

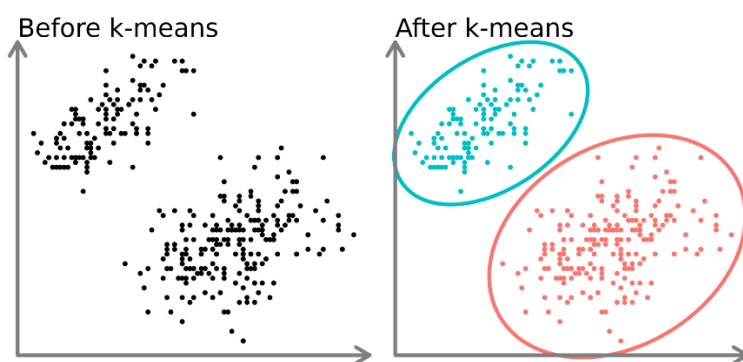


**Centro Universitario de Ciencias Exactas e
Ingenierías**
Universidad de Guadalajara



Práctica 7: k-medias (k-means)

Aprendizaje Máquina



Alumno: Samuel David Pérez Brambila

Código: 222966286

Profesora: Karla Ávila Cárdenas

Sección: D01

Fecha de Entrega: 10 de Noviembre de 2024

Introducción

El algoritmo k-means es “un método de agrupamiento que divide un conjunto de datos en k grupos o clusters. Los datos se agrupan de tal manera que los puntos en el mismo clúster sean más similares entre sí que los puntos en otros clusters.” (Ramírez, 2023)

Del grupo de los algoritmos de aprendizaje no supervisado, K-means sigue siendo uno de los algoritmos más conocidos para el aprendizaje no supervisado, aunque alternativas más avanzadas como DBSCAN o algoritmos basados en clustering espectral han ganado popularidad en ciertos escenarios debido a su capacidad para manejar conjuntos de datos más complejos y de mayor dimensión. De acuerdo con Ramírez (2023), la razón por la que existe este método es porque en 2024, la cantidad total de datos creados, capturados, copiados y consumidos globalmente ha superado los 200 Zettabytes, impulsada principalmente por el crecimiento de dispositivos IoT, la inteligencia artificial generativa y las redes 5G. Con el algoritmo k-means es posible recopilar grandes cantidades de información similar en un mismo lugar, hecho que ayuda a encontrar patrones y hacer predicciones en grandes conjuntos de datos.

Para implementar el algoritmo K-means, primero se especifica el número de clusters deseados (k). Por ejemplo, al establecer k igual a 2, su conjunto de datos se agrupará en 2 grupos, mientras que, si establece k igual a 4 agrupará los datos en 4 grupos. Cada grupo está representado por su centro o centroide, que corresponde a la media aritmética de los puntos de datos asignados al grupo. De esta manera, el algoritmo funciona a través de un proceso iterativo hasta que cada punto de datos está más cerca del centroide de su propio grupo que de los centroides de otros grupos, minimizando la distancia dentro del grupo en cada paso. A continuación, se detalla paso a paso el funcionamiento de este algoritmo:

1. Especificar el número de clústeres deseados (k): El primer paso es especificar cuántos clústeres queremos dividir el conjunto de datos. Este número se denomina k.
2. Seleccionar k puntos al azar del conjunto de datos como los centroides iniciales de cada clúster: Luego, se eligen k puntos al azar del conjunto de datos para servir como los centroides iniciales de cada clúster. Estos centroides son el punto central o el promedio de cada clúster.
3. Asignar cada punto del conjunto de datos al clúster cuyo centroide esté más cerca: A continuación, el algoritmo asigna cada punto del conjunto de datos al clúster cuyo centroide esté más cerca. Para hacer esto, se calcula la distancia entre cada punto y cada centroide y se asigna el punto al clúster cuyo centroide tenga la menor distancia.

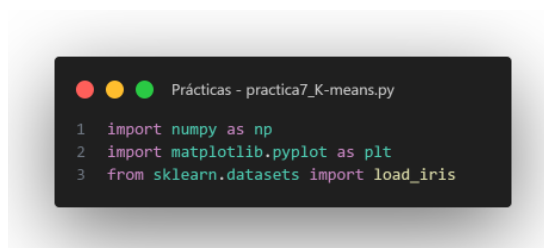
4. Recalcular los centroides de cada clúster como la media de todos los puntos del clúster: Una vez que todos los puntos han sido asignados a un clúster, se recalculan los centroides de cada clúster como la media de todos los puntos del clúster. Esto significa que se actualiza la posición del centroide para reflejar la nueva agrupación.
5. Repetir los pasos 3 y 4 hasta que los centroides de los clústeres ya no cambien o hasta que se alcance el número máximo de iteraciones.

Teniendo entendido cómo debe ser la implementación del algoritmo K-Means, en la siguiente práctica se implementará un algoritmo de este tipo utilizando el dataset de Iris. Para desarrollar este modelo, emplearemos librerías adicionales como Numpy, Pandas, Matplotlib, entre otras, que facilitarán la carga, procesamiento y visualización de los resultados.

Contenido de la Actividad

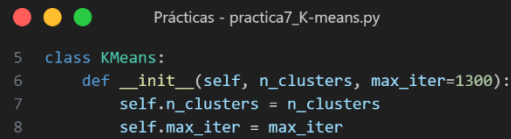
El presente código implementa el algoritmo de K-means para el conjunto de datos de Iris, el cual contiene tres clases de flores (setosa, versicolor, y virginica), utilizando 4 características: largo y ancho del sépalo, y largo y ancho del pétalo. Se aplican tres configuraciones de K-means con 3, 5 y 10 centroides, para explorar la agrupación del conjunto de datos en estos distintos niveles de granularidad. Cada agrupación muestra los datos de Iris junto con los centroides y una simbología que ayuda a identificar cada grupo. Además, el código calcula e imprime las distancias entre los centroides en cada configuración de K-means utilizando las métricas Manhattan, Euclidiana (Minkowski con $p=2$), y Chebyshev, proporcionando una idea de la proximidad relativa entre los grupos en cada caso.

Importación de bibliotecas para tratamiento de datos



- import numpy as np: numpy se usa para manejar arreglos y realizar operaciones matemáticas avanzadas de forma eficiente, como la manipulación de matrices y el cálculo de distancias entre puntos.
- import matplotlib.pyplot as plt: matplotlib.pyplot es una biblioteca para visualizar gráficos en Python. Aquí se utiliza para mostrar la distribución de los clusters y sus centroides.
- from sklearn.datasets import load_iris: load_iris permite cargar el conjunto de datos de Iris, que contiene información sobre tres especies de flores: setosa, versicolor, y virginica, con características como el ancho y largo del sépalo y del pétalo.

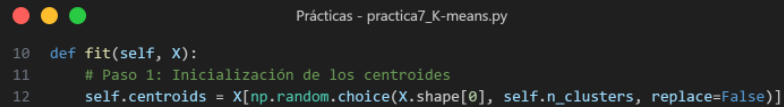
Definición de la clase KMeans



```
Prácticas - practica7_K-means.py
5 class KMeans:
6     def __init__(self, n_clusters, max_iter=1300):
7         self.n_clusters = n_clusters
8         self.max_iter = max_iter
```

- Clase KMeans: Esta clase implementa el algoritmo de K-means, que agrupa datos en un número específico de clusters (`n_clusters`).
- Método `__init__`: Este es el inicializador de la clase.
 - `self.n_clusters`: Define la cantidad de clusters (grupos) en los que se quiere dividir los datos.
 - `self.max_iter`: Limita el número de iteraciones que el algoritmo ejecutará al intentar encontrar la mejor agrupación de datos.

Inicialización de los centroides (entrenar el modelo)



```
Prácticas - practica7_K-means.py
10 def fit(self, X):
11     # Paso 1: Inicialización de los centroides
12     self.centroids = X[np.random.choice(X.shape[0], self.n_clusters, replace=False)]
```

- Método `fit`: Este método ejecuta el algoritmo de K-means. Recibe como entrada `X`, el conjunto de datos.
- Inicialización de Centroides:
 - `self.centroids` selecciona aleatoriamente `n_clusters` puntos del conjunto de datos `X` para servir como los centroides iniciales.
 - `np.random.choice(X.shape[0], self.n_clusters, replace=False)` selecciona índices al azar de `X` sin repetición (`replace=False`), asegurando que cada centroide sea único.

Iteración para Actualización de Centroides

```
Prácticas - practica7_K-means.py
14 for _ in range(self.max_iter):
15     # Paso 2: Asignar etiquetas al centroide más cercano
16     labels = self._assign_labels(X)
```

- Un ciclo for se ejecuta hasta max_iter veces o hasta que el modelo converja.
- Asignación de Etiquetas: labels = self._assign_labels(X) asigna a cada punto en X el índice del centroide más cercano.

Actualización de Centroides

```
Prácticas - practica7_K-means.py
18 # Paso 3: Actualizar los centroides
19 new_centroids = np.array([X[labels == i].mean(axis=0) for i in range(self.n_clusters)])
```

- Esta línea calcula el nuevo centroide para cada cluster promediando las posiciones de los puntos asignados a dicho clúster.
- X[labels == i] selecciona todos los puntos etiquetados con i.
- mean(axis=0) calcula el promedio en cada dimensión (característica) para obtener las coordenadas del nuevo centroide.

Comprobación de convergencia

```
Prácticas - practica7_K-means.py
21 # Comprobar convergencia
22 if np.all(self.centroids == new_centroids):
23     break
24 self.centroids = new_centroids
```

- np.all(self.centroids == new_centroids) compara los centroides actuales con los nuevos. Si son iguales, el algoritmo ha convergido y el ciclo for se detiene (break).
- self.centroids = new_centroids actualiza los centroides para la siguiente iteración si aún no han convergido.

Finalización de fit (o entrenamiento)

```
Prácticas - practica7_K-means.py  
26 self.labels_ = labels  
27 return self
```

Guarda las etiquetas finales en `self.labels_`, que identifica a qué cluster pertenece cada punto, y devuelve el objeto KMeans ajustado.

Método `_assign_labels` para asignar etiquetas

```
Prácticas - practica7_K-means.py  
29 def _assign_labels(self, X):  
30     # Calcular distancias de cada punto a cada centroide  
31     distances = np.sqrt(((X[:, np.newaxis] - self.centroids) ** 2).sum(axis=2))  
32     return np.argmin(distances, axis=1)
```

- Método `_assign_labels`: Asigna a cada punto en X la etiqueta del centroide más cercano.
 - Cálculo de Distancias: `np.sqrt(((X[:, np.newaxis] - self.centroids) ** 2).sum(axis=2))` calcula la distancia euclidiana entre cada punto de X y cada centroide.
 - Asignación de Etiquetas: `np.argmin(distances, axis=1)` devuelve el índice del centroide más cercano para cada punto en X.

Funciones de distancia

```
Prácticas - practica7_K-means.py  
34 # Funciones de distancia  
35 def distancia_manhattan(p, q):  
36     return np.sum(np.abs(q - p))  
37  
38 def distancia_minkowski(p, q, p_value):  
39     return np.power(np.sum(np.power(np.abs(q - p), p_value)), 1 / p_value)  
40  
41 def distancia_chebyshev(p, q):  
42     return np.max(np.abs(q - p))
```

- Distancia de Manhattan: Calcula la suma de las diferencias absolutas entre cada coordenada de p y q.

- Distancia de Minkowski: Generaliza la distancia euclidiana. Con $p_value = 2$, equivale a la distancia euclidiana. Es útil para diferentes métricas de distancia.
- Distancia de Chebyshev: Toma la mayor diferencia absoluta entre las coordenadas de p y q , reflejando la mayor distancia en cualquier dimensión.

Cargar el dataset Iris

```
Prácticas - practica7_K-means.py  
44 # Cargar el conjunto de datos de Iris  
45 data = load_iris()  
46 X = data.data
```

`load_iris()` carga el conjunto de datos y `data.data` contiene las características de las flores.

Ejecución de K-Means y Cálculo de Distancias

```
Prácticas - practica7_K-means.py  
48 # Aplicar K-Means con 3, 5 y 10 centroides  
49 for n_clusters in [3, 5, 10]:  
50     kmeans_custom = KMeans(n_clusters=n_clusters)  
51     kmeans_custom.fit(X)  
52     centroids = kmeans_custom.centroids  
53     labels = kmeans_custom.labels_
```

- Ajustar Modelo para Diferentes Clusters:
Para eficientar el algoritmo, a través de un for se define que haga el proceso de 3, 5 y 10 clústeres desde un inicio, esto lo que hará es primer hacer el proceso para 3 clústeres, al finalizar haría para 5 clústeres y ya al final el de 10 clústeres, así ya no hay necesidad de ajustarlo manualmente.
 - Se define el número de clusters como 3, 5, y 10.
 - Para cada configuración, se ajusta el modelo y se obtienen los centroides y etiquetas.

Cálculo e impresión de distancias entre clústeres

```
Prácticas - practica7_K-means.py

55 # Calcular e imprimir distancias entre los centroides
56 print(f"\nDistancias para {n_clusters} centroides:")
57 for i in range(n_clusters):
58     for j in range(i + 1, n_clusters):
59         dist_manhattan = distancia_manhattan(centroids[i], centroids[j])
60         dist_minkowski = distancia_minkowski(centroids[i], centroids[j], p_value=2) # Euclidiana
61         dist_chebyshev = distancia_chebyshev(centroids[i], centroids[j])
62         print(f"Centroides {i+1} y {j+1}:")
63         print(f" - Distancia Manhattan: {dist_manhattan}")
64         print(f" - Distancia Euclidiana (Minkowski con p=2): {dist_minkowski}")
65         print(f" - Distancia Chebyshev: {dist_chebyshev}")
```

Muestra las distancias entre cada par de centroides en cada configuración de clusters usando las métricas de Manhattan, Minkowski (Euclidiana) y Chebyshev.

Visualización con Leyenda de Clusters

```
Prácticas - practica7_K-means.py

67 # Visualización con cuadro de simbología
68 plt.figure()
69 scatter = plt.scatter(X[:, 0], X[:, 1], c=labels, s=50, cmap='viridis')
70 plt.scatter(centroids[:, 0], centroids[:, 1], c='red', s=200, alpha=0.5, label='Centroides')
71 plt.title(f'Clustering K-Means con {n_clusters} Centroides')
72 plt.xlabel(data.feature_names[0])
73 plt.ylabel(data.feature_names[1])
```

- Visualización de Clusters:
 - Se genera una gráfica que muestra cada punto coloreado según su etiqueta y el centroide de cada cluster en rojo.
 - `data.feature_names[0]` y `[1]` se utilizan para etiquetar los ejes con los nombres de las características.

```
Prácticas - practica7_K-means.py

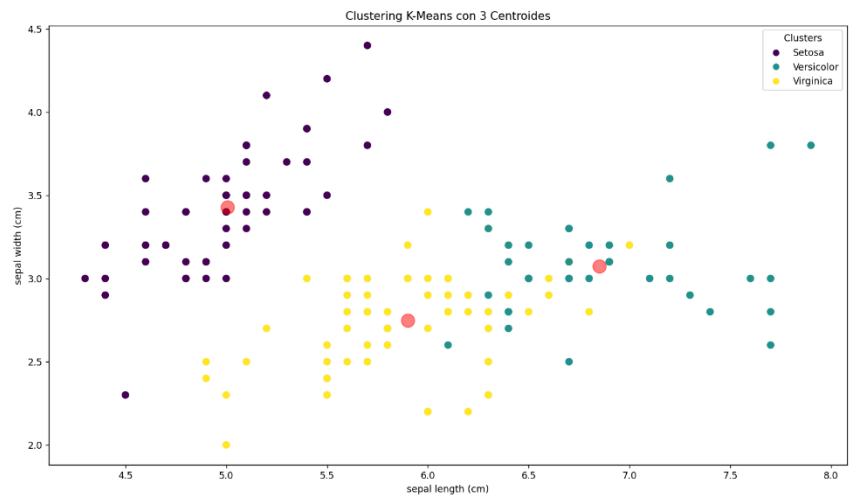
75 # Añadir leyenda para poder visualizar de manera más sencilla los clústeres
76 if n_clusters <= 3:
77     species_names = ["Setosa", "Versicolor", "Virginica"]
78     legend_labels = species_names[:n_clusters]
79 else:
80     legend_labels = [f'Clúster {i+1}' for i in range(n_clusters)]
81 handles, _ = scatter.legend_elements()
82 plt.legend(handles=handles, labels=legend_labels + ['Centroides'], title="Clusters")
83 plt.show()
```

- Leyenda de clústeres:
 - Aplicamos un condicional, donde para el caso de 3 clústeres, le colocamos el nombre a cada clúster en base a la especie incluida en el dataset, solamente aplicaría para el de 3, ya que solamente se incluyen las especies “Setosa”, “Versicolor” y “Virginica” en el dataset.

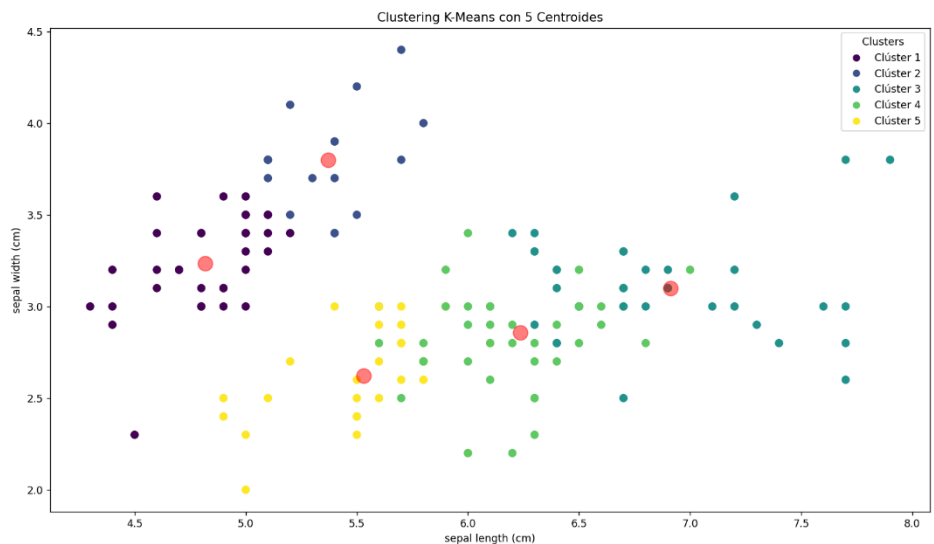
- Define la leyenda para diferenciar visualmente cada cluster, usando `legend_labels`.
- `plt.legend` crea una leyenda final que identifica los clusters y sus centroides en el gráfico.

Es así que obtenemos los siguientes resultados, donde tenemos las condiciones de 3, 5 y 10 clústeres aplicando K-means:

```
Distancias para 3 centroides:
Centroides 1 y 2:
- Distancia Manhattan: 8.303473684210527
- Distancia Euclidiana (Minkowski con p=2): 5.017568519752919
- Distancia Chebyshev: 4.280105263157893
Centroides 1 y 3:
- Distancia Manhattan: 5.6946451612903255
- Distancia Euclidiana (Minkowski con p=2): 3.35693454695641
- Distancia Chebyshev: 2.931548387096775
Centroides 2 y 3:
- Distancia Manhattan: 3.259422750424444
- Distancia Euclidiana (Minkowski con p=2): 1.7971817988854295
- Distancia Chebyshev: 1.3485568760611182
```



```
Distancias para 5 centroides:
Centroides 1 y 2:
- Distancia Manhattan: 1.2465240641711228
- Distancia Euclidiana (Minkowski con p=2): 0.795033991595036
- Distancia Chebyshev: 0.5636363636363639
Centroides 1 y 3:
- Distancia Manhattan: 8.545170454545453
- Distancia Euclidiana (Minkowski con p=2): 5.243826190199232
- Distancia Chebyshev: 4.413541666666665
Centroides 1 y 4:
- Distancia Manhattan: 6.561862527716186
- Distancia Euclidiana (Minkowski con p=2): 3.9338369137179603
- Distancia Chebyshev: 3.373983739837398
Centroides 1 y 5:
- Distancia Manhattan: 4.821212121212121
- Distancia Euclidiana (Minkowski con p=2): 2.8542931564308547
- Distancia Chebyshev: 2.507407407407408
Centroides 2 y 3:
- Distancia Manhattan: 8.425919117647059
- Distancia Euclidiana (Minkowski con p=2): 5.004988823474803
- Distancia Chebyshev: 4.32922794117647
Centroides 2 y 4:
- Distancia Manhattan: 6.442611190817791
- Distancia Euclidiana (Minkowski con p=2): 3.7773736012828216
- Distancia Chebyshev: 3.2896700143472017
Centroides 2 y 5:
- Distancia Manhattan: 4.701960784313726
- Distancia Euclidiana (Minkowski con p=2): 2.858547175268822
- Distancia Chebyshev: 2.4230936819172118
Centroides 3 y 4:
- Distancia Manhattan: 2.4662347560975593
- Distancia Euclidiana (Minkowski con p=2): 1.3620686856028565
- Distancia Chebyshev: 1.0395579268292678
Centroides 3 y 5:
- Distancia Manhattan: 4.679513888888887
- Distancia Euclidiana (Minkowski con p=2): 2.5704140985054873
- Distancia Chebyshev: 1.9061342592592578
Centroides 4 y 5:
- Distancia Manhattan: 2.2132791327913277
- Distancia Euclidiana (Minkowski con p=2): 1.2121647476176276
- Distancia Chebyshev: 0.86657633242999
```



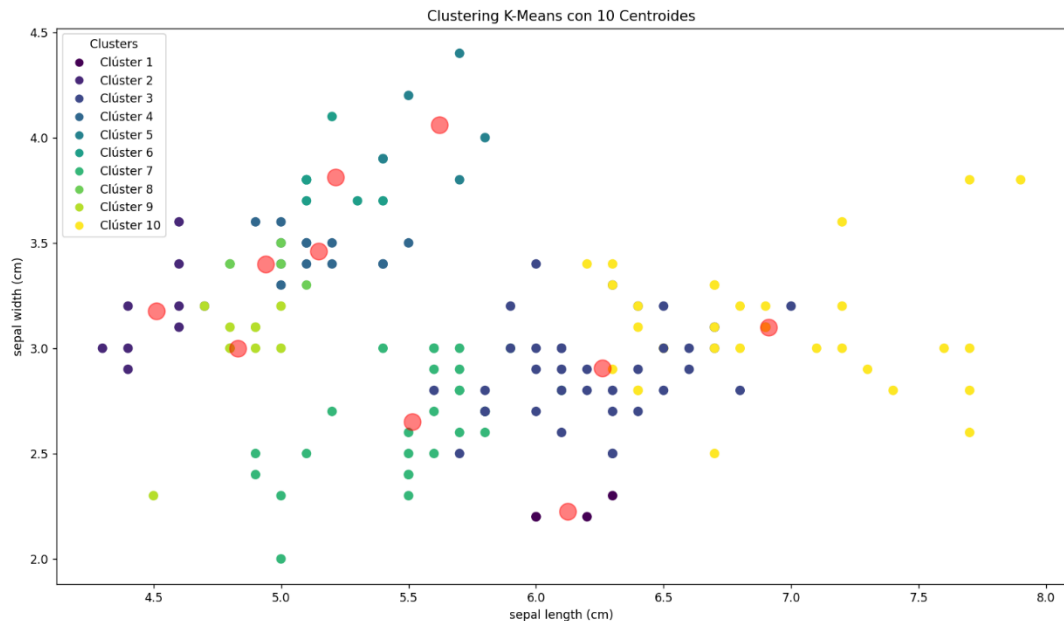
Distancias para 10 centroides:
Centroides 1 y 2:
- Distancia Manhattan: 6.866666666666667
- Distancia Euclidiana (Minkowski con p=2): 3.854694908745201
- Distancia Chebyshev: 3.175
Centroides 1 y 3:
- Distancia Manhattan: 1.485135135135134
- Distancia Euclidiana (Minkowski con p=2): 0.840493828544723
- Distancia Chebyshev: 0.6804054054054047
Centroides 1 y 4:
- Distancia Manhattan: 6.353846153846153
- Distancia Euclidiana (Minkowski con p=2): 3.5947138211019376
- Distancia Chebyshev: 3.036538461538461
Centroides 1 y 5:
- Distancia Manhattan: 6.420000000000002
- Distancia Euclidiana (Minkowski con p=2): 3.742445724389333
- Distancia Chebyshev: 3.0549999999999997
Centroides 1 y 6:
- Distancia Manhattan: 6.437499999999999
- Distancia Euclidiana (Minkowski con p=2): 3.5767259819561237
- Distancia Chebyshev: 2.8874999999999997
Centroides 1 y 7:
- Distancia Manhattan: 1.6407407407407413
- Distancia Euclidiana (Minkowski con p=2): 0.9100848631648953
- Distancia Chebyshev: 0.6101851851851858
Centroides 1 y 8:
- Distancia Manhattan: 6.100000000000005
- Distancia Euclidiana (Minkowski con p=2): 3.389675500693245
- Distancia Chebyshev: 2.795
Centroides 1 y 9:
- Distancia Manhattan: 6.22
- Distancia Euclidiana (Minkowski con p=2): 3.562849982888706
- Distancia Chebyshev: 3.0249999999999995
Centroides 1 y 10:
- Distancia Manhattan: 3.8406249999999984
- Distancia Euclidiana (Minkowski con p=2): 1.9793588174267436
- Distancia Chebyshev: 1.3718749999999993
Centroides 2 y 3:
- Distancia Manhattan: 6.9909909999999915
- Distancia Euclidiana (Minkowski con p=2): 4.20298199737792
- Distancia Chebyshev: 3.5297297297297288

Centroides 2 y 4:
- Distancia Manhattan: 1.0803418803418798
- Distancia Euclidiana (Minkowski con p=2): 0.7095798355299581
- Distancia Chebyshev: 0.6350427350427355
Centroides 2 y 5:
- Distancia Manhattan: 2.211111111111111
- Distancia Euclidiana (Minkowski con p=2): 1.4256053511697933
- Distancia Chebyshev: 1.1088888888888881
Centroides 2 y 6:
- Distancia Manhattan: 1.698611111111111
- Distancia Euclidiana (Minkowski con p=2): 0.9915139549393667
- Distancia Chebyshev: 0.7013888888888884
Centroides 2 y 7:
- Distancia Manhattan: 5.225925925925925
- Distancia Euclidiana (Minkowski con p=2): 3.0710136483793478
- Distancia Chebyshev: 2.659259259259259
Centroides 2 y 8:
- Distancia Manhattan: 1.211111111111111
- Distancia Euclidiana (Minkowski con p=2): 0.6404126755942043
- Distancia Chebyshev: 0.4288888888888884
Centroides 2 y 9:
- Distancia Manhattan: 0.6466666666666665
- Distancia Euclidiana (Minkowski con p=2): 0.394708831581452
- Distancia Chebyshev: 0.3188888888888889
Centroides 2 y 10:
- Distancia Manhattan: 8.957291666666668
- Distancia Euclidiana (Minkowski con p=2): 5.493315647814174
- Distancia Chebyshev: 4.546874999999999
Centroides 3 y 4:
- Distancia Manhattan: 6.478170478170477
- Distancia Euclidiana (Minkowski con p=2): 3.8805458915830573
- Distancia Chebyshev: 3.39126819126819
Centroides 3 y 5:
- Distancia Manhattan: 6.544324324324326
- Distancia Euclidiana (Minkowski con p=2): 3.8942653798623867
- Distancia Chebyshev: 3.4097297297297287
Centroides 3 y 6:
- Distancia Manhattan: 6.561824324324325
- Distancia Euclidiana (Minkowski con p=2): 3.780965392792883
- Distancia Chebyshev: 3.2422297297297287

Centroides 3 y 7:
- Distancia Manhattan: 2.272172172172171
- Distancia Euclidiana (Minkowski con p=2): 1.240692938892725
- Distancia Chebyshev: 0.8704704704704693
Centroides 3 y 8:
- Distancia Manhattan: 6.224324324324325
- Distancia Euclidiana (Minkowski con p=2): 3.6736027143457948
- Distancia Chebyshev: 3.149729729729729
Centroides 3 y 9:
- Distancia Manhattan: 6.344324324324324
- Distancia Euclidiana (Minkowski con p=2): 3.943529614434035
- Distancia Chebyshev: 3.3797297297297284
Centroides 3 y 10:
- Distancia Manhattan: 2.3554898648648646
- Distancia Euclidiana (Minkowski con p=2): 1.3189803934369502
- Distancia Chebyshev: 1.0171452702702704
Centroides 4 y 5:
- Distancia Manhattan: 1.1676923076923078
- Distancia Euclidiana (Minkowski con p=2): 0.7674271162124943
- Distancia Chebyshev: 0.5984615384615397
Centroides 4 y 6:
- Distancia Manhattan: 0.6182692307692301
- Distancia Euclidiana (Minkowski con p=2): 0.3904923591369987
- Distancia Chebyshev: 0.35096153846153877
Centroides 4 y 7:
- Distancia Manhattan: 4.713105413105411
- Distancia Euclidiana (Minkowski con p=2): 2.8590277329191527
- Distancia Chebyshev: 2.520797720797721
Centroides 4 y 8:
- Distancia Manhattan: 0.6661538461538461
- Distancia Euclidiana (Minkowski con p=2): 0.35951644085224291
- Distancia Chebyshev: 0.24153846153846126
Centroides 4 y 9:
- Distancia Manhattan: 0.8123076923076914
- Distancia Euclidiana (Minkowski con p=2): 0.560032754747909
- Distancia Chebyshev: 0.46153846153846034
Centroides 4 y 10:
- Distancia Manhattan: 8.444471153846152
- Distancia Euclidiana (Minkowski con p=2): 5.1308793138940905
- Distancia Chebyshev: 4.40841346153846

Centroides 5 y 6:
- Distancia Manhattan: 0.8475000000000006
- Distancia Euclidiana (Minkowski con p=2): 0.5059582492656882
- Distancia Chebyshev: 0.40749999999999975
Centroides 5 y 7:
- Distancia Manhattan: 4.989629629629629
- Distancia Euclidiana (Minkowski con p=2): 3.0528382080989047
- Distancia Chebyshev: 2.5392592592592593
Centroides 5 y 8:
- Distancia Manhattan: 1.6800000000000002
- Distancia Euclidiana (Minkowski con p=2): 0.9859006035092991
- Distancia Chebyshev: 0.6799999999999997
Centroides 5 y 9:
- Distancia Manhattan: 1.9799999999999999
- Distancia Euclidiana (Minkowski con p=2): 1.32612216631802
- Distancia Chebyshev: 1.06
Centroides 5 y 10:
- Distancia Manhattan: 8.510625000000001
- Distancia Euclidiana (Minkowski con p=2): 5.05399397290153
- Distancia Chebyshev: 4.426874999999999
Centroides 6 y 7:
- Distancia Manhattan: 4.796759259259258
- Distancia Euclidiana (Minkowski con p=2): 2.826527129813063
- Distancia Chebyshev: 2.3717592592592593
Centroides 6 y 8:
- Distancia Manhattan: 0.8824999999999994
- Distancia Euclidiana (Minkowski con p=2): 0.5138032210876063
- Distancia Chebyshev: 0.41249999999999964
Centroides 6 y 9:
- Distancia Manhattan: 1.4074999999999984
- Distancia Euclidiana (Minkowski con p=2): 0.9115885859311744
- Distancia Chebyshev: 0.8124999999999991
Centroides 6 y 10:
- Distancia Manhattan: 8.528125
- Distancia Euclidiana (Minkowski con p=2): 4.998559362768936
- Distancia Chebyshev: 4.259374999999999
Centroides 7 y 8:
- Distancia Manhattan: 4.459259259259259
- Distancia Euclidiana (Minkowski con p=2): 2.6114503588390083
- Distancia Chebyshev: 2.2702592592592595

Centroides 7 y 9:
- Distancia Manhattan: 4.579259259259259
- Distancia Euclidiana (Minkowski con p=2): 2.8217027323964725
- Distancia Chebyshev: 2.5092592592592595
Centroides 7 y 10:
- Distancia Manhattan: 4.627662037037036
- Distancia Euclidiana (Minkowski con p=2): 2.5528553910078755
- Distancia Chebyshev: 1.8876157407407397
Centroides 8 y 9:
- Distancia Manhattan: 0.9199999999999986
- Distancia Euclidiana (Minkowski con p=2): 0.5073460357586322
- Distancia Chebyshev: 0.39999999999999947
Centroides 8 y 10:
- Distancia Manhattan: 8.190625
- Distancia Euclidiana (Minkowski con p=2): 4.940696315108327
- Distancia Chebyshev: 4.166874999999999
Centroides 9 y 10:
- Distancia Manhattan: 8.510624999999997
- Distancia Euclidiana (Minkowski con p=2): 5.23536460794518
- Distancia Chebyshev: 4.396874999999999



Conclusiones

Al concluir la práctica de agrupamiento utilizando el algoritmo K-means en Python, se ha evidenciado que este método es una herramienta efectiva para resolver problemas de clasificación no supervisada. A lo largo del proceso, se enfatizó la importancia de seleccionar adecuadamente el número de clusters, ya que este parámetro impacta directamente en la capacidad del modelo para representar la estructura subyacente de los datos, algo que pudimos visualizar al momento de hacer 3 casos distintos (3, 5 y 10 clústeres).

Durante el desarrollo del modelo, se destacó la relevancia de la preparación de los datos, como la correcta normalización y selección de características, que influye en el rendimiento de K-means. Además, el cálculo de distancias entre los centroides permitió una comprensión más profunda de las relaciones entre los distintos grupos, lo cual es esencial para la interpretación de los resultados.

El uso de visualizaciones gráficas facilitó la identificación de los clústeres y sus centroides, ofreciendo una representación clara de cómo se agrupan los datos en el espacio de características. Esta representación visual fue particularmente útil para observar la distribución de las distintas especies en el conjunto de datos de Iris y cómo K-means las clasifica.

Asimismo, el análisis de las distancias entre los centroides proporcionó información valiosa sobre la separabilidad de los clústeres y permitió evaluar la efectividad del modelo. Las métricas utilizadas para cuantificar estas distancias, como las de Manhattan, Minkowski y Chebyshev, ofrecieron diferentes perspectivas sobre la estructura de los datos y la relación entre los clústeres.

En resumen, esta práctica ha demostrado que K-means es un enfoque robusto para la agrupación de datos en un contexto no supervisado, enfatizando la importancia de la elección adecuada de hiperparámetros y la visualización de resultados. La experiencia adquirida ha sido fundamental para comprender mejor los fundamentos del aprendizaje no supervisado, proporcionando una base sólida para futuros proyectos en el ámbito de la ciencia de datos y el aprendizaje automático. Herramientas como NumPy y Matplotlib han sido cruciales en el desarrollo de modelos, subrayando su utilidad en el análisis y visualización de datos.

Referencias

- Ramírez, L. (2023, 5 de enero). *Algoritmo k-means: ¿Qué es y cómo funciona?*. IEBS School. Recuperado el 31 de Octubre de 2024 de: <https://www.iebschool.com/blog/algoritmo-k-means-que-es-y-como-funciona-big-data/>