**ALGORITMO DE MACHINE LEARNING CLUSTERING-KMEANS**

Jose Daniel Velasquez-1088349647

Inteligencia artificial.

El clustering es una agrupacion de datos automatica y no es supervisado. Las técnicas de clustering intentan descubrir cuál es el mejor agrupamiento de los datos. Algunas de estas técnicas necesitan que especifiquemos el número de grupos de queremos encontrar.

Explicaresmos uno de los tantos algoritmos de clustering que existen, el cual es Kmeans ya que es uno de los mas usados. A este algoritmo toca indicarle el numero de grupos a encontrar, a esto se le denomina K.

El algoritmo K-Means sigue los siguientes pasos:

* Inicialización: se elige la localización de los centroides de los K grupos aleatoriamente.
* Asignación: se asigna cada dato al centroide más cercano.
* Actualización: se actualiza la posición del centroide a la media aritmética de las posiciones de los datos asignados al grupo.

Los pasos 2 y 3 se siguen iterativamente hasta que no haya más cambios.

El algoritmo sera ejecutado en python y se utilizaran las librerias scikit-learn, pandas, matplotlib y numpy.

El ejemplo que utilizaremos se saco del articulo “K-Means en Python paso a paso”[2] .

El conjunto de datos consta de un proyecto en el cual se analizan rasgos de la personalidad de usuarios de Twitter famosos. En total son 140 y tiene las siguientes caracteristicas de entrada.

* usuario (el nombre en Twitter)
* «op» = Openness to experience – grado de apertura mental a nuevas experiencias, curiosidad, arte
* «co» =Conscientiousness – grado de orden, prolijidad, organización
* «ex» = Extraversion – grado de timidez, solitario o participación ante el grupo social
* «ag» = Agreeableness – grado de empatía con los demás, temperamento
* «ne» = Neuroticism, – grado de neuroticismo, nervioso, irritabilidad, seguridad en sí mismo.
* Wordcount – Cantidad promedio de palabras usadas en sus tweets
* Categoria – Actividad laboral del usuario (actor, cantante, etc.)

Se agruparan por sus similitudes de personalidad y solo se utilizara 3 columnas de las 8 como entradas.

**IMPLEMENTACION**

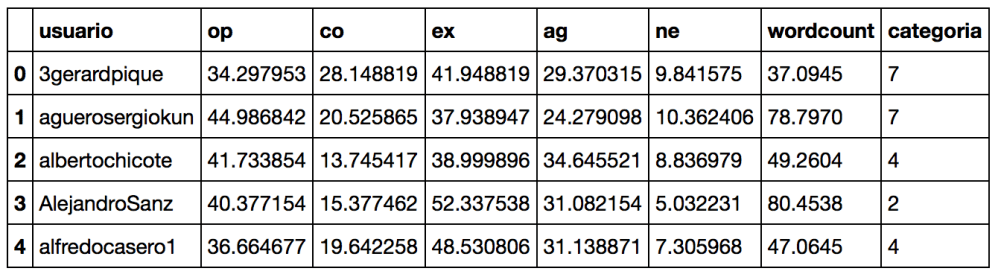
Se comensara importando las librerias necesarias para el ejercicio.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11 | import pandas as pd  import numpy as np  import matplotlib.pyplot as plt  import seaborn as sb  from sklearn.cluster import KMeans  from sklearn.metrics import pairwise\_distances\_argmin\_min    %matplotlib inline  from mpl\_toolkits.mplot3d import Axes3D  plt.rcParams['figure.figsize'] = (16, 9)  plt.style.use('ggplot') |

Se importa el archivo csv donde estan los datos.

|  |  |
| --- | --- |
| 2 | 1 dataframe = pd.read\_csv(r"analisis.csv")  dataframe.head() |

El cual arroja tambien esta tabla que describe los datos.



Las categorias son:

1. Actor/actriz
2. Cantante
3. Modelo
4. Tv, series
5. Radio
6. Tecnología
7. Deportes
8. Politica
9. Escritor

Para saber cuantos registros tenemos de cada uno hacemos.

|  |  |
| --- | --- |
| 1 | print(dataframe.groupby('categoria').size()) |

Lo cual nos suelta.



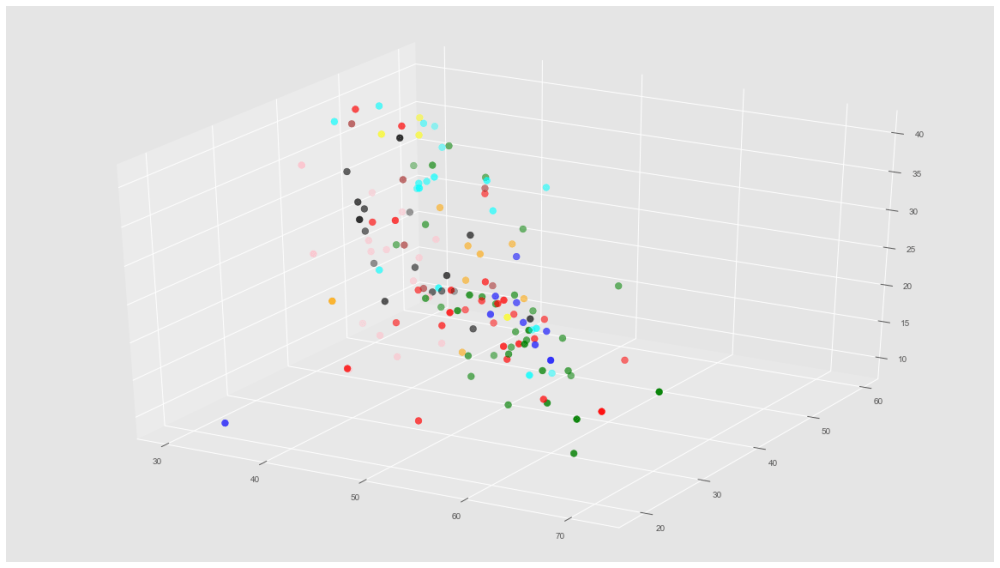
**Definir entradas.**

|  |  |
| --- | --- |
| 1  2 | X = np.array(dataframe[["op","ex","ag"]])  y = np.array(dataframe['categoria']) |

como se puede ver en la variable x solo se cargaron las 3 columnas op,ex,ag.

Ahora se vera una gráfica en 3D con 9 colores representando las categorías.

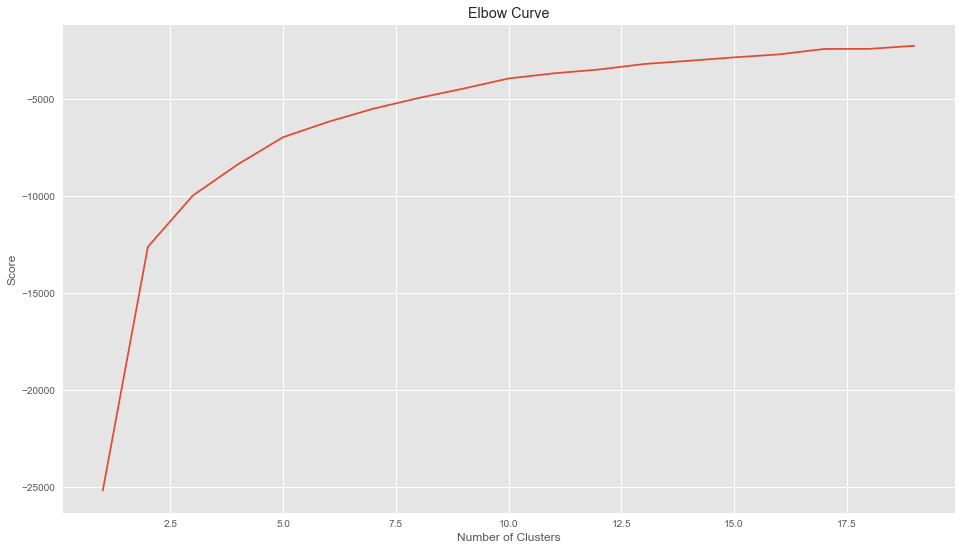
|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | fig = plt.figure()  ax = Axes3D(fig)  colores=['blue','red','green','blue','cyan','yellow','orange','black','pink','brown','purple']  asignar=[]  for row in y:      asignar.append(colores[row])  ax.scatter(X[:, 0], X[:, 1], X[:, 2], c=asignar,s=60) |



Con Kmeans se puede trazar esta grafica de forma diferente.

Pero primero debemos obtener el valor de K, por medio de una grafica y hallando el punto codo.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10 | Nc = range(1, 20)  kmeans = [KMeans(n\_clusters=i) for i in Nc]  kmeans  score = [kmeans[i].fit(X).score(X) for i in range(len(kmeans))]  score  plt.plot(Nc,score)  plt.xlabel('Number of Clusters')  plt.ylabel('Score')  plt.title('Elbow Curve')  plt.show() |
| 1  2  3  4  5  6  7  8  9  10 | Nc = range(1, 20)  kmeans = [KMeans(n\_clusters=i) for i in Nc]  kmeans  score = [kmeans[i].fit(X).score(X) for i in range(len(kmeans))]  score  plt.plot(Nc,score)  plt.xlabel('Number of Clusters')  plt.ylabel('Score')  plt.title('Elbow Curve')  plt.show() |

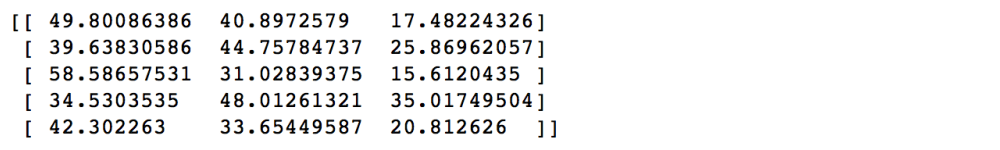


Para este caso según la grafica se utilizo el numero 5 como numero de K.

**Ejecucion algoritmo.**

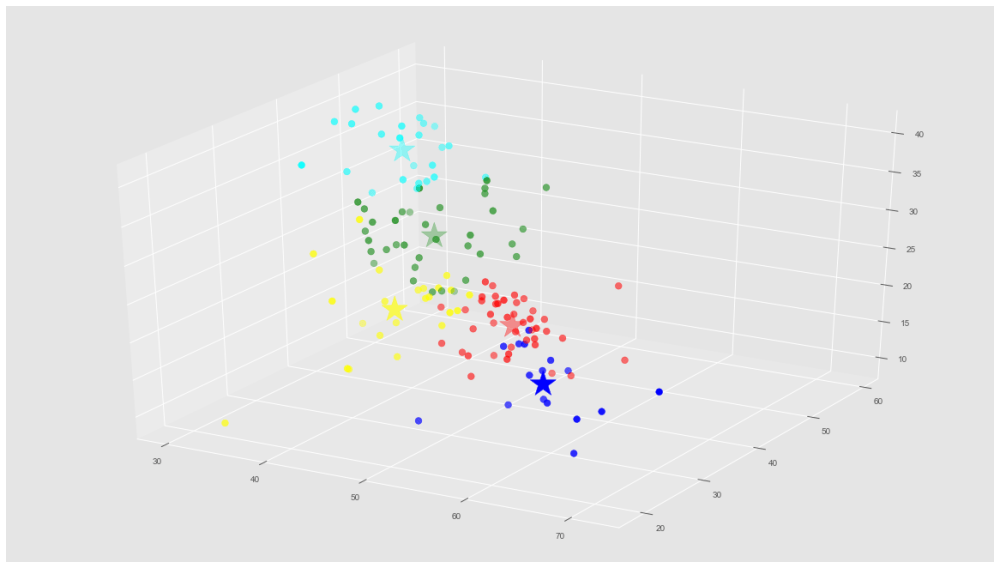
Se ejecuta el algoritmo para 5 clusters y se obtienen as etiquetas y los centroids.

|  |  |
| --- | --- |
| 1  2  3 | kmeans = KMeans(n\_clusters=5).fit(X)  centroids = kmeans.cluster\_centers\_  print(centroids) |



Ahora se vera en una grafica 3D con solores para los grupos. Las estrellas marcan el centro de cada cluster.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13 | # Predicting the clusters  labels = kmeans.predict(X)  # Getting the cluster centers  C = kmeans.cluster\_centers\_  colores=['red','green','blue','cyan','yellow']  asignar=[]  for row in labels:      asignar.append(colores[row])    fig = plt.figure()  ax = Axes3D(fig)  ax.scatter(X[:, 0], X[:, 1], X[:, 2], c=asignar,s=60)  ax.scatter(C[:, 0], C[:, 1], C[:, 2], marker='\*', c=colores, s=1000) |



Aquí se puede apreciar que el algoritmo de K-means con k=5 ha agrupado a los 140 usuarios de twitter del conjunto de datos por su personalidad, teniendo encuenta las 3 dimensiones que usamos, que son, Openess, Extraversion y Agreeablenes. En el grafico se puede ver que no hay necesariamente una relacion en los grupos con sus actividades de celebrity.

**BIBLIOGRAFIA.**

**[1]** Articulo “Clustering (Agrupamiento), K-Means con ejemplos en python”, iartificial.net/clustering-agrupamiento-kmeans-ejemplos-en-python/

**[2]** Articulo “K-Means en Python paso a paso” www.aprendemachinelearning.com/k-means-en-python-paso-a-paso/