In [1]:

```
from sklearn.datasets import load_digits
digits = load_digits()
digits['data'].shape
```

Out[1]:

```
(1797, 64)
```

In [3]:

```
from sklearn.decomposition import PCA
pca = PCA(2)
projected = pca.fit_transform(digits.data)
projected.shape
```

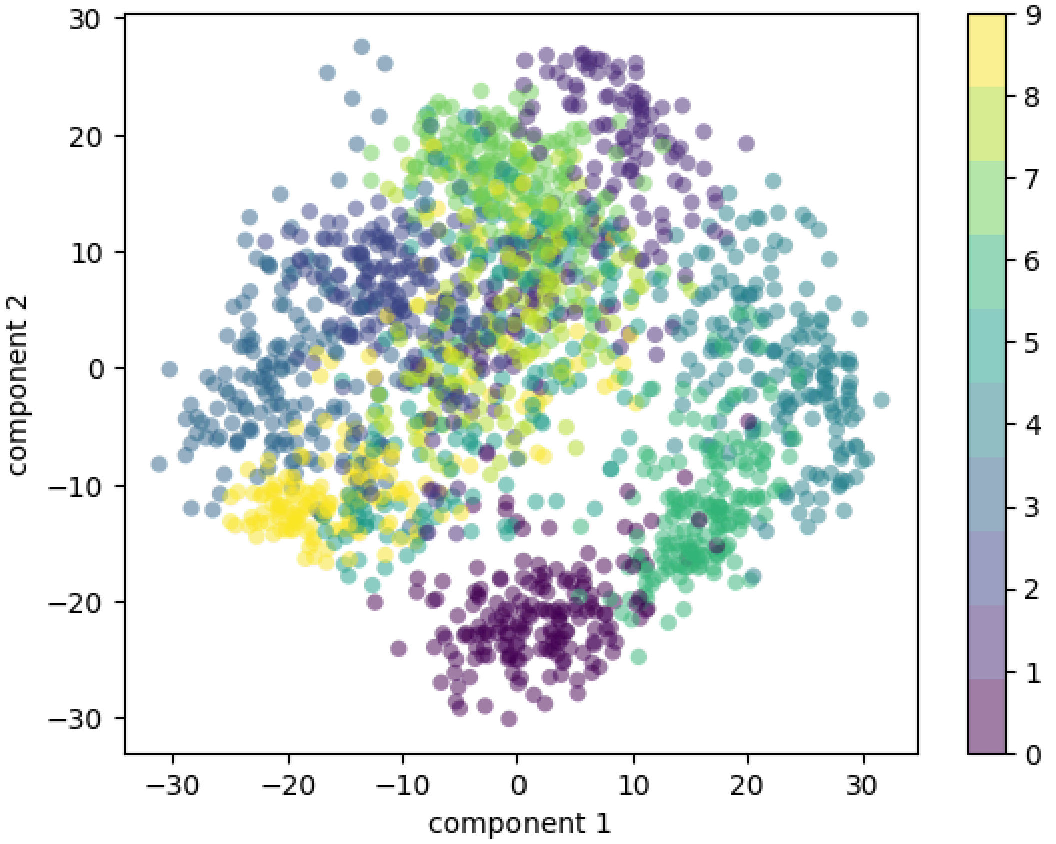
Out[3]:

```
(1797, 2)
```

In [6]:

```
import matplotlib.pyplot as plt

plt.scatter(projected[:, 0], projected[:, 1], c=digits.target, edgecolor='none',
plt.xlabel('component 1')
plt.ylabel('component 2')
plt.colorbar()
plt.show()
```

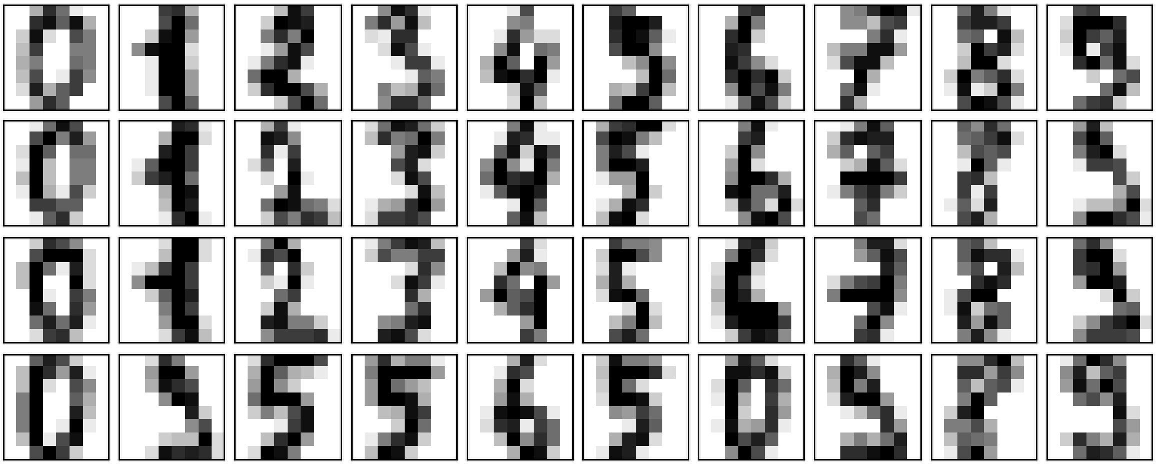
```
C:\Users\fl\AppData\Local\Temp\ipykernel_4792\483197420.py:3: MatplotlibDeprecati
onWarning: The get_cmap function was deprecated in Matplotlib 3.7 and will be rem
oved two minor releases later. Use ``matplotlib.colormaps[name]`` or ``matplotli
b.colormaps.get_cmap(obj)`` instead.
  plt.scatter(projected[:, 0], projected[:, 1], c=digits.target, edgecolor='non
e', alpha=0.5, cmap=plt.cm.get_cmap('viridis', 10))
```



In [7]:

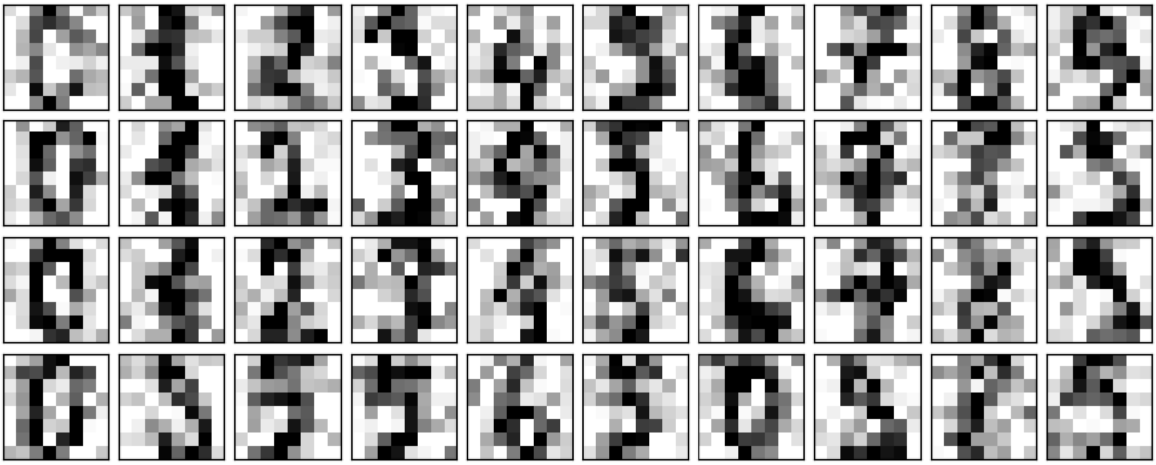
```
import matplotlib.pyplot as plt
from sklearn import datasets
digits = datasets.load_digits()
```

```
def plot_digits(data):
    fig, axes = plt.subplots(4, 10, figsize=(10, 4),
                             subplot_kw={'xticks': [], 'yticks': []},
                             gridspec_kw={'hspace': 0.1, 'wspace': 0.1})

    for i, ax in enumerate(axes.flat):
        ax.imshow(data[i].reshape(8, 8), cmap='binary', interpolation='nearest',

plot_digits(digits.data)
plt.show()
```



In [9]:

```
import numpy as np
np.random.seed(42)
nosiy = np.random.normal(digits.data,4)
# plt.show(nosiy)
plot_digits(nosiy)
```



In [11]:

```
pca = PCA(0.5).fit(nosiy)
pca.n_components_
```

Out[11]:

```
12
```

In [12]:

```
components = pca.transform(nosiy)
filterrd = pca.inverse_transform(components)
plot_digits(filterrd)
```