Lloyd's algorithm

Finding n points equally spaced in a grid - using K means on a random set of points

Wikipedia Article (https://en.wikipedia.org/wiki/Lloyd%27s_algorithm)

In [27]:
```
%matplotlib inline
import matplotlib.pyplot as plt
import seaborn as sns; sns.set()  # for plot styling
import numpy as np
```

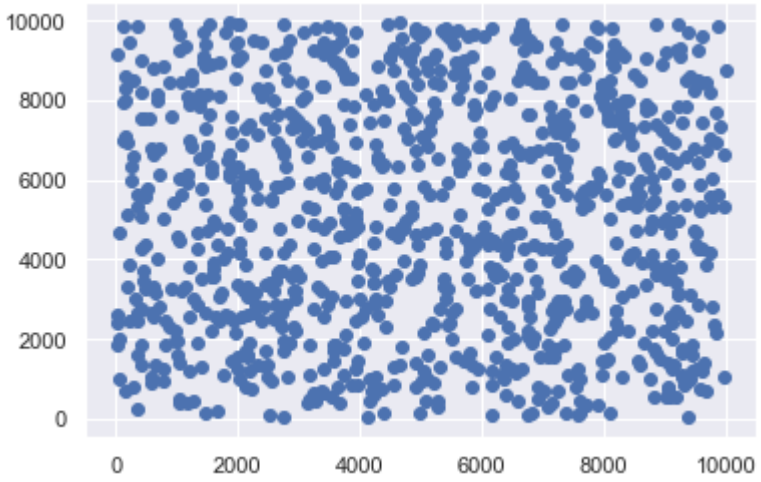
In [64]:
```
x = np.random.randint(0 , 9999, 999)
y = np.random.randint(0 , 9999, 999)
```

In [65]:
```
firstPoint=[x[0], y[0]]
secondPoint=[x[1], y[1]]

points=np.empty([len(x), 2])
n=0
while n < len(x):
    point = [x[n], y[n]]
    points[n] = point
    n = n+1
```

In [66]:
```
plt.scatter(x, y)
```

Out[66]:
```
<matplotlib.collections.PathCollection at 0x1b381c134c0>
```



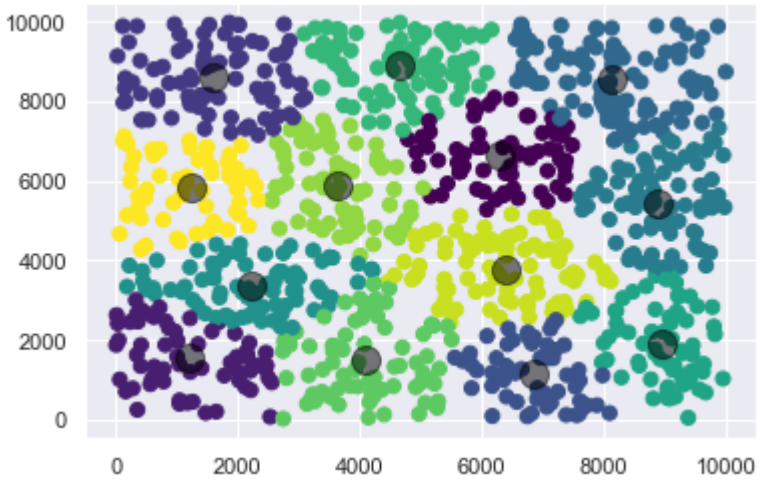
In [67]:
```
numPoints = 13
```

In [68]:
```
from sklearn.cluster import KMeans
kmeans = KMeans(n_clusters=numPoints)
kmeans.fit(points)
y_kmeans = kmeans.predict(points)
```

In [69]:
```
plt.scatter(x, y, c=y_kmeans, s=50, cmap='viridis')

centers = kmeans.cluster_centers_
plt.scatter(centers[:, 0], centers[:, 1], c='black', s=200, alpha=0.5);
print("Centers: ")
print(centers)
```

```
Centers:
[[6281.10447761 6628.82089552]
 [1199.640625   1544.        ]
 [1595.25609756 8592.23170732]
 [6848.98214286 1158.69642857]
 [8120.26213592 8576.3592233 ]
 [8883.01204819 5459.15662651]
 [2215.23170732 3368.90243902]
 [8948.91428571 1904.78571429]
 [4646.52808989 8922.19101124]
 [4080.27272727 1504.67532468]
 [3647.81081081 5885.52702703]
 [6376.24731183 3785.05376344]
 [1225.72881356 5840.66101695]]
```



In [ ]: