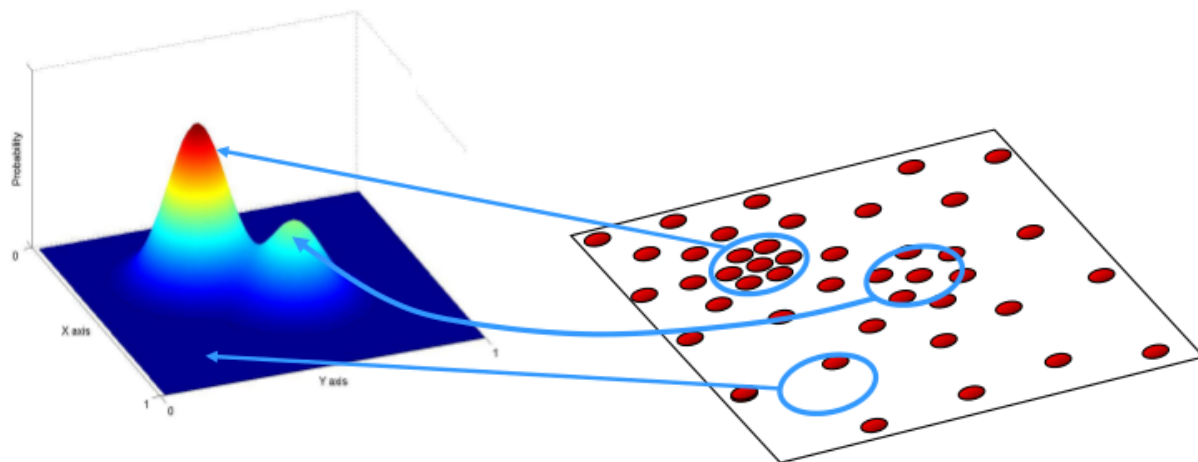


## Orta Yerdəyişmə Klasteri

### Mean Shift Clustering:

Klasterləşmə üçün istifadə edilən alqoritmlərdən biri orta yerdəyişmə klasterləşdirmə (mean-shift clustering) adlanır. Bu üsulda klaster mərkəzləri (cluster centers) əvvəlcə təsadüfi olaraq müəyyən edilir. Daha sonra hər bir obyektin bu mərkəzlərdən hansına daha yaxın olduğu müəyyən edilir və obyekt həmin klasterə alınır. Sonra obyektlərin mərkəzi istiqamətləri yeni klasterlərə uyğun olaraq yenidən tapılır. Eyni proses davamlı olaraq aparılır. Obyektlərin yerləşdirildiyi klasterdə heç bir dəyişiklik olmayana qədər. Bu zaman sıra qurulmuş hesab edilir və klasterləşmə üçün istifadə olunan nəticələr kimi yekun vəziyyətdə olan klaster mərkəzləri götürülür.



### Mean Shift Sinfi

SKLearn skriptinin sklearn.cluster paketində MeanShift adlı sinif klasterləşmə üçün istifadə edilə bilər. Bu sinfə bir nümunə

```
clusterer = MeanShift()
```

formada yaradıla bilər və məlumatlar fit() funksiyasına aşağıdakı kimi verilə bilər:

```
clusterer.fit(X)
```

Daxil olan dəstlər labels\_ xüsusiyyəti ilə əldə edilə bilər:

```
clusterer.labels_
```

Sözügedən xüsusiyyət hər bir obyektin etiketini göstərir. Unique() funksiyası neçə müxtəlif klasterin olduğunu öyrənmək üçün istifadə edilə bilər:

**uniques=np.unique(labels)**

Sözügedən sətirin uzunluğu len() (uzunluq) funksiyası ilə götürülərsə, dəstələrin sayını verir:

**count = len( np.unique(labels) )**

Klasterlərin mərkəz nöqtələrini cluster\_centers\_ adlı sahə ilə də əldə etmək olar:

**clusterer.cluster\_centers\_**

### Orphans

Bəzi hallarda, mərkəz mövqelərindən uzaq olan hər hansı bir dəstə götürməmək lazım ola bilər. Bunlara (orphan), yəni heç bir klasterə aid olmayan obyektlər kimi baxılır. Bu halda, cluster\_all xüsusiyyətinə False dəyəri verilir:

**clusterer = MeanShift( cluster\_all = False )**

Bu dəyişən istifadə edildikdə, orphan obyektlərin etiket dəyəri -1 kimi qaytarılır.



```

import numpy as np
from sklearn.cluster import MeanShift
import matplotlib.pyplot as plt
from sklearn.datasets import make_blobs
centers = [[1,2],[8,3],[4,10]]
X, y = make_blobs(
    n_samples = 500, centers = centers, cluster_std = 1)
clusterer = MeanShift()
clusterer.fit(X)
labels = clusterer.labels_
print("Cluster Labels:\n",labels)
cluster_centers = clusterer.cluster_centers_
print("Cluster Centers:\n",cluster_centers)
n_clusters_ = len(np.unique(labels))
print("N_Clusters:", n_clusters_)
plt.scatter(X[:,0],X[:,1])
colors = 10*['r.','g.','b.','c.','y.','m.','w.']
for i in range(len(X)):
    plt.plot( X[i][0], X[i][1],
              colors[labels[i]], markersize = 10)
plt.scatter(cluster_centers[:,0],cluster_centers[:,1],
            marker = "x",color = 'w', s = 350,
            linewidths = 10, zorder = 10)
plt.show()

```

Cluster Labels:

```

[2 0 1 0 1 1 1 2 1 0 2 1 0 2 2 2 2 0 1 1 2 2 0 0 2 0 1 0 0 0 1 1 0 2 0 2 0
 2 0 2 0 0 0 1 1 2 0 0 2 1 1 0 1 1 2 1 0 0 0 1 2 1 0 2 1 0 1 2 0 2 1 0 1 1
 2 0 0 1 1 2 0 2 2 2 2 2 2 1 0 1 1 0 1 1 0 2 0 0 1 0 0 2 1 1 0 0 1 1 2 0 1
 1 1 0 0 0 2 1 1 1 0 1 1 0 0 2 2 1 1 2 0 1 1 2 1 1 1 2 1 1 2 0 2 0 2 2 1 1
 0 0 1 1 2 2 2 2 1 1 1 1 2 0 1 2 1 0 1 0 1 2 0 1 0 1 0 2 0 2 0 2 2 2 1 0 1
 0 1 0 0 0 2 0 2 0 2 0 2 0 1 2 0 0 2 2 2 0 0 1 1 1 0 0 2 0 0 0 0 2 2 1 2 0
 2 1 1 1 0 0 1 0 1 2 1 2 0 2 2 1 1 1 2 0 1 1 1 0 2 1 2 0 0 2 2 1 2 0 1 1 2
 2 0 2 1 2 0 1 1 0 2 2 1 0 0 2 2 1 0 2 2 2 2 2 2 0 2 2 1 2 0 1 0 0 2 1 0 0 2
 0 1 2 1 2 2 2 2 1 2 1 0 2 1 1 2 0 1 1 2 1 0 0 1 0 1 0 0 1 0 1 0 0 0 2 0 2
 1 1 0 2 1 2 2 2 1 0 2 1 0 1 2 2 2 2 2 0 1 0 2 1 0 1 2 2 1 0 2 0 2 2 0 0 2
 2 2 2 2 1 0 2 2 2 1 0 0 0 1 2 0 2 2 2 1 0 0 2 2 2 0 0 0 1 2 0 0 1 1 0 1 1
 0 0 0 0 1 2 0 2 2 1 2 0 1 1 1 2 2 2 2 1 1 2 2 1 2 0 2 2 0 2 2 0 1 2 2 0 2
 1 2 0 0 0 0 1 1 1 1 1 0 0 0 2 1 0 1 2 2 1 1 1 2 1 0 0 1 0 0 0 1 1 2 1 1 1
 1 0 1 0 0 0 1 1 1 0 1 1 2 2 0 0 2 1 2]

```

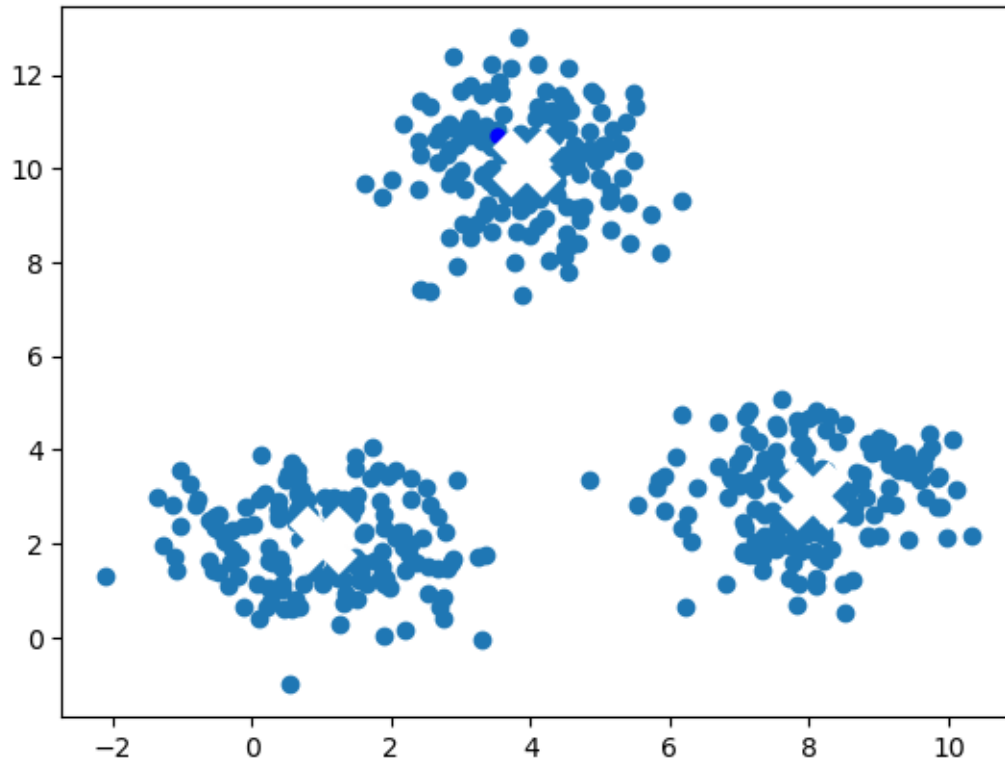
Cluster Centers:

```

[[ 8.05076491  3.03088498]
 [ 1.00874093  2.07979236]
 [ 3.94680822 10.14285598]]

```

N\_Clusters: 3



Yuxarıdakı misalda `make_blobs()` metodundan istifadə edərək 3 qovşaq verilməklə 500 təsadüfi obyekt yaradılır. Klasterləşmə `X` ilə göstərilən verilənləri yaradılmış `MeanShift` obyektinin `fit()` metoduna verməklə həyata keçirilir. Sonra `labels_xassəsi` ilə hər bir obyektin hansı klasterə aid olduğunu görə bilərsiniz; 0, 1, 2 formatında ədəd kimi götürülür. Nümunədə klasterlərin mərkəzləri `cluster_centers_sahəsi` ilə götürülür.