UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN
FACULTAD DE CIENCIAS FÍSICO MATEMÁTICAS
MAESTRÍA EN CIENCIA DE DATOS

APRENDIZAJE AUTOMÁTICO

Profesor: Jose Anastacio Hernandez Saldaña

Tarea 4: Agrupamiento

Dora Alicia Guevara Villalpando
Matrícula: 1551003
Grupo: 003

21 de julio de 2024

UANL
UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

FCFM
FACULTAD DE CIENCIAS FÍSICO MATEMÁTICAS

Tarea 4: Agrupamiento

Introducción.

El presente documento presenta los resultados obtenidos durante el proceso de realizar la Tarea 3 de Aprendizaje Automático. Esta tarea consiste en realizar el modelo de clasificación a una base de datos de nuestro interés.

Contexto.

Este conjunto de datos contiene detalles médicos de los pacientes, incluidas características como el nivel de glucosa, la presión arterial, el nivel de insulina, el IMC, la edad y más. La variable objetivo indica si un paciente tiene diabetes. El objetivo de este conjunto de datos es crear y evaluar varios modelos de aprendizaje automático o aprendizaje profundo para predecir la aparición de la diabetes.

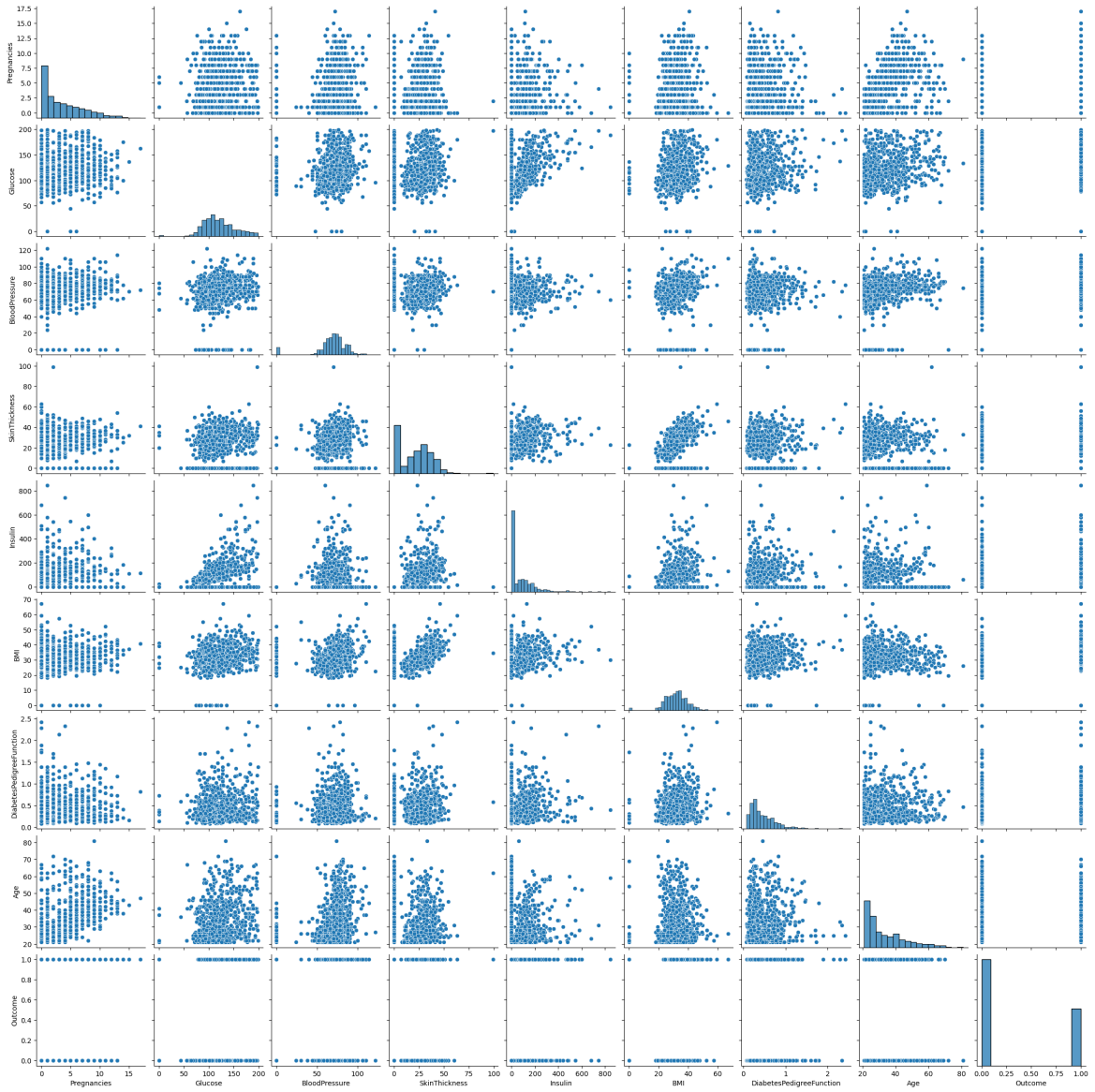
Este archivo contiene los registros médicos de los pacientes, que incluyen diversas métricas relacionadas con la salud. El objetivo es utilizar estas características para predecir si un paciente tiene diabetes. A continuación, se incluye una descripción detallada de cada columna del conjunto de datos:

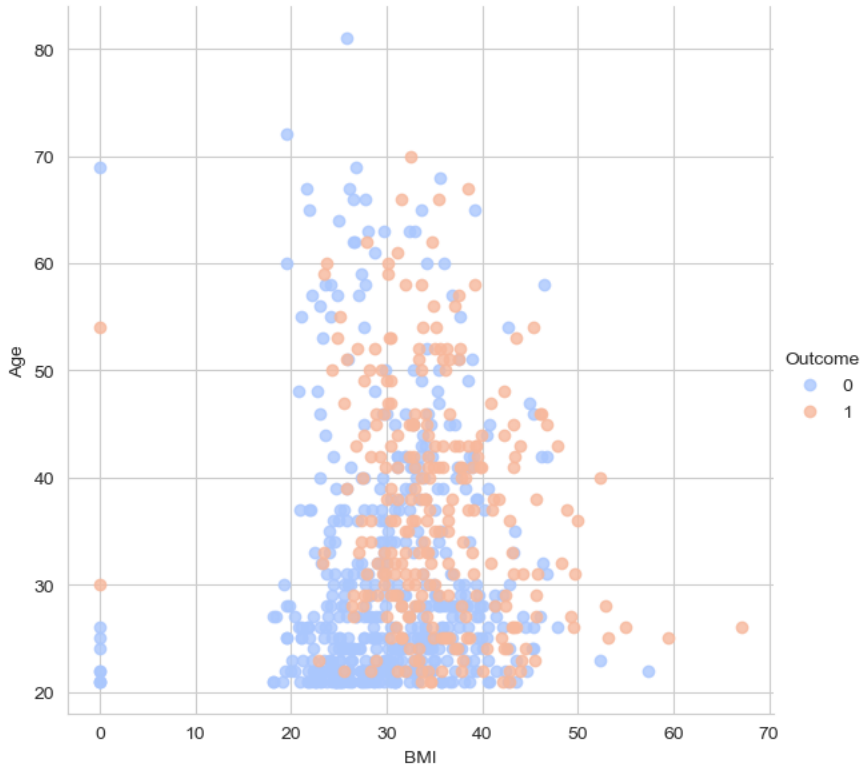
- **Embarazos**: Número de veces que la paciente ha estado embarazada.
- **Glucosa**: Concentración de glucosa plasmática a las 2 horas en una prueba de tolerancia a la glucosa oral.
- **Presión arterial**: Presión arterial diastólica (mm Hg).
- **Grosor de la piel:** Grosor del pliegue cutáneo del tríceps (mm).
- **Insulina**: Insulina sérica a las 2 horas (mu U/ml).
- **IMC**: Índice de masa corporal (peso en kg/(altura en m)^2).
- **DiabetesPedigreeFunction**: Función que puntúa la probabilidad de diabetes en función de los antecedentes familiares.
- **Edad**: Edad de la paciente (años).
- **Resultado**: Variable de clase (0 o 1), donde 1 representa la presencia de diabetes y 0 representa la ausencia de diabetes.

Desarrollo.

La base de datos cuenta con 768 filas y 9 columnas:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1





K-Means

K-means es un método de aprendizaje no supervisado para agrupar puntos de datos. El algoritmo divide iterativamente los puntos de datos en K grupos minimizando la varianza en cada grupo.

La agrupación en grupos de K-medias requiere que seleccionemos K, la cantidad de grupos en los que queremos agrupar los datos. El método del codo nos permite graficar la inercia (una métrica basada en la distancia) y visualizar el punto en el que comienza a disminuir linealmente. Este punto se conoce como "elbow" y es una buena estimación del mejor valor para K en función de nuestros datos.

Utilizando la base de datos de Diabetes.

En el caso del k-means para la base de datos de *Diabetes* analizada consideramos lo siguiente:

- Usaremos dos grupos, porque sabemos que solo hay dos tipos de diagnóstico: positivo o negativo.

Tenemos los valores de los centroides:

```
# Centroides
```

```
kmeans.cluster_centers_
```

```
array([[  3.7030303 , 141.46060606,  72.78787879,  31.2       ,
        253.70909091,  34.98545455,   0.59724848,  33.7030303 ],
       [  3.88391376, 115.26699834,  68.09784411,  17.6185738 ,
         32.21227197,  31.17363184,   0.43757048,  33.11442786]])
```

A continuación tenemos la matriz de clasificación:

```
[[421  79]
 [182  86]]
```

	precision	recall	f1-score	support
0	0.70	0.84	0.76	500
1	0.52	0.32	0.40	268
accuracy			0.66	768
macro avg	0.61	0.58	0.58	768
weighted avg	0.64	0.66	0.64	768

Tiene un accuracy del 66% considerando la clasificación previamente dada en el conjunto de datos.

Probando el método de k-means con otro conjunto de datos.

Datos:

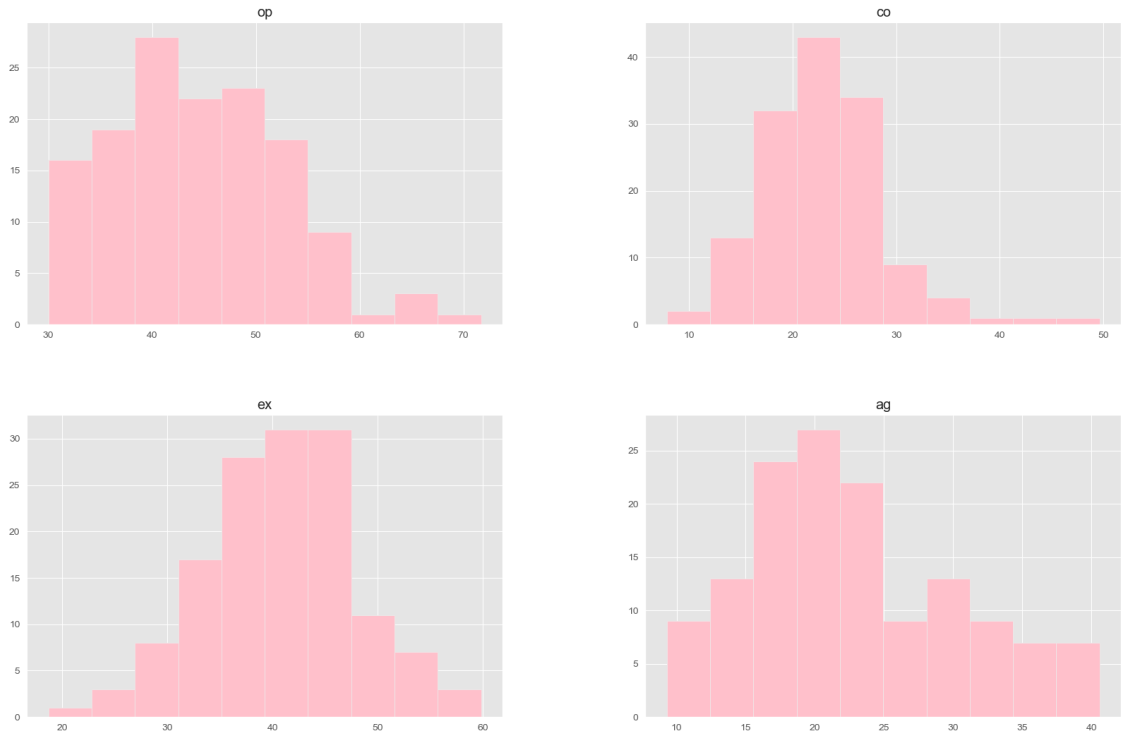
- **Usuario** - el nombre en Twitter.
- **op** = Openness to experience – grado de apertura mental a nuevas experiencias, curiosidad, arte.
- **co** = Conscientiousness – grado de orden, prolijidad, organización.
- **ex** = Extraversion – grado de timidez, solitario o participación ante el grupo social.
- **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:
    1. Actor / actriz
    2. Cantante

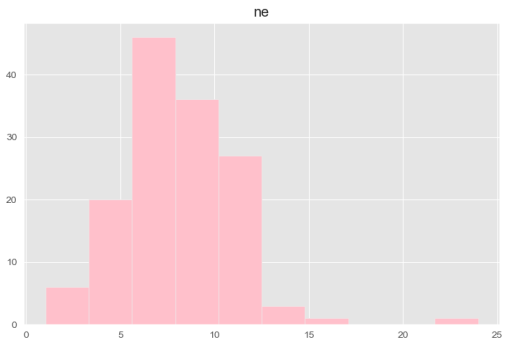
3. Modelo
4. TV, Seties
5. Radio
6. Tecnología
7. Deportes
8. Política
9. Escritor

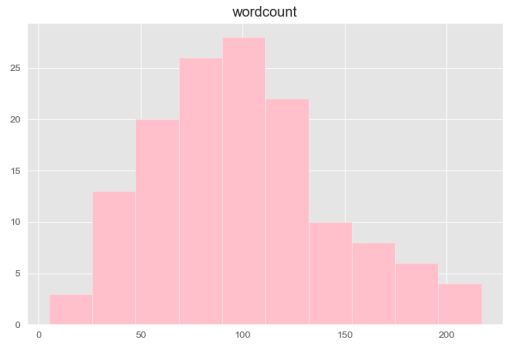
Primeras 5 filas de la base de datos:

	usuario	op	co	ex	ag	ne	wordcount	categoria
0	3gerardpique	34.297953	28.148819	41.948819	29.370315	9.841575	37.0945	7
1	aguerosergiokun	44.986842	20.525865	37.938947	24.279098	10.362406	78.7970	7
2	albertochicote	41.733854	13.745417	38.999896	34.645521	8.836979	49.2604	4
3	AlejandroSanz	40.377154	15.377462	52.337538	31.082154	5.032231	80.4538	2
4	alfredocasero1	36.664677	19.642258	48.530806	31.138871	7.305968	47.0645	4

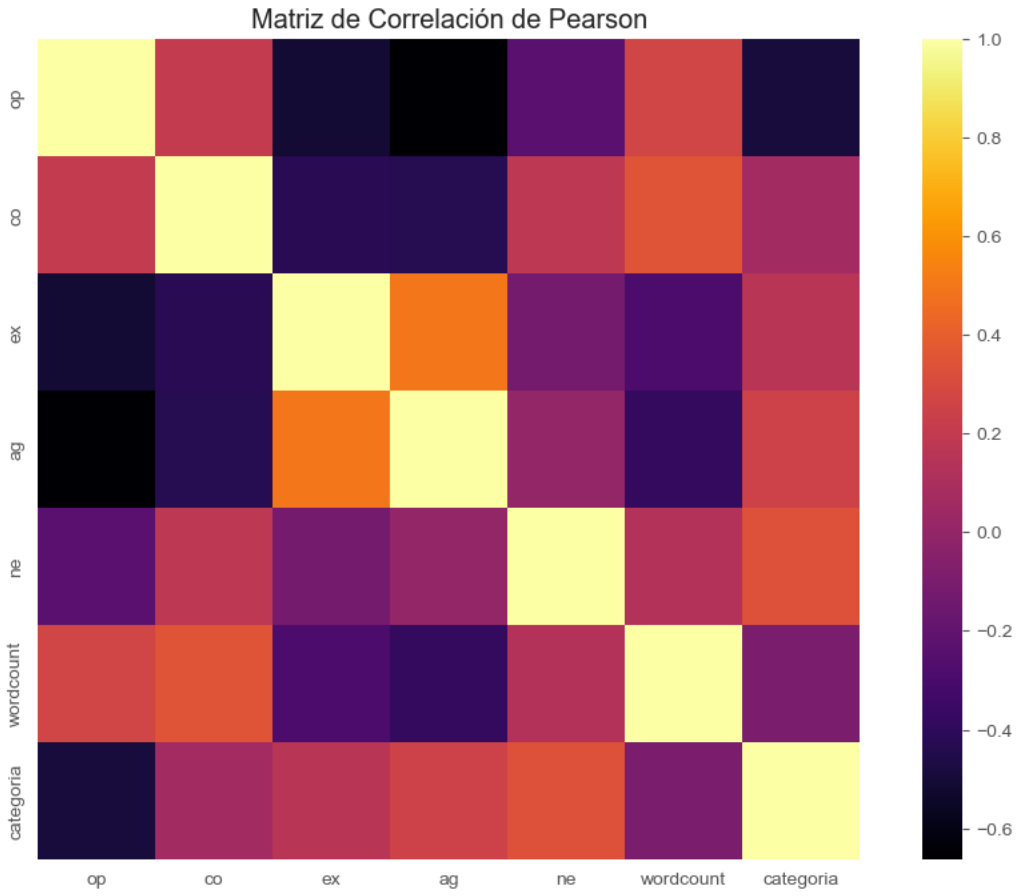
Histogramas para todas las variables:



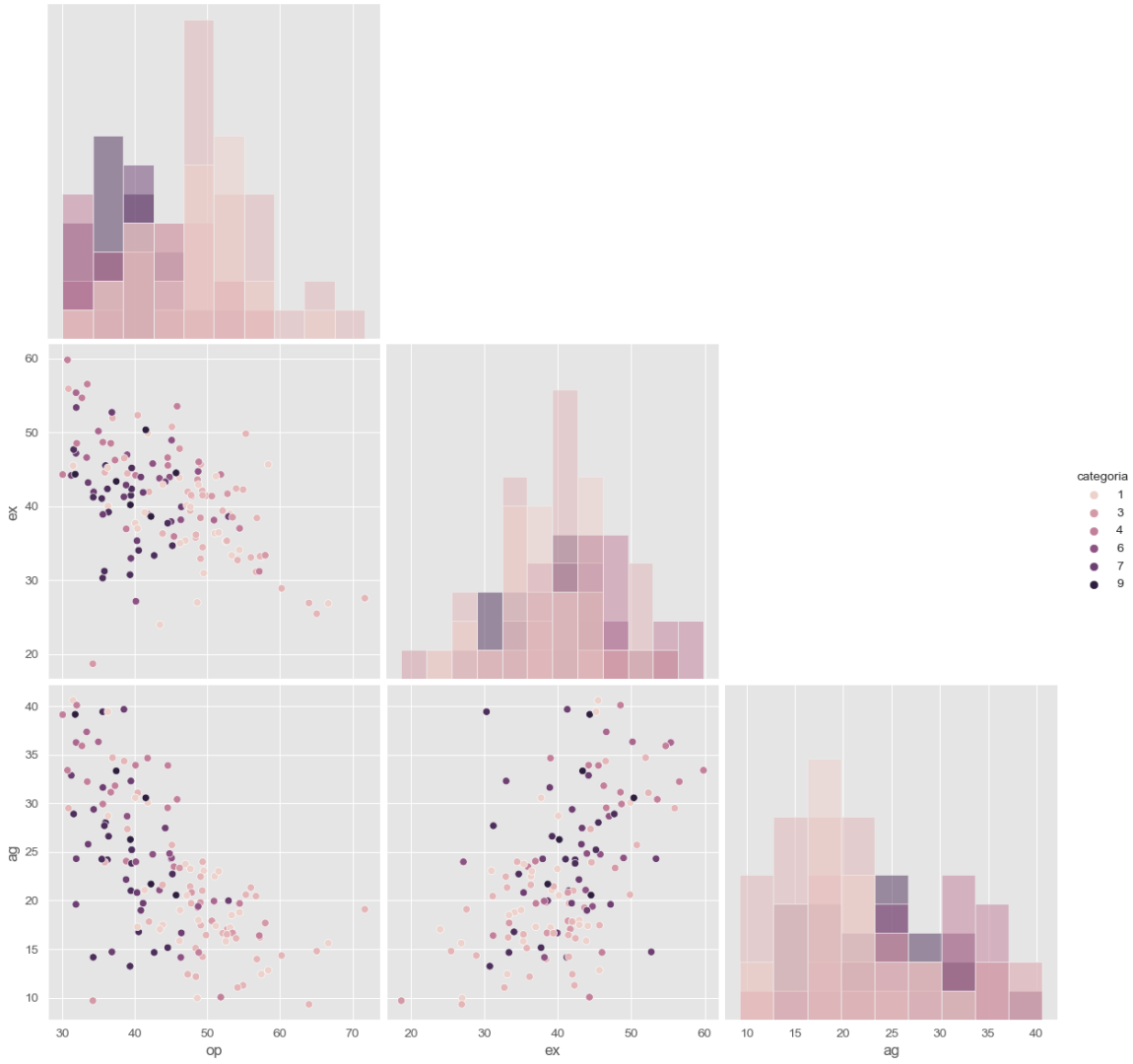




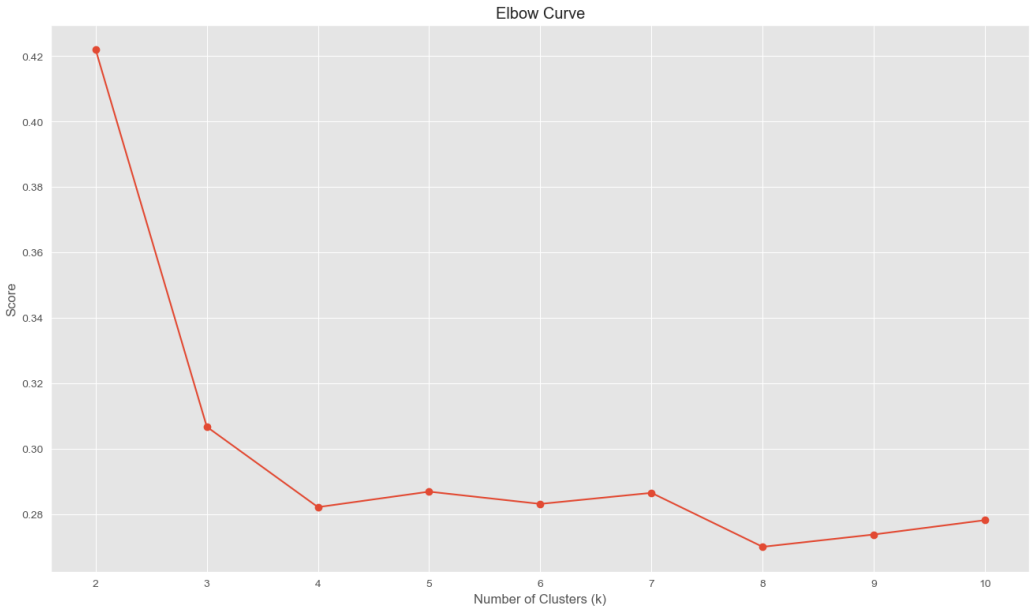
Se realiza una matriz de correlación de Pearson.



Se consideran las variables: **op**, **ex** y **ag**.



Identificar la cantidad de clústeres.



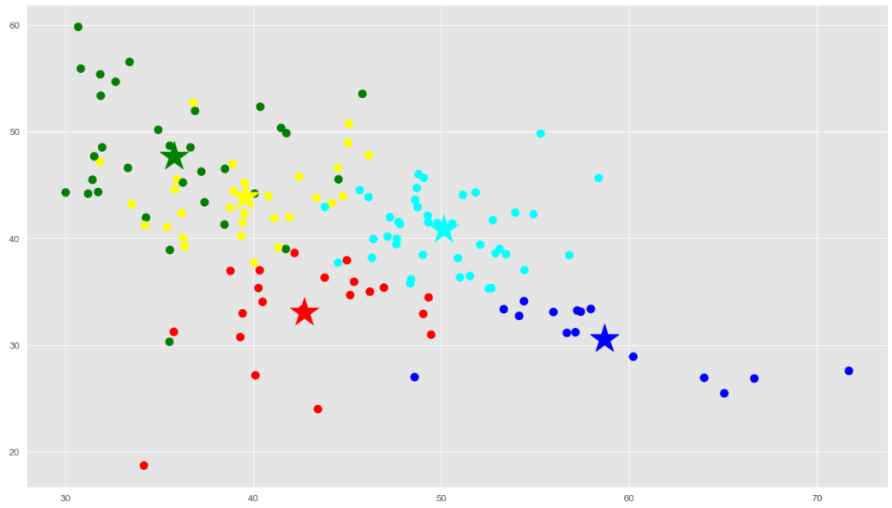
De acuerdo a las imagenes anteriores nos quedamos con 5 clusters

A continuación se muestran los centroides:

```
array([[42.73275924, 33.0308789 , 21.11743814],
       [35.80703706, 47.6507045 , 33.91511891],
       [58.70462307, 30.53566167, 15.72207033],
       [50.15530371, 40.81295548, 17.39048745],
       [39.59124977, 43.8789952 , 23.78707927]])
```

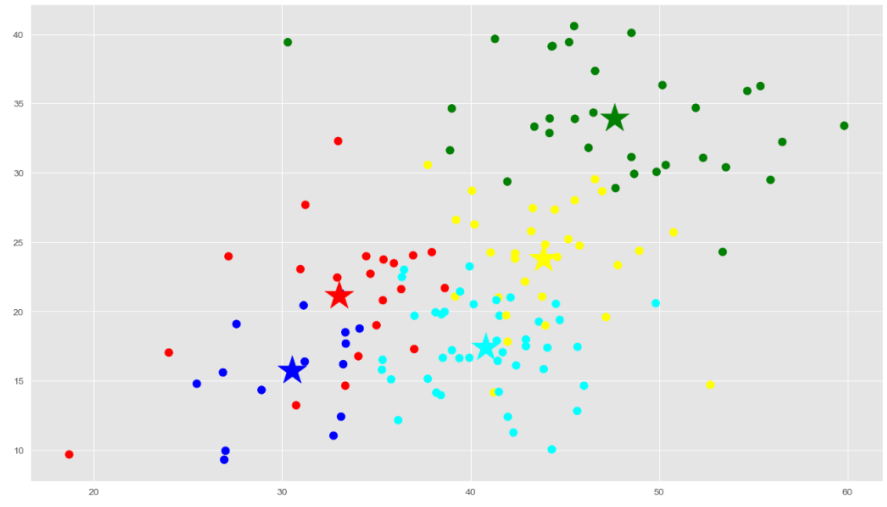
```
f1 = df_twitter['op'].values
f2 = df_twitter['ex'].values

plt.scatter(f1, f2, c=asignar, s=70)
plt.scatter(C[:, 0], C[:, 1], marker='*', c=colores, s=1000)
plt.show()
```




```
f1 = df_twitter['ex'].values
f2 = df_twitter['ag'].values

plt.scatter(f1, f2, c=asignar, s=70)
plt.scatter(C[:, 1], C[:, 2], marker='*', c=colores, s=1000)
plt.show()
```



Cantidad de elementos por cluster.

	color	cantidad
0	red	21
1	green	32
2	blue	15
3	cyan	42
4	yellow	30