



# Convergencia do Affinity Propagation

☰ Ciclo	Ciclo 08: Outros algoritmos Clusterização
# Aula	70
🕒 Created	@June 27, 2023 1:34 PM
☑ Done	<input type="checkbox"/>
☑ Ready	<input checked="" type="checkbox"/>

## Objetivo da Aula:

- ☐ Convergência do algoritmo
- ☐ O uso do Affinity Propagation
- ☐ Próxima aula

## Conteúdo:

### ▼ 1. Convergência do algoritmo

```
from matplotlib import pyplot as plt
from sklearn import datasets as dt
from sklearn import cluster as c

# dataset
X, clusters = dt.make_blobs( n_samples=300, centers=4, cluster_std=0.60,
                             random_state=0)

# show dataset
plt.scatter(X[:,0], X[:,1], alpha=0.7, edgecolors='b')

# Convergency testing
p = np.arange( -1, -61, -1 )
ss_list = []

fig = plt.figure()
for i in range( len( p ) ):
    model = c.AffinityPropagation( preference=p[i] )
    model.fit( X )

    labels = model.predict( X )

    # Metric
    ss = m.silhouette_score( X, labels )
    ss_list.append( ss )

# Plot
```

```

plt.plot( p[:i], ss_list[:i] )

# Plot
plt.title( 'Affinity Propagation - Convergence' );
plt.xlabel( 'Preference' );
plt.ylabel( 'Silhouette Score' )
display.clear_output(wait=True)
plt.show()
plt.gcf()
time.sleep(0.05)

# Showing final Clustering
max_index = ss_list.index( max( ss_list ) )
af = AffinityPropagation( preference=p[max_index] )
clustering = af.fit(X)

fig = plt.figure()
plt.title( "Separação dos Clients VIP")
plt.scatter(X[:,0], X[:,1], c=clustering.labels_, cmap='rainbow', alpha=0.7, edgecolors='b');

```

## ▼ 2. O uso do Affinity Propagation

### ▼ 2.1 Vantagens

1. Capaz de encontrar clusters de diferentes formatos e tamanhos
2. Não requer o número de clusters como entrada
3. Utiliza informação sobre a similaridade entre instâncias para criar clusters

### ▼ 2.2. Desvantagens

1. Computacionalmente intensivo, especialmente para conjuntos de dados grandes.

## ▼ 3. Próxima aula

Exercícios