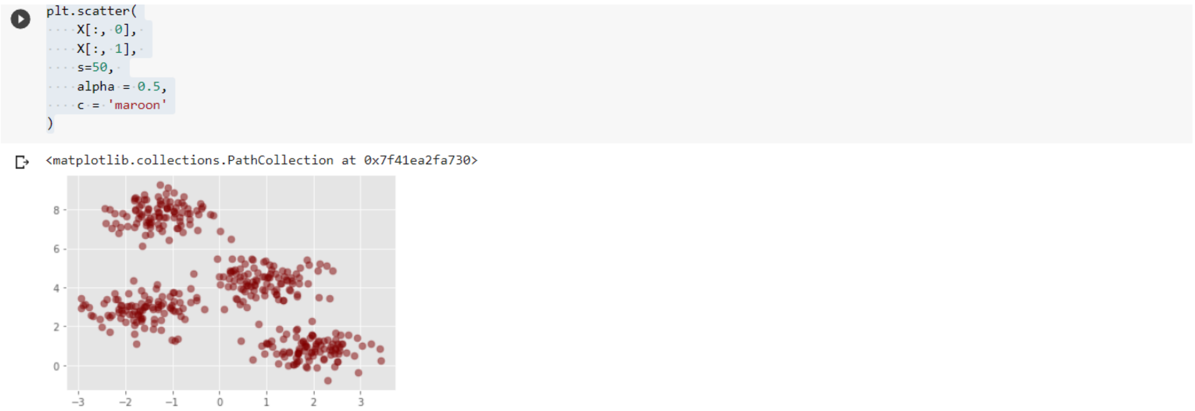
Practical 6

Aim: K-means Algorithm

Code:

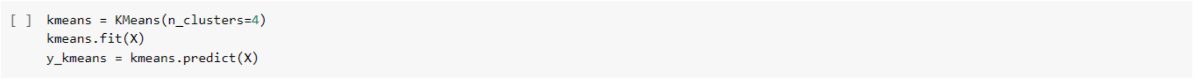
```
import matplotlib.pyplot as plt
import numpy as np

from sklearn.cluster import KMeans
from sklearn.datasets import make_blobs
plt.style.use('ggplot')

X, y_true = make_blobs(
    n_samples=400,
    centers=4,
    cluster_std=0.60,
    random_state=0
)

plt.scatter(
    X[:, 0],
    X[:, 1],
    s=50,
    alpha = 0.5,
    c = 'maroon'
)

kmeans = KMeans(n_clusters=4)
kmeans.fit(X)
y_kmeans = kmeans.predict(X)

centriods = kmeans.cluster_centers_

plt.scatter(
    X[:, 0],
    X[:, 1],
    c=y_kmeans,
    s=50,
    cmap='Blues',
    alpha = 0.5
)

plt.scatter(
    centriods[:, 0],
    centriods[:, 1],
    c='black',
    s=200,
    alpha=0.55
)

sse = []
for k in range(1, 8):
    km = KMeans(n_clusters=k)
    km.fit(X)
    sse.append(km.inertia_)

plt.figure(figsize=(6, 6))
plt.plot(range(1, 8), sse, '-o', c = 'maroon')
plt.xlabel('Count of Clusters')
plt.ylabel('SSE');
```

Output:

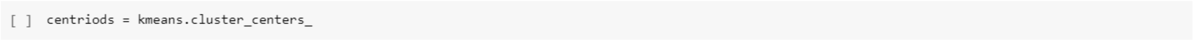


```
plt.scatter(
    X[:, 0],
    X[:, 1],
    s=50,
    alpha = 0.5,
    c = 'maroon'
)
```

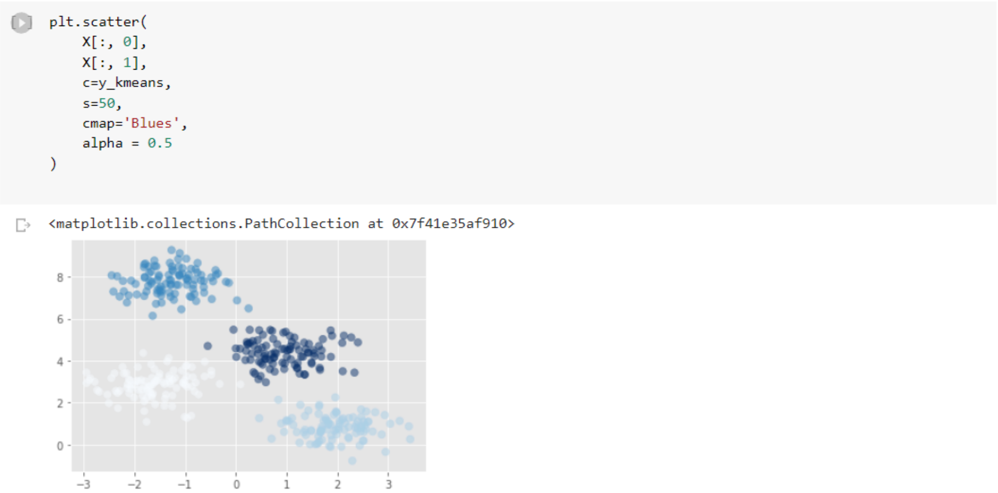
```
<matplotlib.collections.PathCollection at 0x7f41ea2fa730>
```



```
[ ] kmeans = KMeans(n_clusters=4)
    kmeans.fit(X)
    y_kmeans = kmeans.predict(X)
```

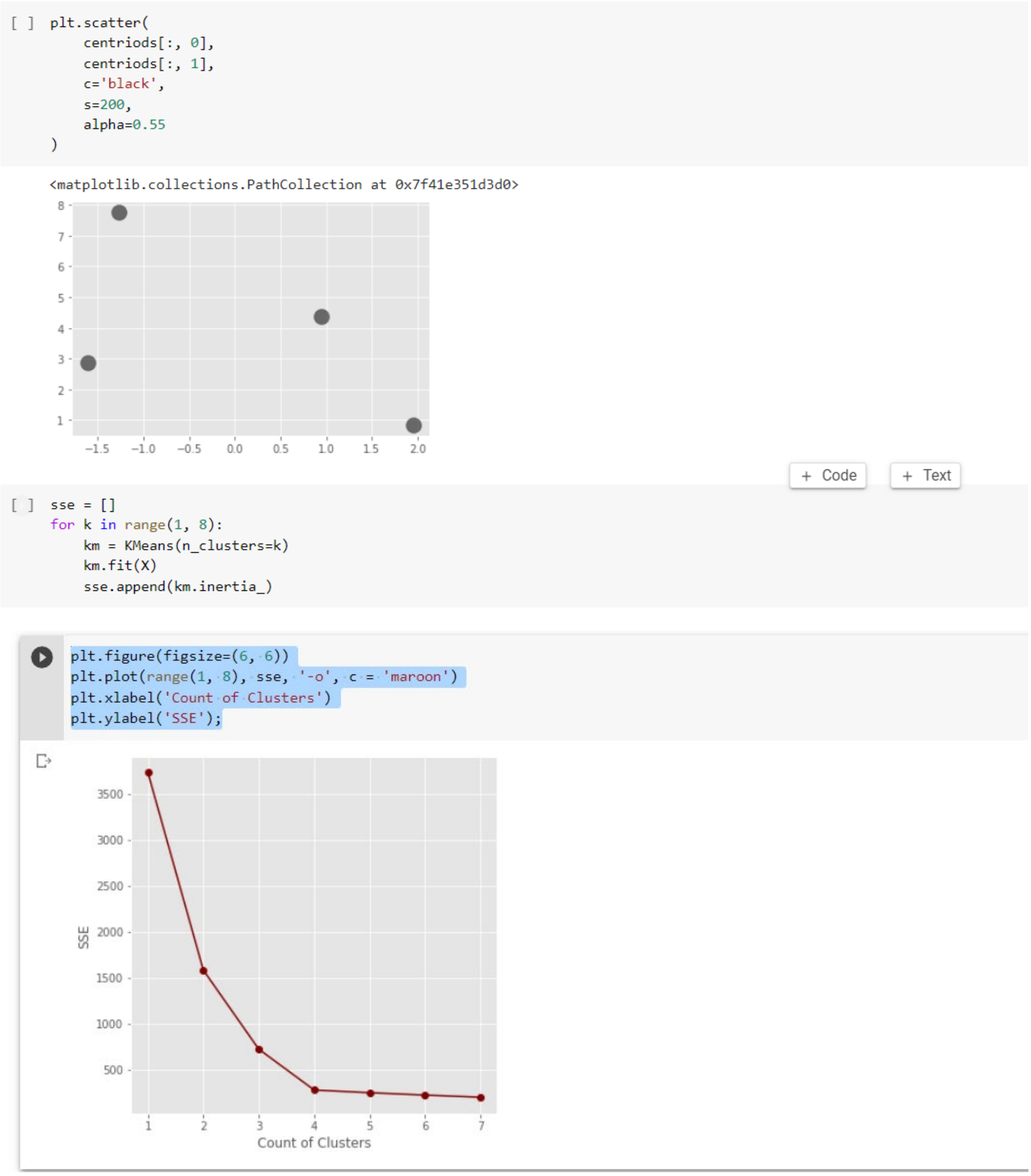


```
[ ] centriods = kmeans.cluster_centers_
```



```
plt.scatter(
    X[:, 0],
    X[:, 1],
    c=y_kmeans,
    s=50,
    cmap='Blues',
    alpha = 0.5
)
```

```
<matplotlib.collections.PathCollection at 0x7f41e35af910>
```



```
[ ] plt.scatter(
        centriods[:, 0],
        centriods[:, 1],
        c='black',
        s=200,
        alpha=0.55
    )
```

```
<matplotlib.collections.PathCollection at 0x7f41e351d3d0>
```

```
[ ] sse = []
    for k in range(1, 8):
        km = KMeans(n_clusters=k)
        km.fit(X)
        sse.append(km.inertia_)
```

```
plt.figure(figsize=(6, 6))
plt.plot(range(1, 8), sse, '-o', c = 'maroon')
plt.xlabel('Count of Clusters')
plt.ylabel('SSE');
```