



Quintanar Alvarado José Manuel

Mtro. Eduardo Antonio Hinojosa Palafox

Minería de Datos

Instituto Tecnológico de Hermosillo

Proyecto Final – Dataset extinción del Hielo en el Ártico

7mo Semestre

13/12/2022

INTRODUCCIÓN.

En este proyecto investigué varios Datasets interesantes y también que fueran recientes y actualizables, encontré muchos, pero ninguno me convencía, hasta que encontré uno realmente interesante y que muchas personas deberían ver. Los datos obtenidos aquí a la mayor parte de las personas les puede gustar.

Utilicé el modelo Kmeans para agrupar los datos y así dividirlos por clústeres para permitirnos ver como es que se comportan los datos y el resultado es notorio, cada año fue disminuyendo y otro año creciendo. Es fascinante ver como se relacionan los datos entre si para formar una agrupación del cual podemos visualizarlos mediante clústeres.

DESCRIPCION DEL PROBLEMA A DESARROLLAR.

El principal problema es la gravedad en la que conlleva estos datos de acuerdo a los años, nos muestran que no ha habido mejoras mediante el tiempo transcurrido.

Para eso es este proyecto, realizará una visualización detallada mediante grupos de datos cuyos datos puedan ser entrenados y puestos a prueba para la graficación de la misma.

Puede sonar complejo el hecho de que cualquier usuario nuevo o no experimentado con este algoritmo llamado KMeans, pero estoy seguro que entenderá cada paso a continuación del proyecto.

DESCRIPCIÓN DEL CONJUNTO DE DATOS

Este Dataset presenta datos de la extinción del Hielo en el ártico mediante los años, dando por entendido que se irá actualizando conforme pase el tiempo (actualizaciones por año).

Datos que usaremos:

“year” = que será el que se encargue de mostrarnos los años usados en la tabla.

“extinto” = Porcentaje que muestra el hielo extinto en el ártico.

DESCRIPCION DE LA SOLUCIÓN PROPUESTA.

Dividiremos en 2 grupos (clusters) que nos mostrará que grupo de datos fue en donde se extinguió más el hielo, estos grupos estarían formados de tantos años a tantos años.

Dichos grupos tendrán datos como el Año y el Porcentaje de Hielo que fue extinto, lo visualizaremos mediante una `plt.show` para captar el comportamiento de los datos.

DESCRIPCIÓN DEL CÓDIGO UTILIZADO.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from sklearn.preprocessing import MinMaxScaler

df = pd.read_csv('arctic_ice_extent.csv')
df.head()
```

Daremos uso de 5 librerías, de las cuales explicaré a continuación para que sirve cada una. A excepción de pandas y numpy que esas son las librerías más básicas en las que se trabaja colab.

Import matplotlib.pyplot as plt: Es una biblioteca para la generación de gráficos en dos dimensiones, a partir de datos contenidos en listas o arrays en el lenguaje de programación Python. Asimismo, la nombramos como plt (as plt) cada que queremos dar uso de esa biblioteca.

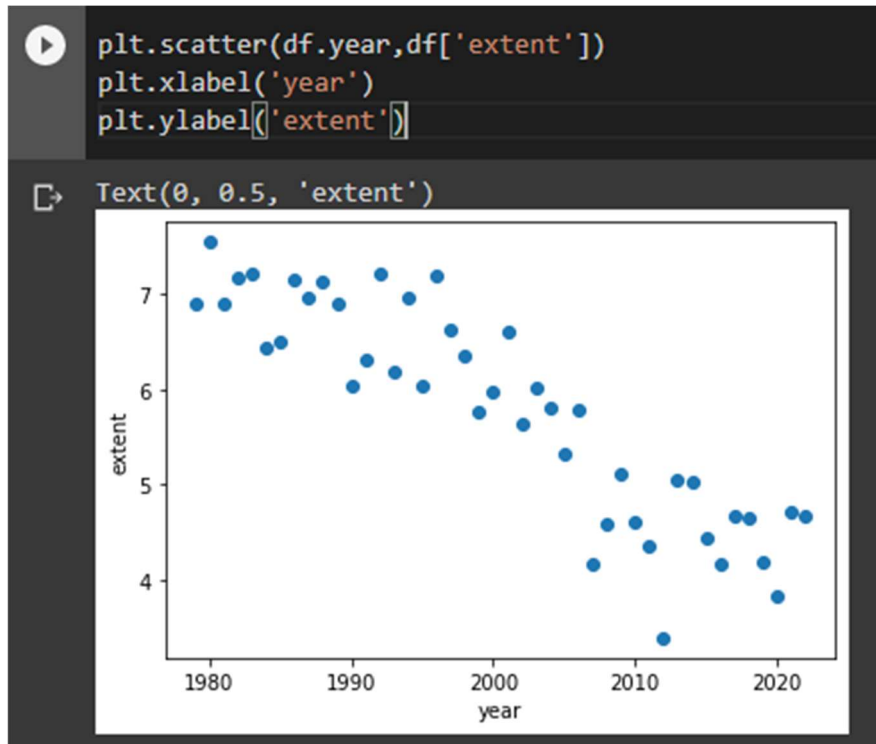
From sklearn.cluster import Kmeans: Es un algoritmo de clasificación no supervisada (clusterización) que agrupa objetos en k grupos basándose en sus características. Y de esta forma añadimos la librería Kmeans para dar uso de ella en Python Colab.

From sklearn.preprocessing import MinMaxScaler:

Esta librería se encarga de transformar las características escalando cada característica a un rango dado. Más adelante mostraré ya una vez implementado para que se entienda mejor.

df= pd.Read_csv('artic_ice_extent.csv'): Y esta ultima de aquí nos sirve para leer nuestro conjunto de datos seleccionado para realizar este proyecto (Dataset). Leyendo cada dato que se encuentra y conociéndolo más a fondo.

RESULTADO.



Como se puede apreciar, `plt.scatter` es utilizado para graficar los datos del Dataset. Esta imagen representa los años y el porcentaje de extinción del Hielo, graficando los 43 puntos del Dataset que existen dentro de ella. Dividiéndolo por año. Si se fijan bien, ningún punto está más arriba de otro en el mismo año, esto es porque solo un dato de extinción pertenece al año y no dos.


```
#Ejecutamos nuestro modelo KMeans
km = KMeans(n_clusters=2)
km


KMeans(n_clusters=2)

[23]
y_pred = km.fit_predict(df[['year', 'extent']])
y_pred

array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
      dtype=int32)
```

En `Km = Kmeans(n_clusters=2)` Aquí lo que hacemos es emplear nuestro modelo Kmeans mediante la librería que implementamos en un principio. Lo igualamos a `km` y le proporcionamos el número de clusters que se vayan a usar, en este caso son 2.

Asimismo, sacamos la predicción mediante un método de Kmeans proporcionado mediante la librería de igual manera. `Km.fit_predict` en el que sacará los valores de los campos del Dataset que son Año y extinción. Posteriormente imprimimos nuestra variable creada y nos dice que los datos del Dataset han sido divididos en 2 grupos que son 0 y 1.

<div>  df['Cluster'] = y_pred df </div>				22	2001	6.603	0
				23	2002	5.638	0
				24	2003	6.007	0
				25	2004	5.794	0
				26	2005	5.319	0
				27	2006	5.774	0
				28	2007	4.155	0
				29	2008	4.586	0
				30	2009	5.119	0
				31	2010	4.615	0
				32	2011	4.344	0
				33	2012	3.387	0
				34	2013	5.054	0
				35	2014	5.029	0
				36	2015	4.433	0
				37	2016	4.165	0
				38	2017	4.665	0
				39	2018	4.656	0
				40	2019	4.192	0
				41	2020	3.818	0
				42	2021	4.720	0
				43	2022	4.670	0

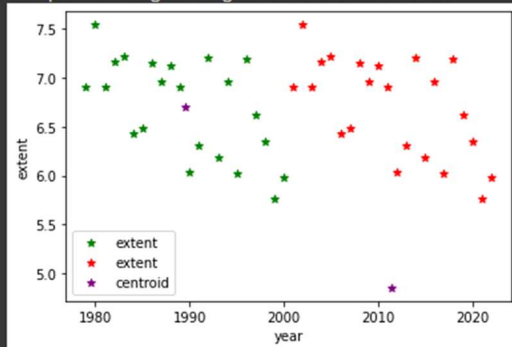
Aquí se muestra lo que comentaba en el apartado anterior, crearemos un Dataframe con el nombre 'Cluster' (En realidad puede ser cualquier nombre, pero para que se entienda mejor opté por este), y lo igualamos a la variable creada anteriormente que es y_pred. Y una vez impreso el df nos saldrán todos los datos con su grupo establecido.

```

df1 = df[df.Cluster == 0]
df2 = df[df.Cluster == 1]
plt.scatter(df1.year, df1['extent'], color='green', marker='*', label='extent')
plt.scatter(df2.year, df2['extent'], color='red', marker='*', label='extent')
plt.scatter(km.cluster_centers[:,0], km.cluster_centers[:,1], color='purple', marker='*', label='centroid')
plt.xlabel('year')
plt.ylabel('extent')
plt.legend()

```

<matplotlib.legend.Legend at 0x7f249a534c40>



En esta parte de aquí ya comienza a visualizarme mucho mejor la manera en que se comportan los datos. Si se fijan bien, creamos 2 variables que contendrán los clusters creados anteriormente, que son 1 y 0. Posteriormente a esto, Agregaremos las características necesarias para la creación de nuestro plt.scatter, como lo son el color, marker, label y la más importante que datos quieren que esté visualizándose, en este caso serán los datos df's recién creados.

Y por último agregamos las etiquetas Y y X para identificar que dato es el que estamos usando. Etiqueta X para Año y etiqueta Y para extinto.

```
df.groupby('Cluster').mean()
```

	year	extent
Cluster		
0	2011.5	4.851955
1	1989.5	6.701409

Y una vez que hayamos hecho todo lo demás, daremos uso del `mean()` para verificar los grupos y como actúan. Podemos ver que el grupo 1 que es de 1989.5 el hielo que se extinguió más contiene mayor número que el grupo 0. Dándonos entender que se extinguió más hielo en los años que pertenecen al grupo 1 que los que pertenecen al grupo 0.

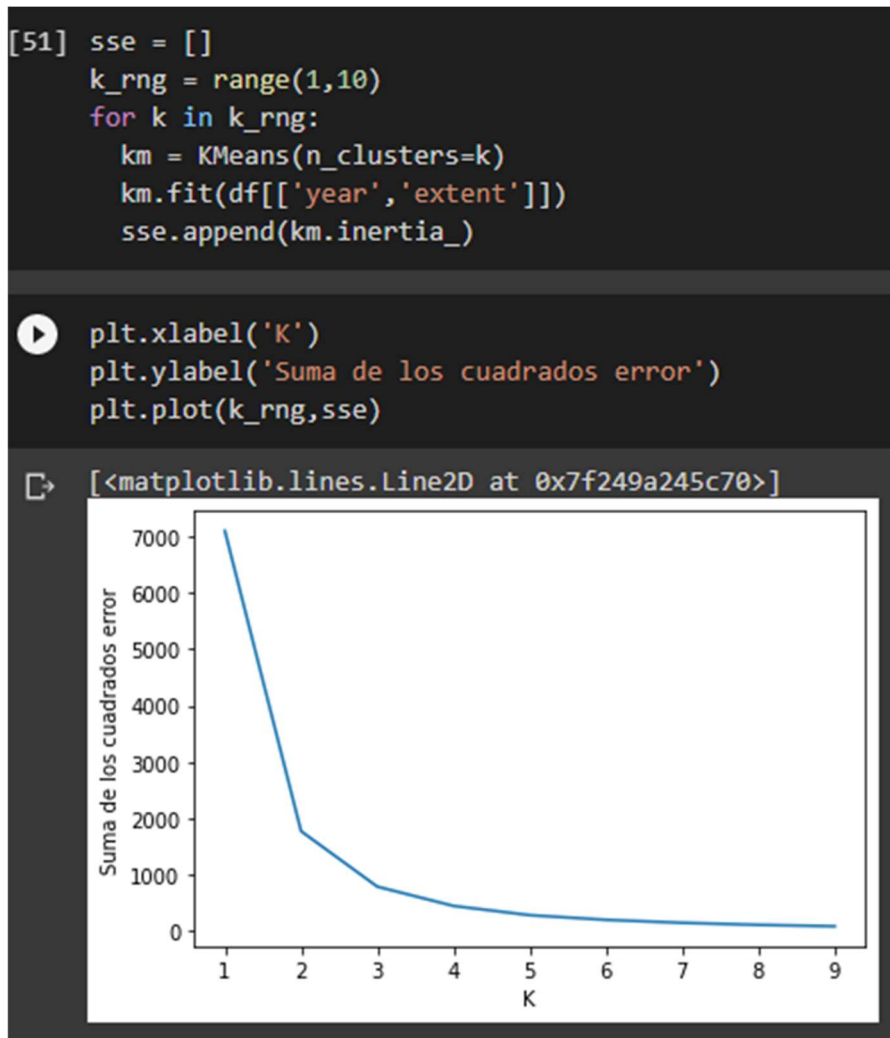
```
scaler = MinMaxScaler()
scaler.fit(df[['extent']])
df['extent'] = scaler.transform(df[['extent']])
```

```
[39] df.head()
```

	year	extent	Cluster
0	1979	0.845802	1
1	1980	1.000000	1
2	1981	0.845802	1
3	1982	0.909069	1
4	1983	0.922300	1

Aquí escalamos nuestros datos que emplearemos para transformar, en este caso usaremos los de extent ya que el de los años, como solo nos dice los años no será necesario entrenar esos datos.

Una vez entrenados los datos extent, mandaremos a imprimir los primeros 5 para visualizar resultados.



Y, para finalizar, metemos también el método del codo.

Ocuparemos sacar el valor K. Le asignaremos un rango de datos, en este caso es de 1-10. Asimismo también obtendremos la suma de las distancias al cuadrado aplicando la `km.inertia_`, que es esencial si daremos uso del método del codo. Entrenamos dichos datos y graficamos, quedándonos de esta forma.

CONCLUSIONES.

Este proyecto, sinceramente, cuando vi que tenía que hacer algo parecido a las clases dadas por el maestro, supe que iba a estar interesante debido a que el comportamiento de los datos no siempre será igual, independientemente si empleas el mismo algoritmo. En la realización de este proyecto noté que hay varias formas de emplear el algoritmo y no exactamente tiene que ser el mismo código que el de todos, por ahí vi unos que añadían características a las gráficas, otros que eran exactamente y no necesariamente era especificar las características de la gráfica.

Me gustó mucho como fueron saliendo los datos, verdaderamente fue un proyecto interesante y algún día, en un futuro, me gustaría volver para ver como yo realizaba mis proyectos y seguir mejorando en ello.

REFERENCIAS.

- 1) <https://stackoverflow.com/questions/64185216/how-can-i-make-a-matplotlib-plot-in-google-colab-interactive>
- 2) <https://towardsdatascience.com/machine-learning-algorithms-part-9-k-means-example-in-python-f2ad05ed5203>
- 3) <https://medium.datadriveninvestor.com/k-means-clustering-c92463d5fa0e>