Tecnológico de Monterrey

Herramientas computacionales: el arte de la analítica

TC1002S.201

Dr. Sergio Ruiz Loza

Actividad Evaluable: Patrones con K-means

Juan Ernesto Díaz Noguez - A01653546
Mauricio Hernández Matías - A01651328
Sara Dayana Camargo Márquez - A01652414

28 de octubre de 2020

Actividad Evaluable: Patrones con K-means

1. Cargar los datos utilizando la librería pandas e importar las librerías, numpy, matplotlib, seaborn debido a que las utilizaremos a lo largo del ejercicio

```
In [66]: 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')
```

Figura 1.

2. Si determinas que alguna variable no sirve basándose en la actividad pasada, elimínala y justifica por qué quitaste o no variables.
    - Se eliminaron las variables ‘Date’, ‘4046’, ‘4225’, ‘4770’. La variable ‘Date’ fue eliminada debido a la complejidad que su estructura de “dd/mm/yy”, además de que se consideró redundante ya que los datos ‘year’ y ‘Unnamed: 0’ nos proporcionan información similar pero esta los datos de la última variable son más fáciles de procesar. En cuanto a las variables ‘4046’, ‘4225’, ‘4770’ no contamos con el contexto suficiente para poder interpretarlas y relacionarlas a alguna otra variable por lo que decidimos omitirlas.

    Datos antes de filtrar y eliminar las variables mencionadas:

Unnamed: 0	AveragePrice	Total Volume	4046	4225	4770	Total Bags	Small Bags	Large Bags	XLarge Bags	year
18249.000000	18249.000000	1.824900e+04	1.824900e+04	1.824900e+04	1.824900e+04	1.824900e+04	1.824900e+04	1.824900e+04	18249.000000	18249.000000
24.232232	1.405978	8.506440e+05	2.930084e+05	2.951546e+05	2.283974e+04	2.396392e+05	1.821947e+05	5.433809e+04	3106.426507	2016.147899
15.481045	0.402677	3.453545e+06	1.264989e+06	1.204120e+06	1.074641e+05	9.862424e+05	7.461785e+05	2.439660e+05	17692.894652	0.939938
0.000000	0.440000	8.456000e+01	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000	2015.000000
10.000000	1.100000	1.083858e+04	8.540700e+02	3.008780e+03	0.000000e+00	5.088640e+03	2.849420e+03	1.274700e+02	0.000000	2015.000000
24.000000	1.370000	1.073768e+05	8.645300e+03	2.906102e+04	1.849900e+02	3.974383e+04	2.636282e+04	2.647710e+03	0.000000	2016.000000
38.000000	1.660000	4.329623e+05	1.110202e+05	1.502069e+05	6.243420e+03	1.107834e+05	8.333767e+04	2.202925e+04	132.500000	2017.000000
52.000000	3.250000	6.250565e+07	2.274362e+07	2.047057e+07	2.546439e+06	1.937313e+07	1.338459e+07	5.719097e+06	551693.650000	2018.000000

Figura 2.

Datos después de filtrar y eliminar las variables mencionadas:

	Unnamed: 0	AveragePrice	Total Volume	Total Bags	Small Bags	Large Bags	XLarge Bags	year
count	18249.000000	18249.000000	1.824900e+04	1.824900e+04	1.824900e+04	1.824900e+04	18249.000000	18249.000000
mean	24.232232	1.405978	8.506440e+05	2.396392e+05	1.821947e+05	5.433809e+04	3106.426507	2016.147899
std	15.481045	0.402677	3.453545e+06	9.862424e+05	7.461785e+05	2.439660e+05	17692.894652	0.939938
min	0.000000	0.440000	8.456000e+01	0.000000e+00	0.000000e+00	0.000000e+00	0.000000	2015.000000
25%	10.000000	1.100000	1.083858e+04	5.088640e+03	2.849420e+03	1.274700e+02	0.000000	2015.000000
50%	24.000000	1.370000	1.073768e+05	3.974383e+04	2.636282e+04	2.647710e+03	0.000000	2016.000000
75%	38.000000	1.660000	4.329623e+05	1.107834e+05	8.333767e+04	2.202925e+04	132.500000	2017.000000
max	52.000000	3.250000	6.250565e+07	1.937313e+07	1.338459e+07	5.719097e+06	551693.650000	2018.000000

Figura 3.

3. Determina un valor de k.
    - Definimos los valores que utilizaremos para el analisis

```
In [160]: #Para el ejercicio, sólo seleccionamos 3 dimensiones, para poder graficarlo
          X = np.array(dataframe[["Small Bags","Large Bags","XLarge Bags"]])
          y = np.array(dataframe['Unnamed: 0'])
          X.shape
          y

Out[160]: array([ 0,  1,  2, ...,  9, 10, 11], dtype=int64)
```

Figura 4.

- Para identificar el valor de K creamos una gráfica con la cual encontraremos el "punto de codo"

Sacando k para 'Unnamed: 0'

```
In [132]: 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()
```

Figura 5.

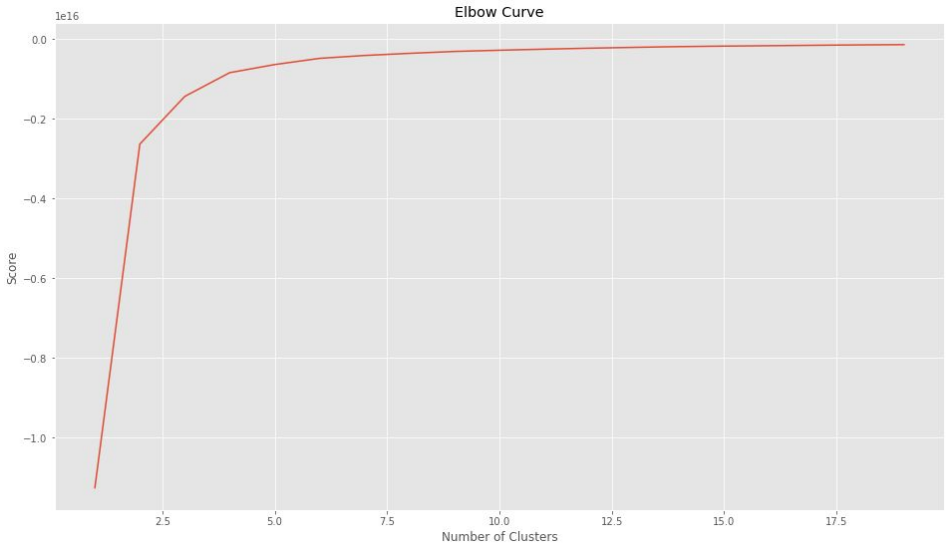

Figura 6.

- La curva comienza a suavizarse en 2 por lo que será nuestro valor para K será 2

4. Utilizando scikitlearn calcula los centros del algoritmo k-means.

```
In [162]: #Por lo tanto k=2
          kmeans = KMeans(n_clusters=2).fit(X)
          centroids = kmeans.cluster_centers_
          print(centroids)

[[1.18842119e+05 3.56533947e+04 1.95282475e+03]
 [7.04172239e+06 2.07743166e+06 1.28013175e+05]]
```

Figura 7.

Basado en los centros responde las siguientes preguntas:

- ¿Crees que estos centros puedan ser representativos de los datos? ¿Por qué?
  Si los considero representativos, el principio de operación del algoritmo k-means garantiza que los centros sean generados en base a las características principales de los datos
- ¿Cómo obtuviste el valor de k a usar?

Al analizar la gráfica:

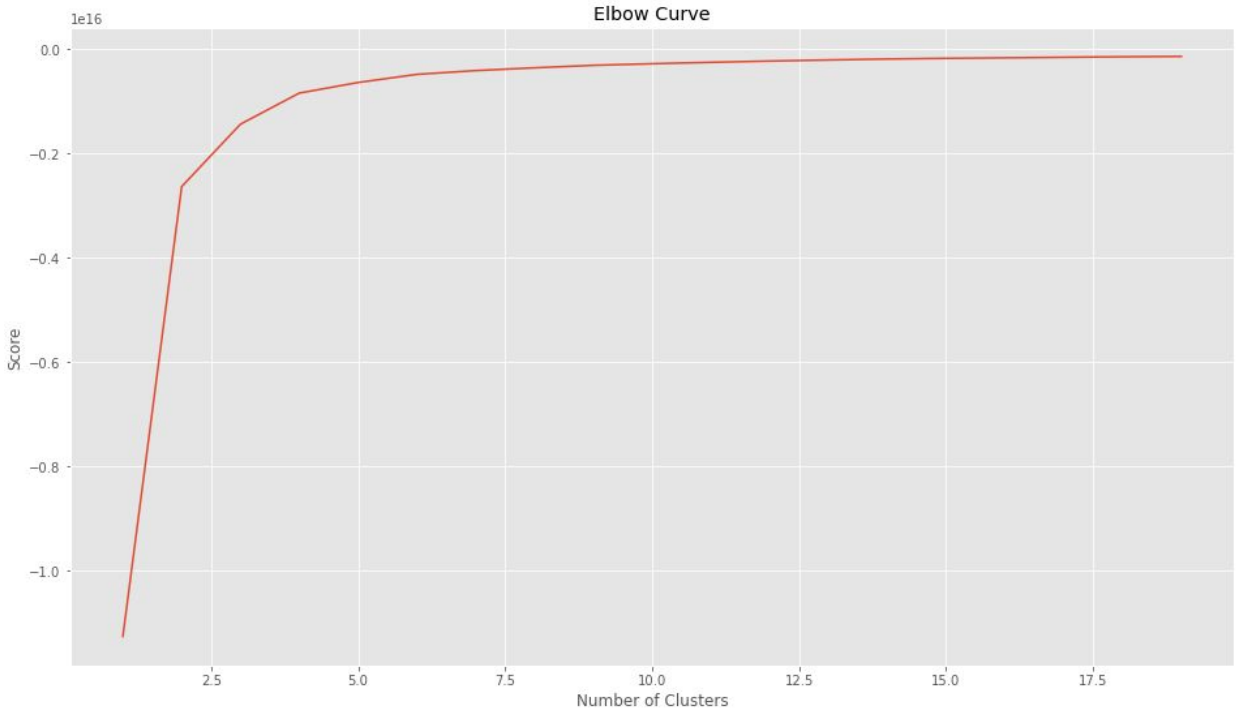

Figura 8.

Nos dimos cuenta que los valores en los que la curva comienza a suavizarse son 1 y 2, sin embargo con la finalidad de obtener diversos centros nuestro equipo decidió tomar K=2

- ¿Los centros serían más representativos si usaras un valor más alto? ¿Más bajo?

  Los centros podrían ser más representativos si se usara un valor más pequeño, pues la distancia media a los centroides sería igual más pequeña, garantizando que los grupos tengan más características en común.

- ¿Qué distancia tienen los centros entre sí? ¿Hay alguno que esté muy cercano a otros?
  De los dos centros ninguno es cercano al otro, en cuanto a la distancia que existe entre los centroides utilizando parte del algoritmo proporcionado por el profesor para obtener K-means

```
def distance2(a, b):
    dx = b[0] - a[0]
    dy = b[1] - a[1]
    dx2 = dx * dx
    dy2 = dy * dy
    return dx2 + dy2
```

  donde a y b son las coordenadas de los centroides:

  a= [1.18842119e+05, 3.56533947e+04]
  b= [7.04172239e+06, 2.07743166e+06]

  El valor de distancia que la función regresa es: **52,095,129,731,252.516** . Esto es la suma de los cuadrados de las distancia horizontales y verticales de los centroides, es decir, es el cuadrado de la distancia de los centroides.

- ¿Qué pasaría con los centros si tuviéramos muchos outliers en el análisis de cajas y bigotes?

  Como la media de los datos se vería afectada, los outliers causarían que la gráfica de cajas y bigotes se viera una pequeña barra horizontal representando los cuartiles mientras que los outliers estarían por encima de de la caja y en los extremos de los bigotes. Esto provocaría que la media se viera afectada y al momento de hacer los clusters usando el algoritmo, el resultado de la media de los clusters cambiaría y sería más fácil que converjan los clusters a comparación si los datos fueron limpiados y no se encontrarán outliers.

- ¿Qué puedes decir de los datos basándose en los centros?

Existe una alta concentración de datos en el centro más cercano al origen mientras que los datos relacionados al segundo centro son dispersos, por lo que podemos decir que no tienen altos niveles de correlación y el valor de la desviación estándar será alto. Una limpieza de datos y ser proporcionados con información relacionada al contexto de los datos sería necesaria para poder llevar a cabo un análisis donde puedan vincularse variables y encontrar una posible correlación entre estas.