Rafael Cueto Rodríguez – Machine Learning

Base de datos en Excel de nombre “prueba” con 403 instancias y 6 columnas

```
data=pd.read_excel("prueba.xlsx")
data
```

	STG	SCG	STR	LPR	PEG	UNS
0	0.00	0.10	0.50	0.26	0.05	Very Low
1	0.05	0.05	0.55	0.60	0.14	Low
2	0.08	0.18	0.63	0.60	0.85	High
3	0.20	0.20	0.68	0.67	0.85	High
4	0.22	0.22	0.90	0.30	0.90	High
...	...	...	...	...	...	...
398	0.61	0.78	0.69	0.92	0.58	High
399	0.78	0.61	0.71	0.19	0.60	Middle
400	0.54	0.82	0.71	0.29	0.77	High
401	0.50	0.75	0.81	0.61	0.26	Middle
402	0.66	0.90	0.76	0.87	0.74	High

```
data["UNS"]=data["UNS"].replace("Very Low",0)
data["UNS"]=data["UNS"].replace("very_low",0)
data["UNS"]=data["UNS"].replace("Low",1)
data["UNS"]=data["UNS"].replace("Middle",2)
data["UNS"]=data["UNS"].replace("High",3)
```

Se realiza categorización de la variable UNS:

- very low = 0
- very_low = 0
- Low = 1
- Middle = 2
- High = 3

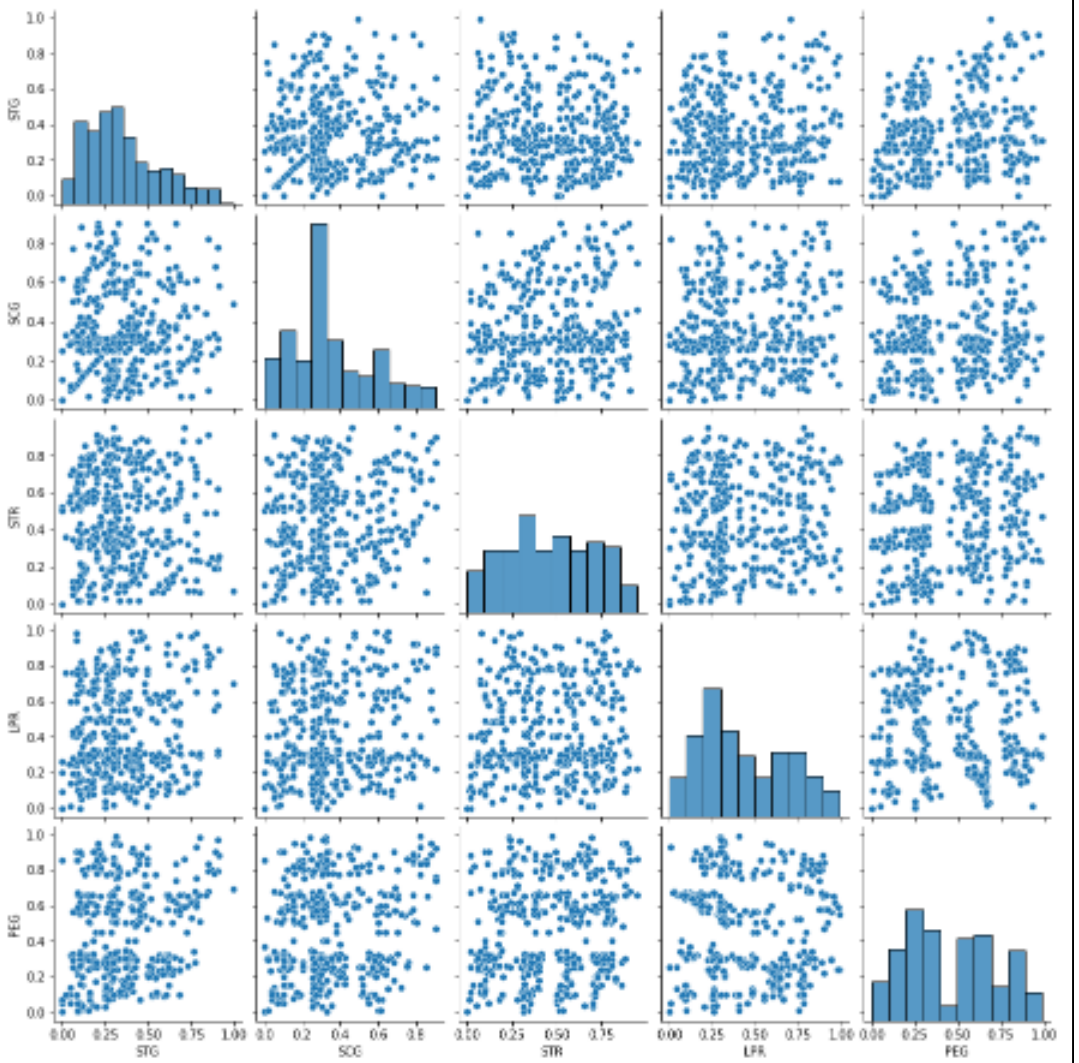
En esta tabla se puede visualizar los datos estadísticos descriptivos más comunes como son la media, desviación estándar, los valores máximos y mínimos.

Por ejemplo la variable STG (El grado de tiempo de estudio para las materias del objeto de la meta) presenta una media de 0.35 horas y una desviación estándar de 0.21 horas.

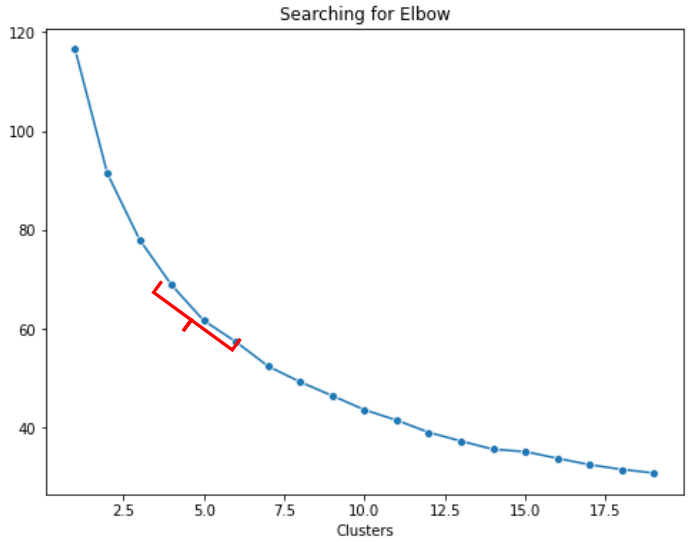
```
data.describe()
```

	STG	SCG	STR	LPR	PEG	UNS
count	403.000000	403.000000	403.000000	403.000000	403.000000	403.000000
mean	0.353141	0.355940	0.457655	0.431342	0.456360	1.684864
std	0.212018	0.215531	0.246684	0.257545	0.266775	0.986195
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	0.200000	0.200000	0.265000	0.250000	0.250000	1.000000
50%	0.300000	0.300000	0.440000	0.330000	0.400000	2.000000
75%	0.480000	0.510000	0.680000	0.650000	0.660000	3.000000
max	0.990000	0.900000	0.950000	0.990000	0.990000	3.000000

Con este grafico se visualiza antes de aplicar los métodos la forma como se comportan los datos con respecto a las variables analizadas y en cuantos posibles cluster puedo agrupar los datos.



Aplicando el método K-Mean

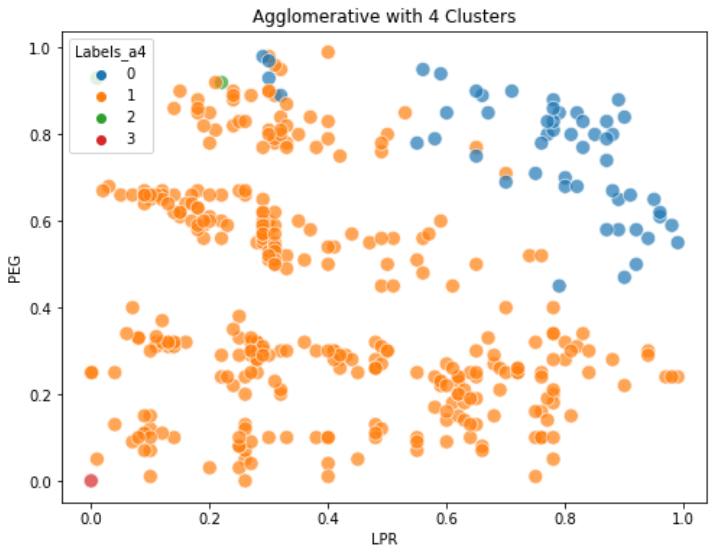


Con la grafica del método del codo se puede visualizar que entre 4 a 6 cluster se pueden utilizar, ya que no se logra ver una disminución considerable en la inercia a medida que se aumentan los cluster

Parece ser una buena opción tomar 4 cluster , debido que los puntos de los cluster se cruzan muy poco.

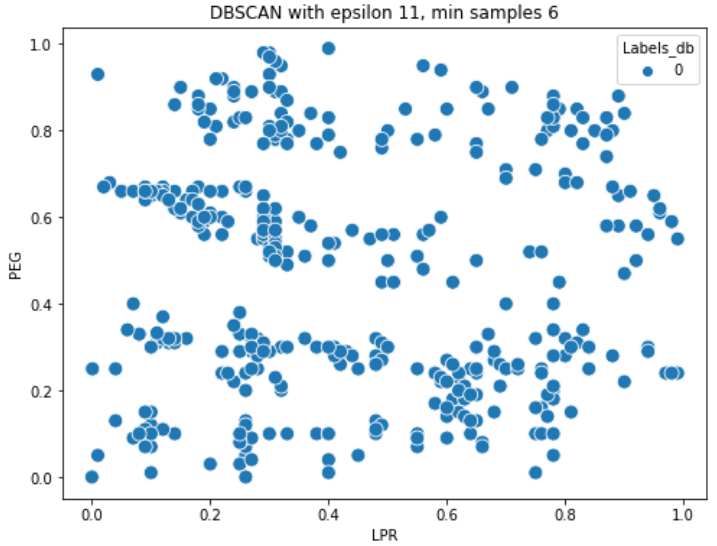
- Cluster 0: Bajo LPR y PEG Bajo
- Cluster 1: Alto LPR y PEG Alto
- Cluster 2: Alto LPR y PEG Bajo
- Cluster 3: Bajo LPR y PEG Alto

Aplicando Clusterin Jerárquico



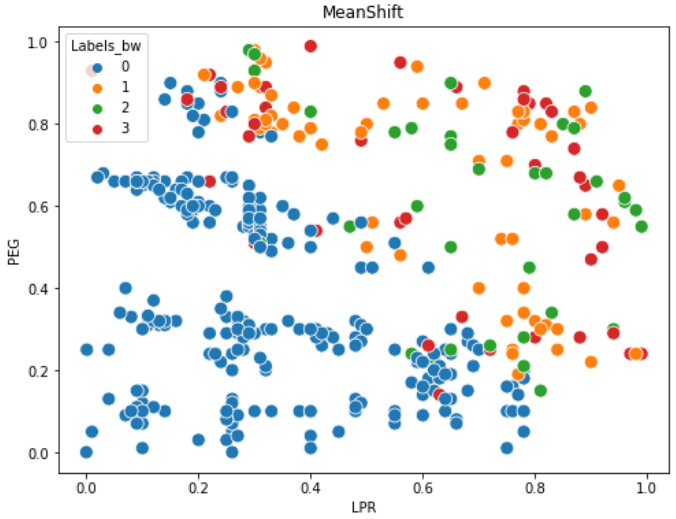
No parece ser una buena opción el método aglomerativo ó clusterin jerárquico en comparación con k-Mean para 4 cluster.

Aplicando el método DBSCAN



Con el método **DBSCAN** solo se observa que toma un solo grupo y no parece ser buena opción.

Aplicando Algoritmo Mean Shift



Con el método **algoritmo mean shift** no parece ser buena opción.

Error promedio de detección para meter los datos de clasificación

```
print("el error de detección promedio para un K-Mean de 4 cluster es=",(np.round((10+6.2015503875969+7.377049180327869+3.9215686274509802)/4,2)),"%")
```

```
el error de detección promedio para un K-Mean de 4 cluster es= 6.88 %
```

¿Qué tan bueno es el método de clasificación?

```
np.round((1 - accuracy_score(data1["UNS_Kmeans"],data1["UNS"]))*100,2)
```

```
97.02
```

CONCLUSIÓN

El método que mejor se ajusta con 4 cluster para la data analizada es el KMean. Tiene un 97.02% de precisión en el modelo de clasificación y además este método nos da un error promedio de detección de 6,88%, es decir, tiene un 93,12% de precisión en meter los datos de clasificación.