

Your download's being scanned.
We'll let you know if there's an issue.

```

1.7755470851681938,
1.6380692742844212,
1.5964601688623428,
1.4891911870878194,
1.3479695658179378,
1.2719313702347599,
1.1658373505919493,
1.0576465985363224,
0.9753159471981139,
0.9445589897319981,
0.8630138269707229,
0.8366428536685183,
0.7976932484112456,
0.7464713709260611,
0.725582151370274,
0.691911245481183,
0.6539085355726172,
0.640792573845988,
0.5913841117223424,
0.571162405223525,
0.5236368034166339,
0.08253509448180278,
0.48180758644514254,
0.45371925985845096,
0.10369573015571817,
0.42316275323278085,
0.4060530699790386,
0.39708480827582743,
0.12510074249730116,
0.13511841133708546,
0.14776269410608744,
0.161236062256729,
0.1679463874955841,
0.18331849919718232,
0.19551242601981858,
0.20359763452537666,
0.21825685771200845,
0.23448300553563545,
0.3564933031426177,
0.25917494088146453,
0.2764892635235469,
0.21070718117030071

```

```

cum_var_exp=np.cumsum(var_exp)
cum_var_exp

```

```

array([ 12.0339161,  21.5949705,  30.03938539,  36.5377933,
        41.39794818,  45.61206805,  49.55415085,  52.94353177,
        55.94175279,  58.87375534,  61.6555608,  64.23261589,
        66.50791921,  68.73509895,  70.90032838,  72.81449499,
        74.59004207,  76.22811135,  77.82457152,  79.3137627,
        80.66173227,  81.93366364,  83.09950099,  84.15714759,
        85.13246353,  86.07702252,  86.94003635,  87.77667921,
        88.57437245,  89.32084382,  90.04642598,  90.73833722,
        91.39224576,  92.03303833,  92.62442244,  93.19558485,
        93.71922165,  93.80175675,  94.28356433,  94.73728359,
        94.84097932,  95.26414208,  95.67019515,  96.06727995,
        96.1923807,  96.32749911,  96.4752618,  96.63649786,
        96.80444425,  96.98776275,  97.18327518,  97.38687281,
        97.60512967,  97.83961267,  98.19610598,  98.45528092,
        98.73177018,  99.07255736,  99.36113266,  99.68896799,
        100., 100., 100., 100.])

```

```

plt.figure(figsize=(10,6))
plt.bar(range(len(var_exp)),var_exp,label='individual variance explained',color='g')
plt.step(range(len(cum_var_exp)),cum_var_exp,label="cum variance explained")
plt.ylabel("variance explained")
plt.xlabel("principal component")
plt.legend()
plt.show()

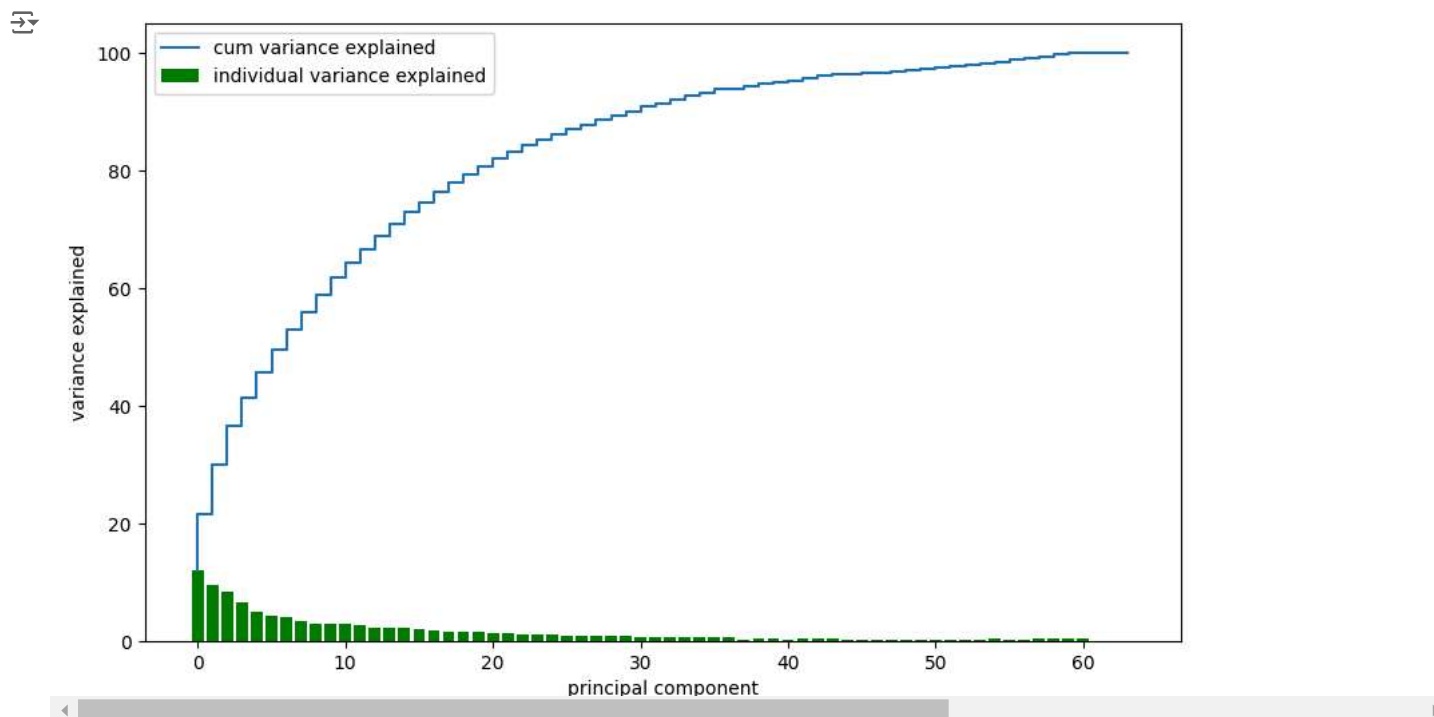
```



McAfee | WebAdvisor



Your download's being scanned.
We'll let you know if there's an issue.



```
from sklearn.model_selection import train_test_split
```

```
xtrain,xtest,ytrain,ytest=train_test_split(x_std,y,test_size=0.20)
```

```
from sklearn.decomposition import PCA
```

```
#pca=PCA(n_components=30)
pca=PCA(0.90)
pca_xtrain=pca.fit_transform(xtrain)
pca_xtest=pca.transform(xtest)
```

```
pca_xtrain.shape
```

```
(1437, 31)
```

```
pca_xtest.shape
```

```
(360, 31)
```

```
from sklearn.tree import DecisionTreeClassifier
```

```
dt=DecisionTreeClassifier()
```

```
dt.fit(pca_xtrain,ytrain)
```

```
DecisionTreeClassifier
DecisionTreeClassifier()
```

```
pred=dt.predict(pca_xtest)
```

```
pred
```

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



McAfee | WebAdvisor

Your download's being scanned.
We'll let you know if there's an issue.

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

ytest

```
0
1776 5
170 8
1113 7
724 0
1684 7
... ...
1163 6
161 9
1694 7
285 9
1077 0
```

360 rows x 1 columns

```
from sklearn.metrics import confusion_matrix, accuracy_score
```

```
confusion_matrix(ytest, pred)
```

```
array([[26, 0, 0, 0, 0, 0, 1, 0, 0, 0],
       [0, 32, 0, 0, 1, 0, 0, 0, 3, 0],
       [0, 0, 25, 2, 0, 0, 0, 0, 1, 0],
       [0, 0, 2, 26, 0, 1, 0, 0, 2, 1],
       [2, 0, 0, 0, 36, 0, 1, 3, 0, 1],
       [2, 0, 1, 0, 1, 36, 0, 0, 4, 4],
       [1, 0, 1, 0, 0, 0, 36, 0, 0, 0],
       [0, 0, 0, 1, 0, 0, 0, 32, 1, 1],
       [0, 2, 1, 2, 0, 3, 1, 1, 21, 2],
       [0, 2, 2, 1, 0, 1, 0, 2, 4, 28]])
```

```
accuracy_score(ytest, pred)
```

```
0.8277777777777777
```


```
sns.heatmap(confusion_matrix(ytest, pred), annot=True, cmap='Greens')
```

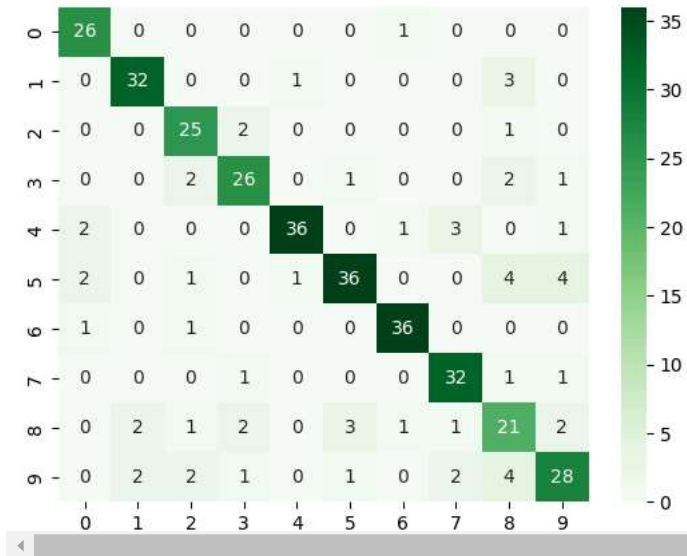


McAfee | WebAdvisor



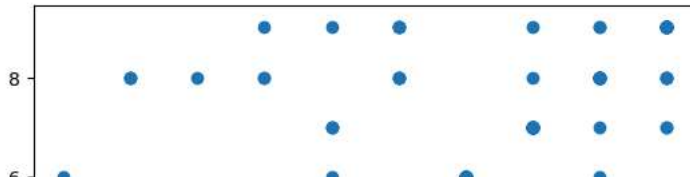
Your download's being scanned.
We'll let you know if there's an issue.

 <Axes: >



```
plt.scatter(ytest,pred)
```

 <matplotlib.collections.PathCollection at 0x7e3ff43ae200>



McAfee | WebAdvisor



Your download's being scanned.
We'll let you know if there's an issue.