Hoja informativa: Análisis de clústeres

Práctica

```
# Clustering
from sklearn.cluster import KMeans

# n_clusters: número de clústeres
# init: centroides iniciales
model = KMeans(n_clusters=n_clusters, init=centers, random_state=12345)
model.fit(data)

# Obtener los centroides de clúster
print(model.cluster_centers_)
# valor de la función objetivo
print(model.inertia_)
```

```
# Traza el gráfico pairplot con relleno de clúster y centroides
import pandas as pd
from sklearn.cluster import KMeans
import seaborn as sns

centroids = pd.DataFrame(model.cluster_centers_, columns=data.columns)
# Añade una columna con el número de clúster
data['label'] = model.labels_.astype(str)
centroids['label'] = ['0 centroid', '1 centroid', '2 centroid']
# Se necesitará reconfigurar el índice más tarde
data_all = pd.concat([data, centroids], ignore_index=True)

# Traza el gráfico
sns.pairplot(data_all, hue='label', diag_kind='hist')
```

```
# Traza el gráfico Pairgrid con relleno de clúster, centroides iniciales y finales

import pandas as pd
from sklearn.cluster import KMeans
import seaborn as sns

centroids = pd.DataFrame(model.cluster_centers_, columns=data.columns)
```

```
# Añade una columna con el número de clúster
data['label'] = model.labels_.astype(str)
centroids['label'] = ['0 centroid', '1 centroid', '2 centroid']
# Se necesitará reconfigurar el índice más tarde
data_all = pd.concat([data, centroids], ignore_index=True)

# Traza el gráfico
pairgrid = sns.pairplot(data_all, hue='label', diag_kind='hist')
pairgrid.data = pd.DataFrame([[20, 80, 8], [50, 20, 5], [20, 30, 10]], \
                             columns=data.drop(columns=['label']).columns)
pairgrid.map_offdiag(func=sns.scatterplot, s=200, marker='*', color='red')
```

```
# encontrar el número óptimo de clústeres con el método de codo

import matplotlib.pyplot as plt
from sklearn.cluster import KMeans

distortion = []
K = range(1, 8) # número de clústeres de 1 a 7
for k in K:
    model = KMeans(n_clusters=k, random_state=12345)
    model.fit(data)
    distortion.append(model.inertia_)

plt.figure(figsize=(12, 8))
plt.plot(K, distortion, 'bx-')
plt.xlabel('Número de clústeres')
plt.ylabel('Valor de la función objetivo')
plt.show()
```

Teoría

El **análisis de clústeres (clustering)** es la tarea que consiste en combinar observaciones similares en grupos, o clústeres.

Centroide es el centro de un clúster.