In [1]:

```
from copy import deepcopy
import numpy as np
import pandas as pd
from matplotlib import pyplot as plt
plt.rcParams['figure.figsize'] = (16, 9)
plt.style.use('ggplot')
```

In [3]:

```
data = pd.read_csv('dataset.csv')
print(data.shape)
data.head()
```

(5000, 2)

Out[3]:

	V1	V2
0	4.53320	6.06542
1	5.20651	5.94591
2	5.16664	6.60290
3	5.70023	5.70741
4	5.24035	5.18443

In [4]:

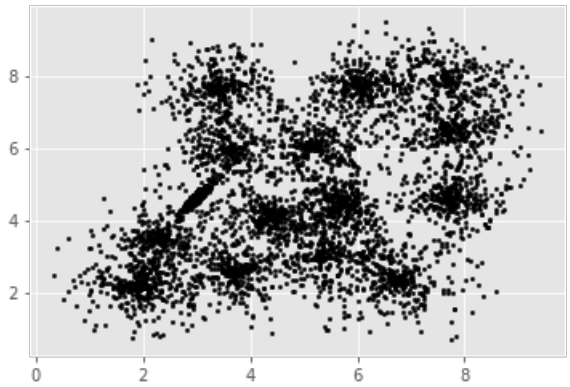
```
f1 = data['V1'].values
f2 = data['V2'].values

for key in data.keys():
    print(key)
X = np.array(list(zip(f1, f2)))
plt.scatter(f1, f2, c='black', s=5)
```

V1
V2

Out[4]:

<matplotlib.collections.PathCollection at 0x26f52a076a0>



In [13]:

```
def dist(a, b, ax=1):
    return np.linalg.norm(a - b, axis=ax)
```

In [14]:

```
k = 1
C_x = np.random.randint(0, np.max(X)-2, size=k)
C_y = np.random.randint(0, np.max(X)-2, size=k)
C = np.array(list(zip(C_x, C_y)), dtype=np.float32)
```

```
print(C)
```

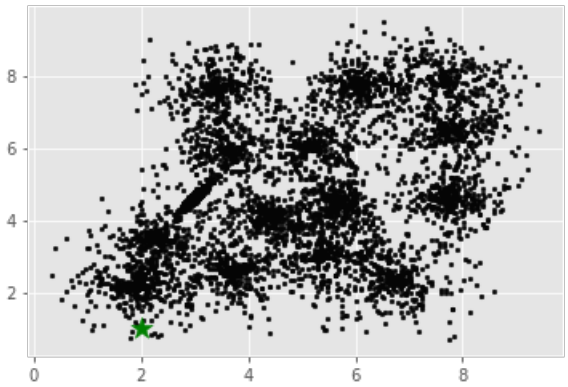
```
[[2. 1.]]
```

In [15]:

```
plt.scatter(f1, f2, c='#050505', s=7)
plt.scatter(C_x, C_y, marker='*', s=200, c='g')
```

Out[15]:

```
<matplotlib.collections.PathCollection at 0x26f52d7bcc0>
```



In [16]:

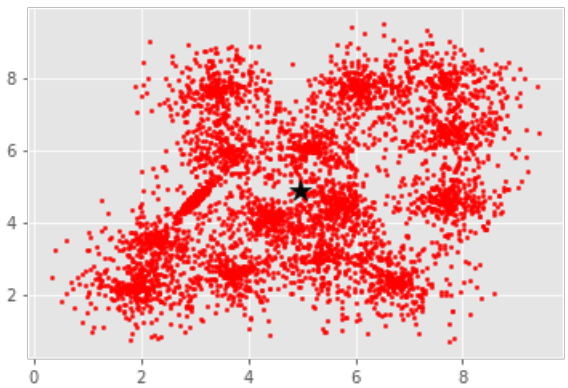
```
C_old = np.zeros(C.shape)
clusters = np.zeros(len(X))
error = dist(C, C_old, None)
while error != 0:
    for i in range(len(X)):
        distances = dist(X[i], C)
        cluster = np.argmin(distances)
        clusters[i] = cluster
    C_old = deepcopy(C)
    for i in range(k):
        points = [X[j] for j in range(len(X)) if clusters[j] == i]
        C[i] = np.mean(points, axis=0)
    error = dist(C, C_old, None)
```

In [17]:

```
colors = ['r', 'g', 'b', 'y', 'c', 'm']
fig, ax = plt.subplots()
for i in range(k):
        points = np.array([X[j] for j in range(len(X)) if clusters[j] == i])
        ax.scatter(points[:, 0], points[:, 1], s=5, c=colors[i])
ax.scatter(C[:, 0], C[:, 1], marker='*', s=200, c='#050505')
```

Out[17]:

```
<matplotlib.collections.PathCollection at 0x26f52e0d898>
```



In [18]:

```
from sklearn.cluster import KMeans
```

```
kmeans = KMeans(n_clusters=3)
kmeans = kmeans.fit(X)
labels = kmeans.predict(X)
centroids = kmeans.cluster_centers_
```

In [19]:

```
print(C)
print(centroids)
```

```
[[4.9656467 4.8579974]]
[[3.04496078 3.59624062]
 [5.84682833 7.14996795]
 [6.38988575 3.6626096 ]]
```

In [20]:

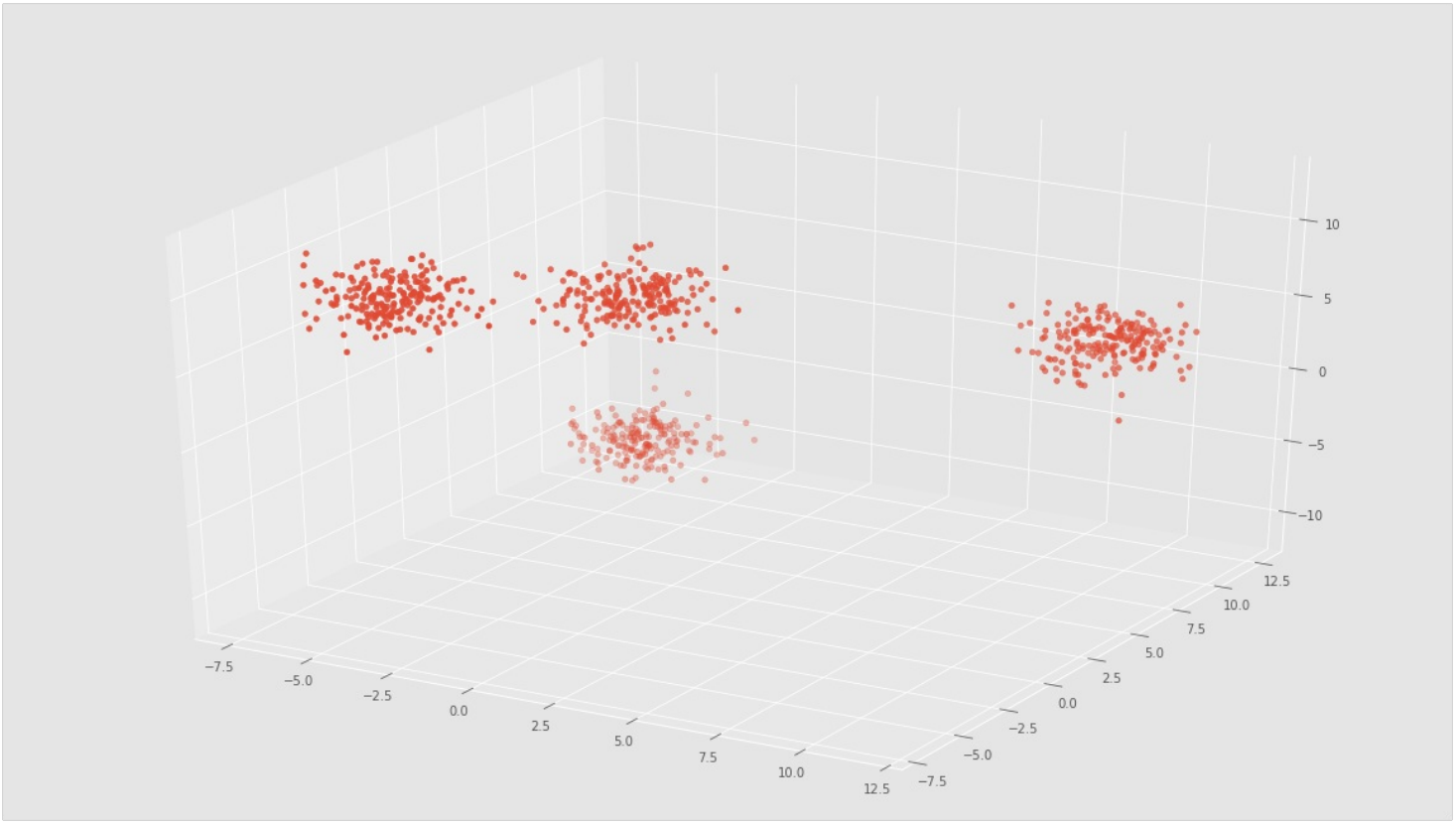
```
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
from sklearn.cluster import KMeans
from sklearn.datasets import make_blobs
plt.rcParams['figure.figsize'] = (16, 9)
X, y = make_blobs(n_samples=800, n_features=3, centers=4)
```

In [21]:

```
fig = plt.figure()
ax = Axes3D(fig)
ax.scatter(X[:, 0], X[:, 1], X[:, 2])
```

Out[21]:

```
<mpl_toolkits.mplot3d.art3d.Path3DCollection at 0x26f55804a20>
```



In [22]:

```
kmeans = KMeans(n_clusters=4)
kmeans = kmeans.fit(X)
labels = kmeans.predict(X)
C = kmeans.cluster_centers_
```

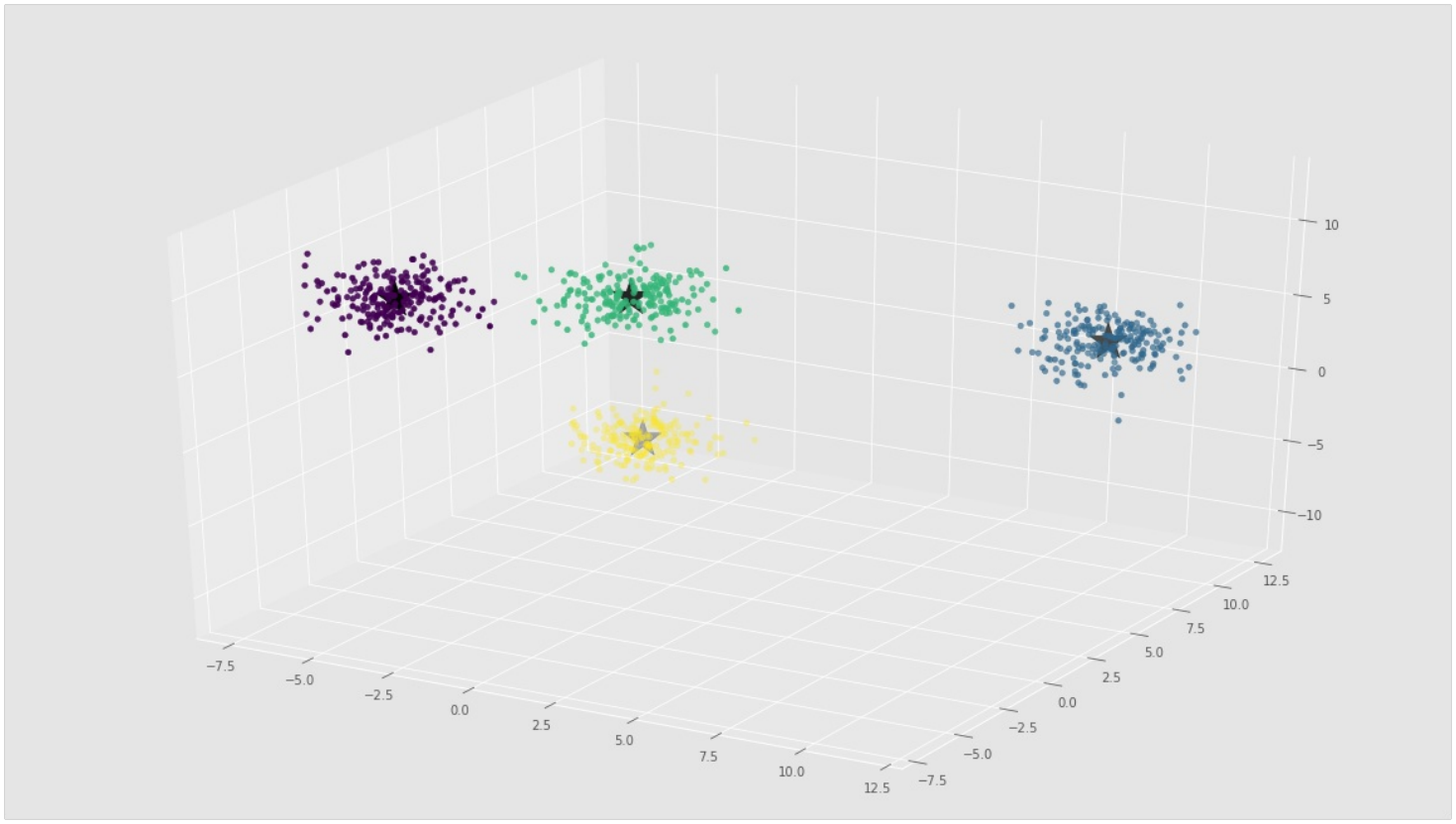
In [23]:

```
fig = plt.figure()
```

```
fig = plt.figure()
ax = Axes3D(fig)
ax.scatter(X[:, 0], X[:, 1], X[:, 2], c=y)
ax.scatter(C[:, 0], C[:, 1], C[:, 2], marker='*', c='#050505', s=1000)
```

Out[23]:

<mpl_toolkits.mplot3d.art3d.Path3DCollection at 0x26f55869470>



In [ ]:

```
```