

Propagación de la afinidad

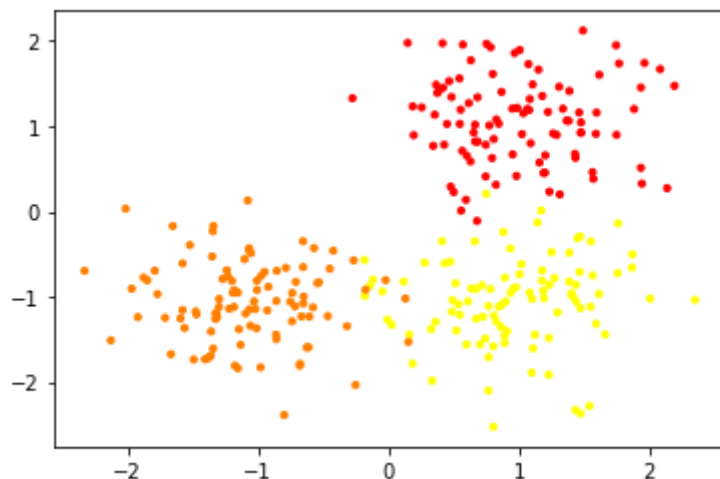
```
In [2]: from sklearn.cluster import AffinityPropagation
from sklearn import metrics
from sklearn.datasets import make_blobs
```

```
In [33]: centers = [[1,1], [-1,-1], [1,-1]]
X, labels = make_blobs(n_samples=300, centers=centers, cluster_std=0.5, random_state=0)
```

```
In [34]: import matplotlib.pyplot as plt
from itertools import cycle
```

```
In [35]: plt.scatter(X[:,0], X[:,1], c=labels, s = 10, cmap = "autumn")
#autum es una paleta de colores, autum es otoñal y s es el tamaño
```

```
Out[35]: <matplotlib.collections.PathCollection at 0x1de8b119c10>
```



```
In [26]: af = AffinityPropagation(random_state=1, preference=-50).fit(X)
```

af es el modelo de propagación de afinidad, donde tenemos que los índices de las 300 muestras, cuáles serían nuestros centros, como se ve a continuación [160, 250, 272] y con labels puedo ver a qué clúster pertenece cada uno de los puntos.

```
In [29]: cluster_center_ids = af.cluster_centers_indices_
af.cluster_centers_indices_
```

```
Out[29]: array([160, 250, 272], dtype=int64)
```

```
In [36]: af_labels = af.labels_
af.labels_
```

```
Out[36]: array([0, 1, 2, 0, 0, 2, 1, 1, 2, 0, 1, 0, 0, 1, 0, 1, 1, 1, 1, 1, 0, 0,
        0, 0, 1, 0, 1, 2, 1, 2, 2, 1, 0, 2, 1, 2, 2, 2, 2, 0, 0, 0, 1, 0,
        2, 1, 1, 1, 1, 0, 0, 0, 1, 2, 2, 2, 2, 0, 2, 0, 1, 2, 2, 1, 1, 1,
        2, 1, 2, 0, 2, 1, 0, 0, 1, 1, 2, 2, 0, 1, 2, 0, 1, 2, 2, 0, 0, 1,
        1, 0, 2, 0, 2, 0, 2, 0, 2, 1, 0, 1, 0, 0, 1, 0, 0, 2, 2, 1, 2, 2,
        2, 2, 2, 1, 0, 2, 0, 1, 2, 1, 2, 2, 2, 2, 2, 0, 1, 1, 1, 0, 1,
        1, 2, 0, 2, 0, 1, 2, 0, 1, 1, 2, 0, 1, 0, 2, 1, 2, 2, 2, 0, 0, 0])
```

```
2, 0, 2, 0, 0, 0, 0, 1, 2, 0, 1, 1, 1, 2, 0, 0, 0, 1, 1, 0, 0, 0,
0, 1, 1, 1, 1, 0, 0, 2, 1, 0, 2, 1, 0, 2, 2, 2, 0, 1, 2, 1, 0, 1,
0, 1, 0, 0, 1, 2, 2, 2, 2, 0, 1, 1, 0, 2, 2, 2, 0, 1, 0, 0, 1, 0,
2, 2, 0, 2, 1, 1, 1, 0, 0, 2, 1, 1, 1, 2, 1, 0, 2, 2, 0, 2, 2, 1,
0, 1, 1, 0, 0, 1, 0, 1, 1, 1, 2, 0, 1, 1, 0, 2, 2, 2, 2, 0, 0, 0,
0, 0, 1, 1, 2, 0, 0, 2, 2, 2, 1, 2, 1, 2, 2, 1, 1, 1, 2, 1, 0, 1,
2, 0, 1, 2, 2, 1, 2, 2, 2, 1, 0, 2, 1, 1], dtype=int64)
```

In [32]:

```
n_clust = len(cluster_center_ids)
n_clust #Vemos tres clusters, parece que 3 centros está bien, es lo que se vé
```

Out[32]:

3

In [45]:

```
def report_affinity_propagation(X):
    af = AffinityPropagation(random_state=1, preference=-50).fit(X)
    cluster_center_ids = af.cluster_centers_indices_
    clust_labels = af.labels_
    n_clust = len(cluster_center_ids)

    print("Número estimado de clusters: %d" %n_clust)
    print("Homogeneidad: %0.3f" %metrics.homogeneity_score(labels, clust_labels))
    print("Compleitud: %0.3f" %metrics.completeness_score(labels, clust_labels))
    print("V-measure: %0.3f" %metrics.v_measure_score(labels, clust_labels))
    print("R2 ajustado: %0.3f" %metrics.adjusted_rand_score(labels, clust_labels))
    print("Información mútua ajustada: %0.3f" %metrics.adjusted_mutual_info_score(labels, clust_labels))
    print("Coeficiente de la silueta: %0.3f" %metrics.silhouette_score(X, labels, metric="sqeuclidean"))

    plt.figure(figsize=(16,9))
    plt.clf()

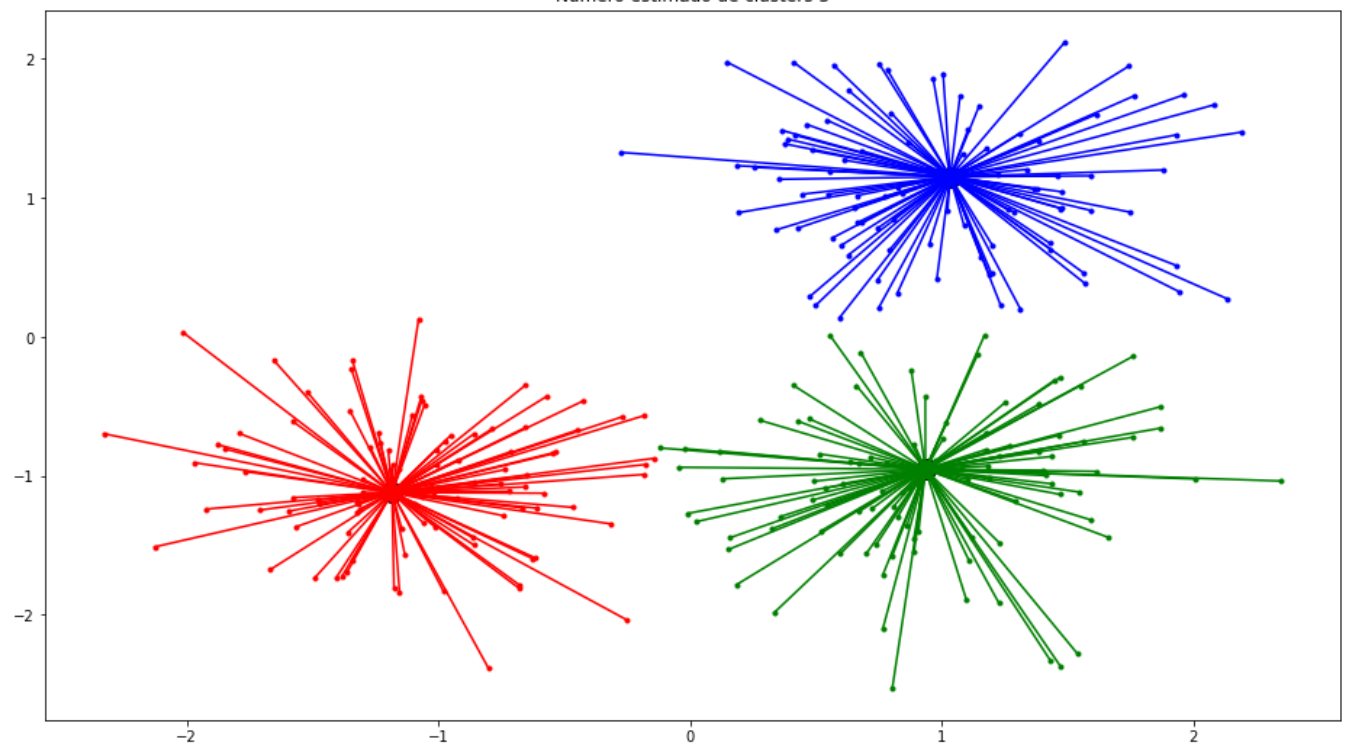
    colors = cycle('bgrcmk')
    for k, col in zip(range(n_clust), colors):
        class_member = (clust_labels == k)
        clust_center = X[cluster_center_ids[k]]
        plt.plot(X[class_member,0], X[class_member, 1], col + '.')
        plt.plot(clust_center[0], clust_center[1], 'o', markerfacecolor=col, markeredgecolor='k')
        for x in X[class_member]:
            plt.plot([clust_center[0], x[0]], [clust_center[1], x[1]], col)
    plt.title("Número estimado de clusters %d" %n_clust)
    plt.show()
```

In [46]:

```
report_affinity_propagation(X)
```

```
Número estimado de clusters: 3
Homogeneidad: 0.872
Compleitud: 0.872
V-measure: 0.872
R2 ajustado: 0.912
Información mútua ajustada: 0.871
Coeficiente de la silueta: 0.735
```

Número estimado de clusters 3



In []: