UNIVERSIDAD NACIONAL DE ROSARIO
FACULTAD DE CIENCIAS EXACTAS, INGENIERÍA Y AGRIMENSURA

Trabajo Práctico N° 1

MINERÍA DE DATOS

INFORME

Ejercicios de aplicación
Unidades 2 y 3

Carrera: Tecnicatura Universitaria en Inteligencia Artificial

Alumno: *Mussi, Miguel*

Ciclo: *2023*

PARTE I: INTRODUCCIÓN

Objetivo

El objetivo de este trabajo practico es integrar los conocimientos adquiridos en las unidades 2 y 3 en un problema real asociado a los cultivos.

Actividades

1. Descargar un conjunto de datos, Crop_recommendation.csv, para realizar el trabajo práctico.

2. Analizar los atributos del conjunto de datos (distribuciones, valores, outliers, tipos de datos, etc.) y elegir un método de estandarización.

3. Realizar PCA y determinar el número de componentes principales considerando alguno de los 3 criterios datos en la práctica. Graficar la varianza acumulada y las componentes de PCA en un gráfico 2 o 3D con sus respectivas clases.

4. Aplicar Isomap y analizar los resultados obtenidos variando el número de vecinos y componentes. Realizar un gráfico en 2D de utilizando dos componentes.

5. Aplicar t-SNE y analizar los resultados obtenidos variando el número de iteraciones, componentes y perplejidad. Realizar un gráfico en 2D de utilizando dos componentes.

6. Aplicar K-means y analizar los resultados obtenidos variando el número de clusters y obtener el número óptimo de clusters mediante GAP. Realizar un gráfico en 3D de utilizando tres atributos de los datos y donde los colores estén asociados a los clusters.

7. Aplicar clustering jerárquico y determinar cuál número sería el que mejor represente los datos. Utilizar el score de Silhouette y calcular el número óptimo de cluster por medio de GAP.

Presentación

La entrega es por grupos de dos estudiantes y se entregan dos archivos por grupo. Cualquier integrante del grupo puede hacer la entrega mediante el campus de la materia.

El formato del informe deberá ser pdf; mientras que el código deberá ser py.

El informe deberá tener una carátula en la que se indique: año, materia, integrantes. Además, deberá contar con una sección de conclusiones al final del mismo.

Las entregas fuera del plazo establecido no serán consideradas salvo excepciones previamente justificadas por el grupo.

PARTE II: DESARROLLO

El código

El código se adjunta en formato .py y se ofrece el enlace al cuaderno de Google Colab para un completo análisis de las ejecuciones y visualizaciones

https://colab.research.google.com/drive/1pxpMOUctDN0go9OrFGLCxU4NOK6RToSt

Dataset

El dataset proviene de un archivo denominado **Crop_recomendatios.csv** y contiene información sobre una serie de cultivos. Los atributos analizados son ***'N', 'P', 'K', 'temperature', 'humidity', 'ph', 'rainfall'*** y ***'label'***. Los tipos de cultivos analizados corresponden a las clases ***'rice', 'maize', 'chickpea', 'kidneybeans', 'pigeonpeas', 'mothbeans', 'mungbean', 'blackgram', 'lentil', 'pomegranate', 'banana', 'mango', 'grapes', 'watermelon', 'muskmelon', 'apple', 'orange', 'papaya', 'coconut', 'cotton', 'jute'*** y ***'coffee'.***

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

	N	P	K	temperature	humidity	ph	rainfall	label
0	90	42	43	20.879744	82.002744	6.502985	202.935536	rice
1	85	58	41	21.770462	80.319644	7.038096	226.655537	rice
2	60	55	44	23.004459	82.320763	7.840207	263.964248	rice
3	74	35	40	26.491096	80.158363	6.980401	242.864034	rice
4	78	42	42	20.130175	81.604873	7.628473	262.717340	rice

Análisis exploratorio

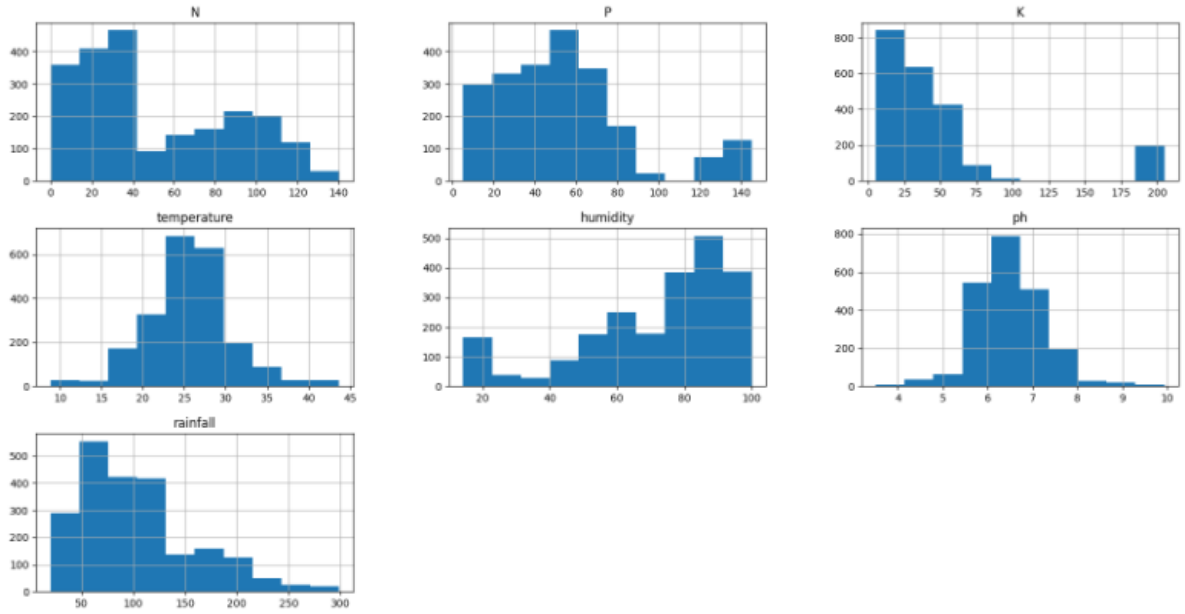
Como primera medida se ejecutan los comandos básicos para comprender el contenido, tipos de atributos, cantidad de datos faltantes/nulos y la distribución de los datos del dataset. Luego se procede a la recategorización de los datos que sean necesarios (Ejemplo: “label”)

- Descripción

```
[ ]  1  df.describe()
```

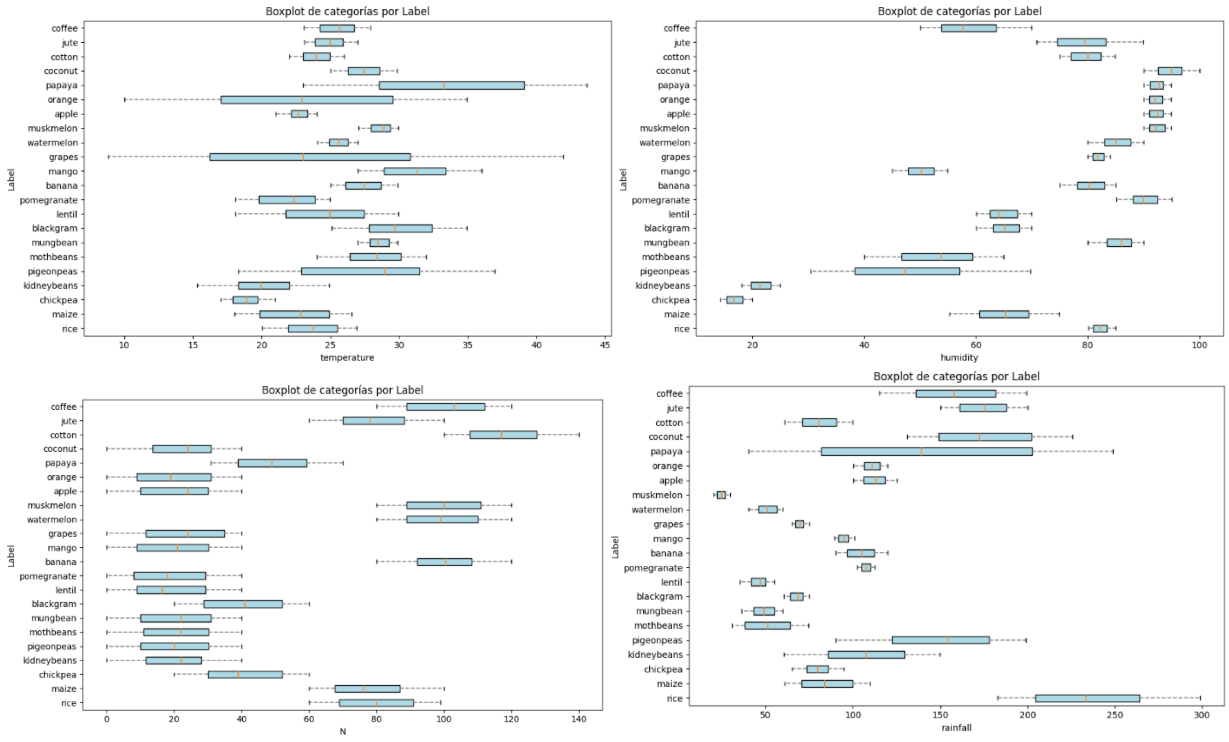
	N	P	K	temperature	humidity	ph	rainfall
count	2200.000000	2200.000000	2200.000000	2200.000000	2200.000000	2200.000000	2200.000000
mean	50.551818	53.362727	48.149091	25.616244	71.481779	6.469480	103.463655
std	36.917334	32.985883	50.647931	5.063749	22.263812	0.773938	54.958389
min	0.000000	5.000000	5.000000	8.825675	14.258040	3.504752	20.211267
25%	21.000000	28.000000	20.000000	22.769375	60.261953	5.971693	64.551686
50%	37.000000	51.000000	32.000000	25.598693	80.473146	6.425045	94.867624
75%	84.250000	68.000000	49.000000	28.561654	89.948771	6.923643	124.267508
max	140.000000	145.000000	205.000000	43.675493	99.981876	9.935091	298.560117

- Histogramas

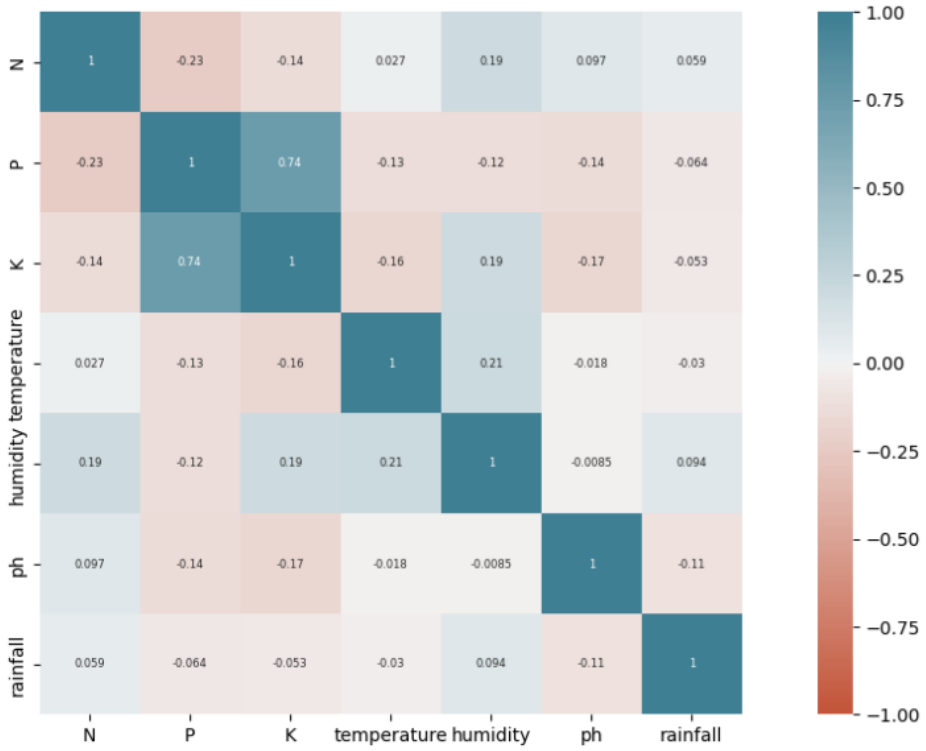


- Boxplots variables

Con el fin de analizar las distribuciones de las variables, valores outliers en cada una de ellas y demás estadísticos, se implementa una solución que muestra un gráfico de tipo boxplot para cada una de las clases de cultivos que tiene a la variable de referencia configurable por el usuario. Se adjuntan ejemplos de salidas.



- Matriz de correlación



- Estandarización

Se optó por la estandarización Z-score para los valores del dataset. A modo experimental se usaron dos métodos, uno por cálculo directo y el otro con la asistencia de la librería ***StandardScaler***. Los resultados son equivalentes. Se adjunta captura de la salida.

	N	P	K	temperature	humidity	ph	rainfall
0	1.068797	-0.344551	-0.101688	-0.935587	0.472666	0.043302	1.810361
1	0.933329	0.140616	-0.141185	-0.759646	0.397051	0.734873	2.242058
2	0.255986	0.049647	-0.081939	-0.515898	0.486954	1.771510	2.921066
3	0.635298	-0.556811	-0.160933	0.172807	0.389805	0.660308	2.537048
4	0.743673	-0.344551	-0.121436	-1.083647	0.454792	1.497868	2.898373
...	...	...	...	...	...	...	...
2195	1.529390	-0.587134	-0.318922	0.228814	-0.227709	0.401395	1.352437
2196	1.312641	-1.163269	-0.417666	0.355720	-0.666947	-0.494413	0.445183
2197	1.827421	-0.617457	-0.358420	-0.293218	-0.191235	-0.138120	1.271418
2198	1.800327	-0.647780	-0.279425	0.129612	-0.869518	0.373904	0.431545
2199	1.448109	-1.072300	-0.358420	-0.397667	-0.498020	0.401096	0.682005

2200 rows × 7 columns

PARTE III: PCA

Componentes principales

En primera instancia se obtienen las componentes principales y luego se determina el criterio adecuado para la selección de la cantidad de componentes a analizar.

- Criterios de selección

- Proporción de variancia acumulada (~75% -80%)
- Criterio de Kaiser (eigenvalues > 1)
- Gráfico del codo (Scree)

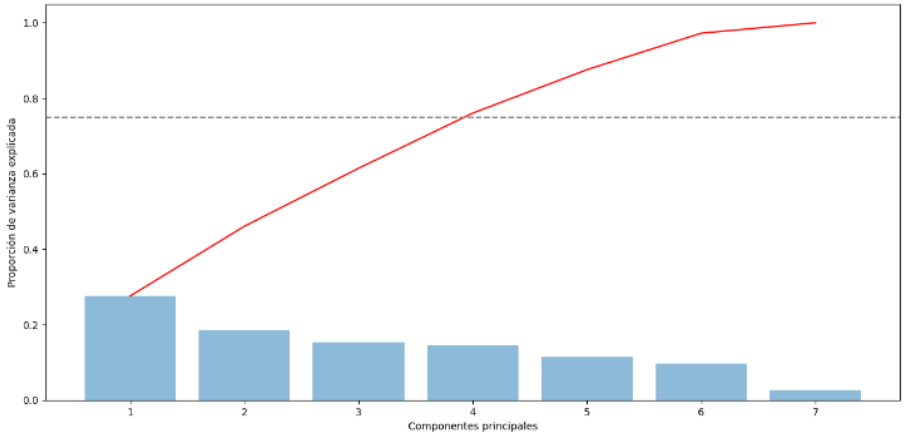
- Autovectores

	PC1	PC2	PC3	PC4	PC5	PC6	PC7
X1	-0.302191	0.643787	0.622607	-0.212428	-0.068483	-0.226943	-0.072532
X2	-0.334107	-0.034358	-0.283829	-0.359487	-0.737917	0.220657	-0.290158
X3	-0.112045	-0.109939	-0.163173	-0.248228	-0.213599	-0.548520	0.735267
X4	-0.541651	-0.046293	-0.154867	0.690826	-0.067171	-0.395700	-0.205318
X5	-0.507785	0.082331	0.033425	0.154865	0.128871	0.651881	0.518382
X6	-0.482904	-0.376847	-0.028967	-0.500418	0.547871	-0.125712	-0.239930
X7	-0.008473	-0.649104	0.692268	0.111282	-0.289624	0.040028	0.038577

- Autovalores y varianza acumulada

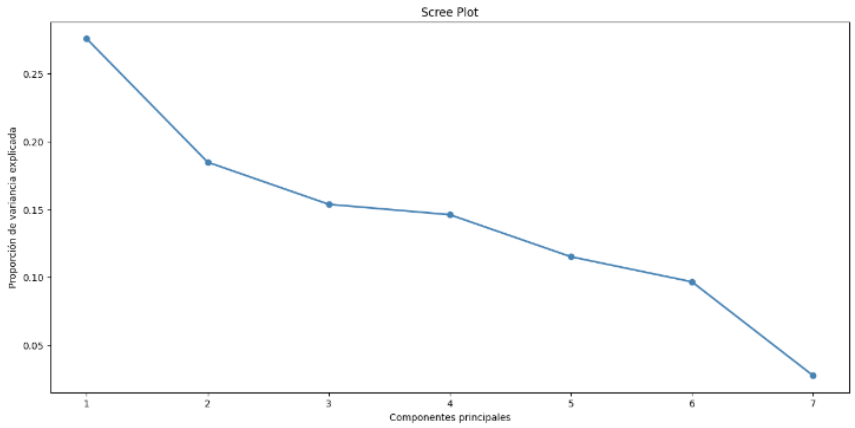
	Eigenvalues	Proporción de variancia explicada	Proporción acumulada de variancia explicada
0	1.932096	0.275888	0.275888
1	1.294499	0.184844	0.460733
2	1.076999	0.153787	0.614520
3	1.023356	0.146127	0.760647
4	0.806295	0.115133	0.875780
5	0.676869	0.096652	0.972431
6	0.193069	0.027569	1.000000

- Gráfico de varianza acumulada



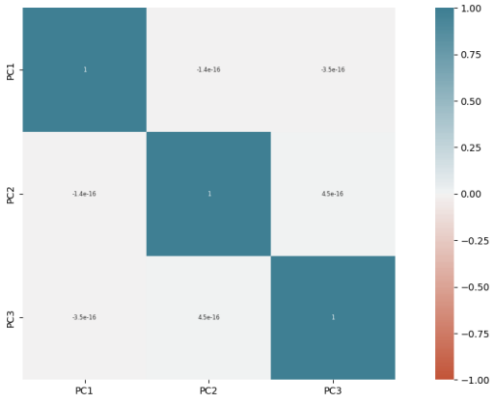
Las tres primeras componentes acumulan el 65% ~ 70% de la variabilidad total, es decir, están cercanas a cumplir con el primer criterio (>~75%). Si se consideraran las componentes cuyos eigenvalues son superiores a 1 (Criterio de Kaiser) se debería optar por extraer cuatro. Por conveniencia y practicidad para graficar las distribuciones de las componentes se decide seleccionar tres de ellas (primer citerio)

- Gráfico de codo



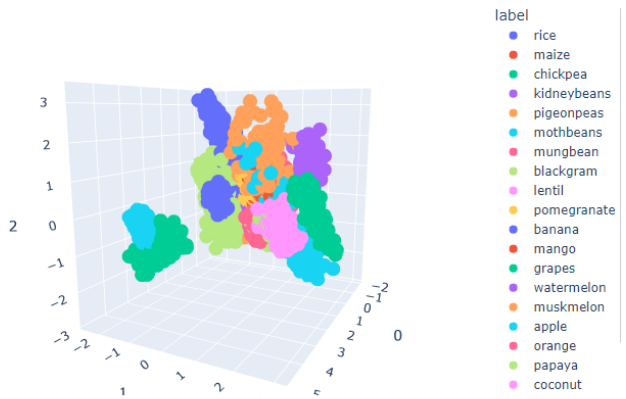
Al observar el gráfico del codo, vemos que el quiebre parece producirse entre la segunda y tercera componente. Considerando la primera y la segunda componente llegaríamos a un ~60% de la variabilidad total, por lo que consideramos óptimo tomar hasta la tercera componente.

- Matriz de correlación de componentes



Al observar la matriz de correlación se observa que las componentes seleccionadas son, efectivamente, linealmente independientes.

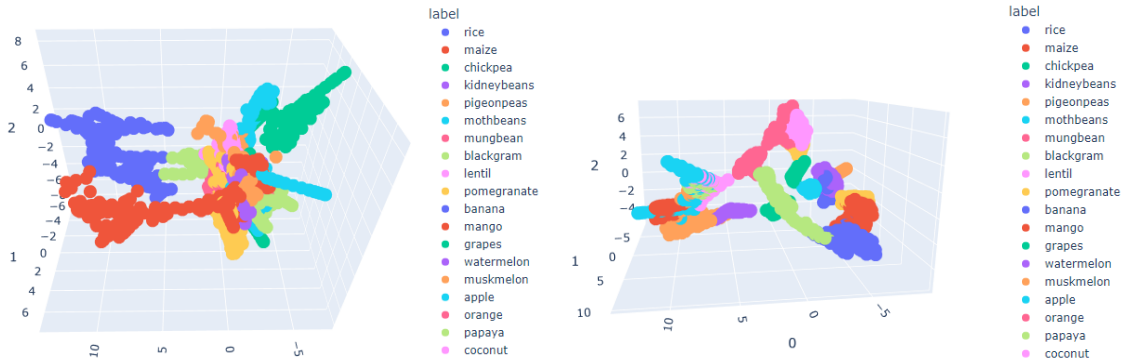
- Distribución de datos



PARTE IV: ISOMAP

Aplicación de la técnica

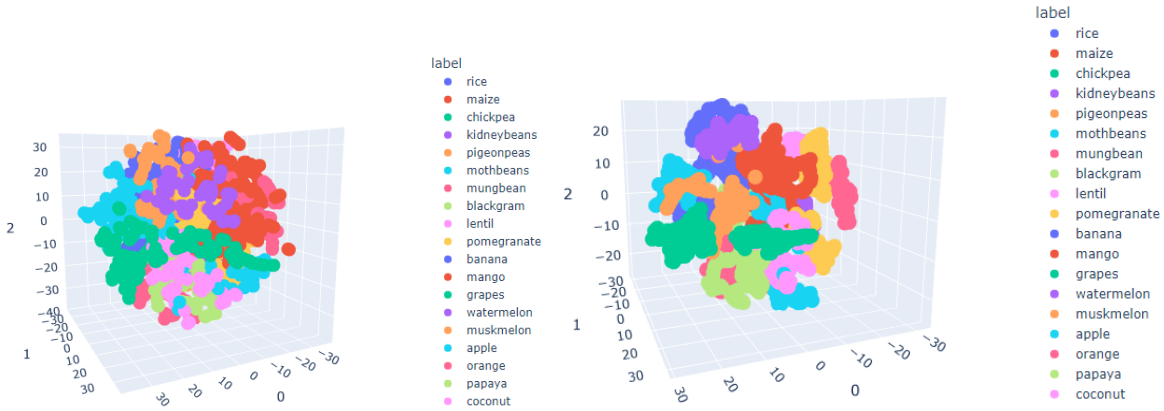
Se adjuntan dos gráficos en los que se experimentó con algunos de los hiperparámetros posibles (número de vecinos). La imagen 1 con **n_neighbors = 2** y la imagen 2 con **n_neighbors = 4**. En ambos, el número de componentes es 3.



PARTE V: T-SNE

Aplicación de la técnica

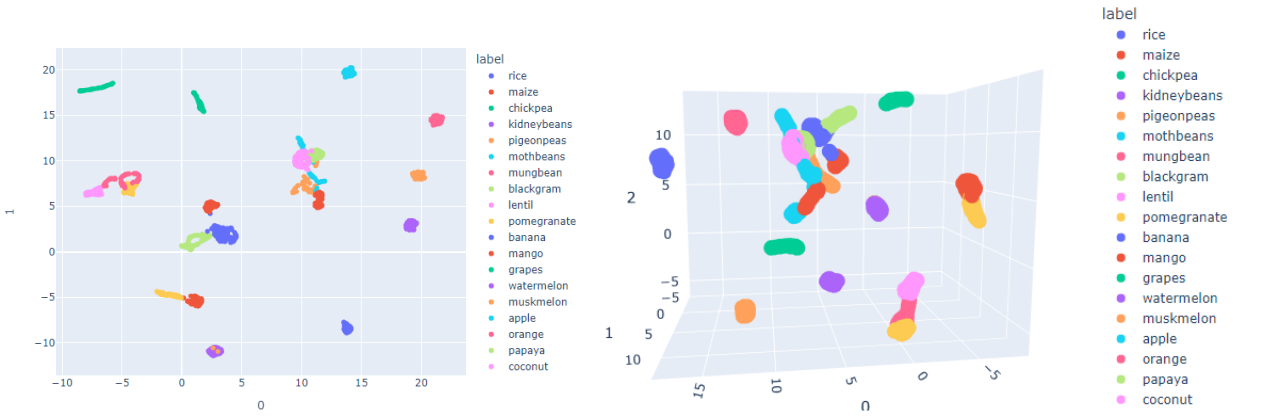
Se adjuntan dos gráficos en los que se experimentó con algunos de los hiperparámetros posibles (perplejidad). La imagen 1 con **perplexity = 2** y la imagen 2 con **perplexity = 5**. En ambos, el número de componentes es 3.



PARTE VI: UMAP

Aplicación de la técnica

Se adjuntan dos gráficos en los que se experimentó con UMAP en dos y tres componentes.



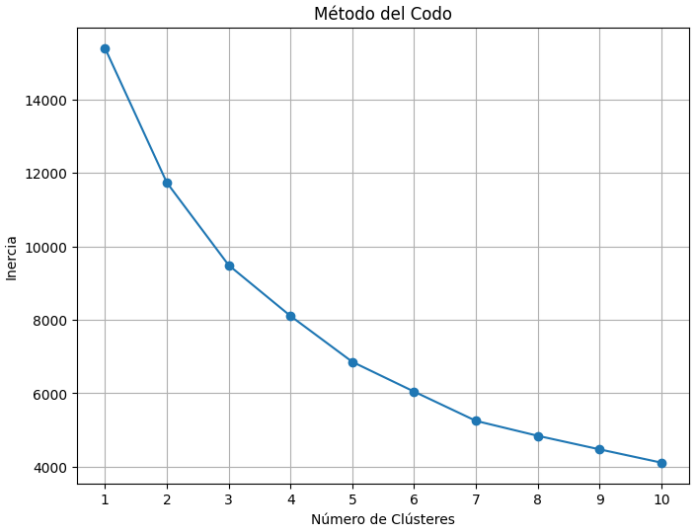
PARTE VII: K-MEANS

Método del codo

El objetivo es identificar un punto en el gráfico donde la disminución en la suma de los cuadrados de las distancias intraclúster (también conocida como inercia) comienza a disminuir de manera significativamente más lenta. Este punto se denomina "codo" y sugiere el número óptimo de clústeres para el conjunto de datos.

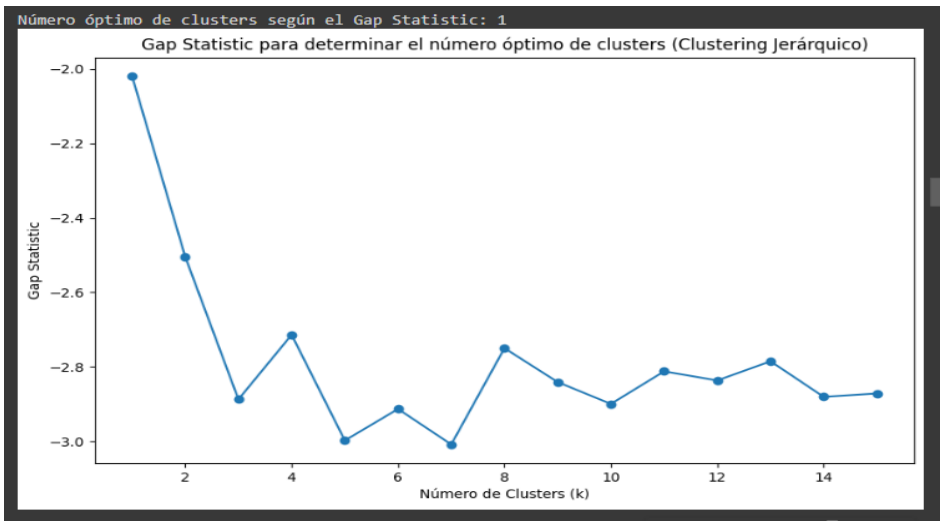
La inercia en K-means se calcula sumando los cuadrados de las distancias entre cada punto de datos y el centroide de su grupo asignado, y luego sumando estas distancias para todos los puntos en el conjunto de datos.

- Cálculo de la inercia



- Cálculo del GAP

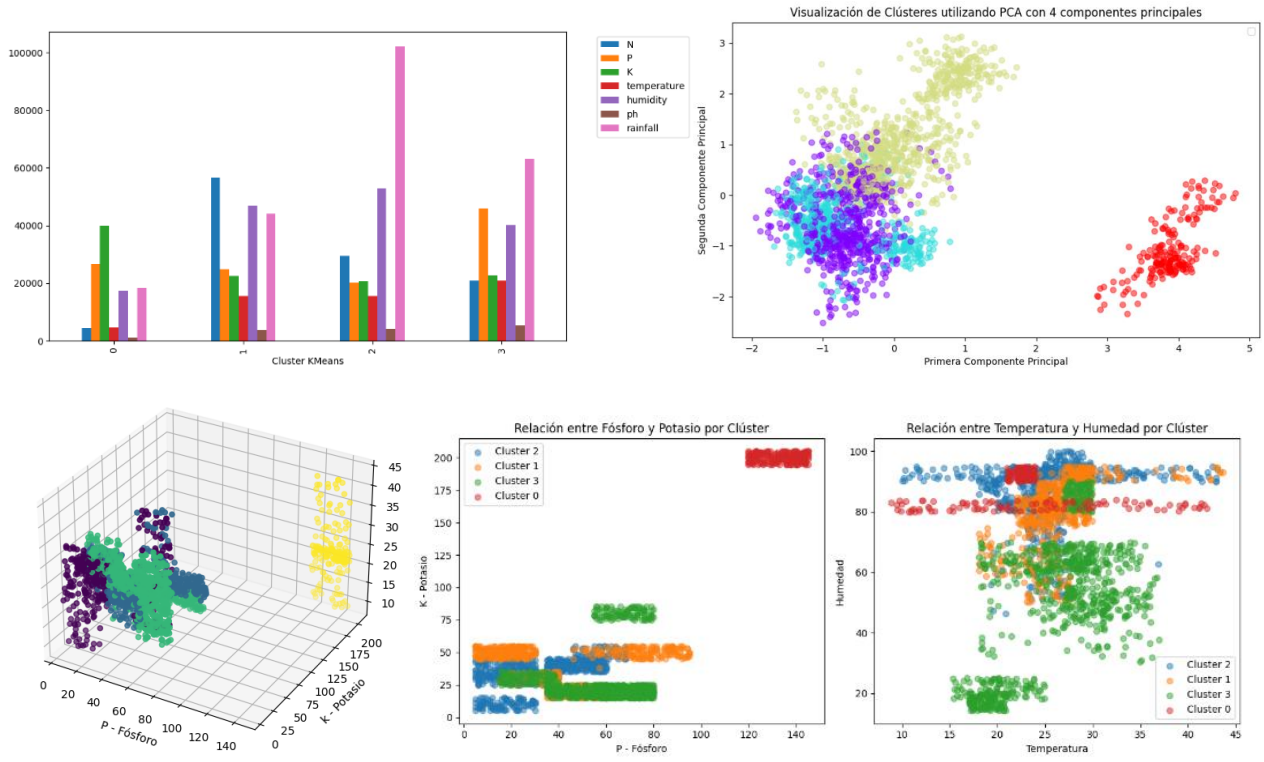
Compara la log-verosimilitud (medida que proporciona información sobre cuán bien un modelo probabilístico se ajusta a los datos observados,) de los datos con clusters y la log-verosimilitud de los datos completamente aleatorios. Cuanto mayor sea la diferencia entre estas log-verosimilitudes, mayor será la probabilidad de que los datos se agrupen en clusters reales en lugar de ser aleatorios. Un Gap Statistic más grande indica que el número correspondiente de clusters es más apropiado.



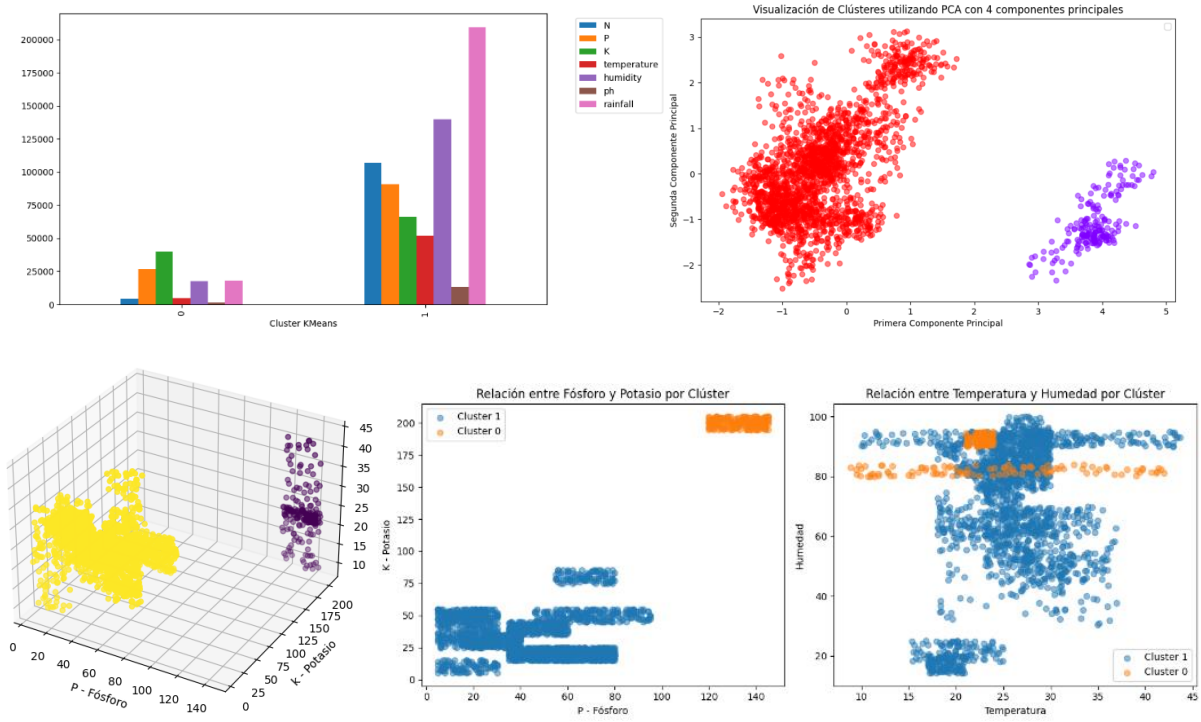
- Clusters

Se adjuntan grupos de cinco imágenes comparativas obtenidas con 2 y 4 clusters respectivamente. La decisión se basa en los resultados obtenidos en el gráfico de la inercia y el GAP.

4 clusters



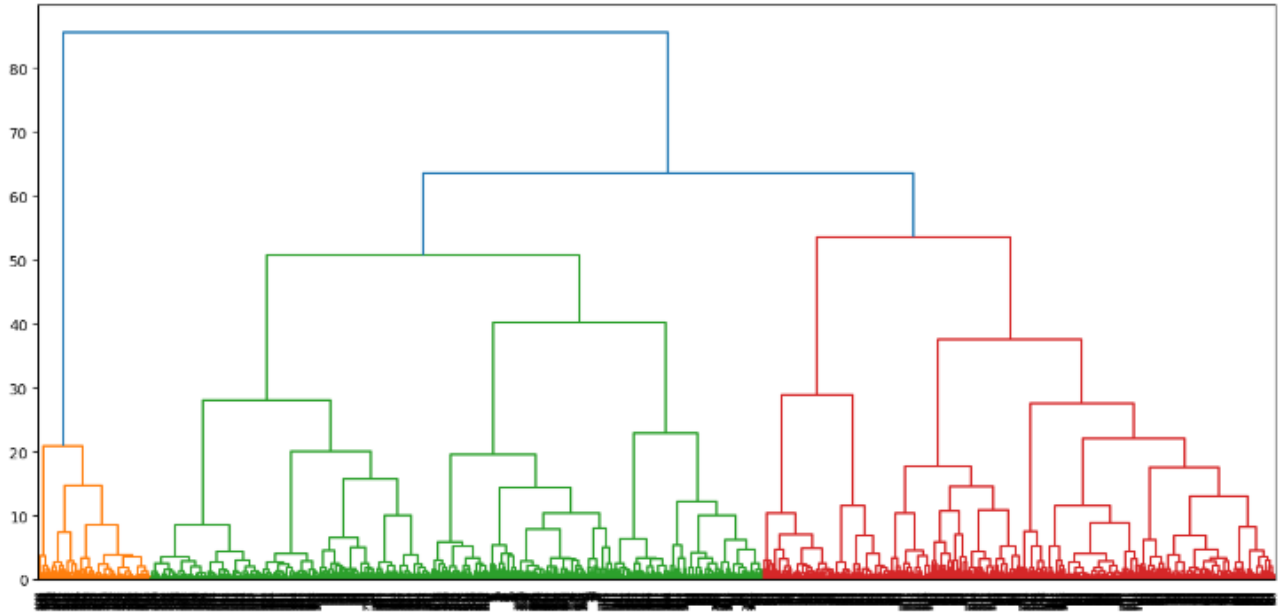
 # 2 clusters

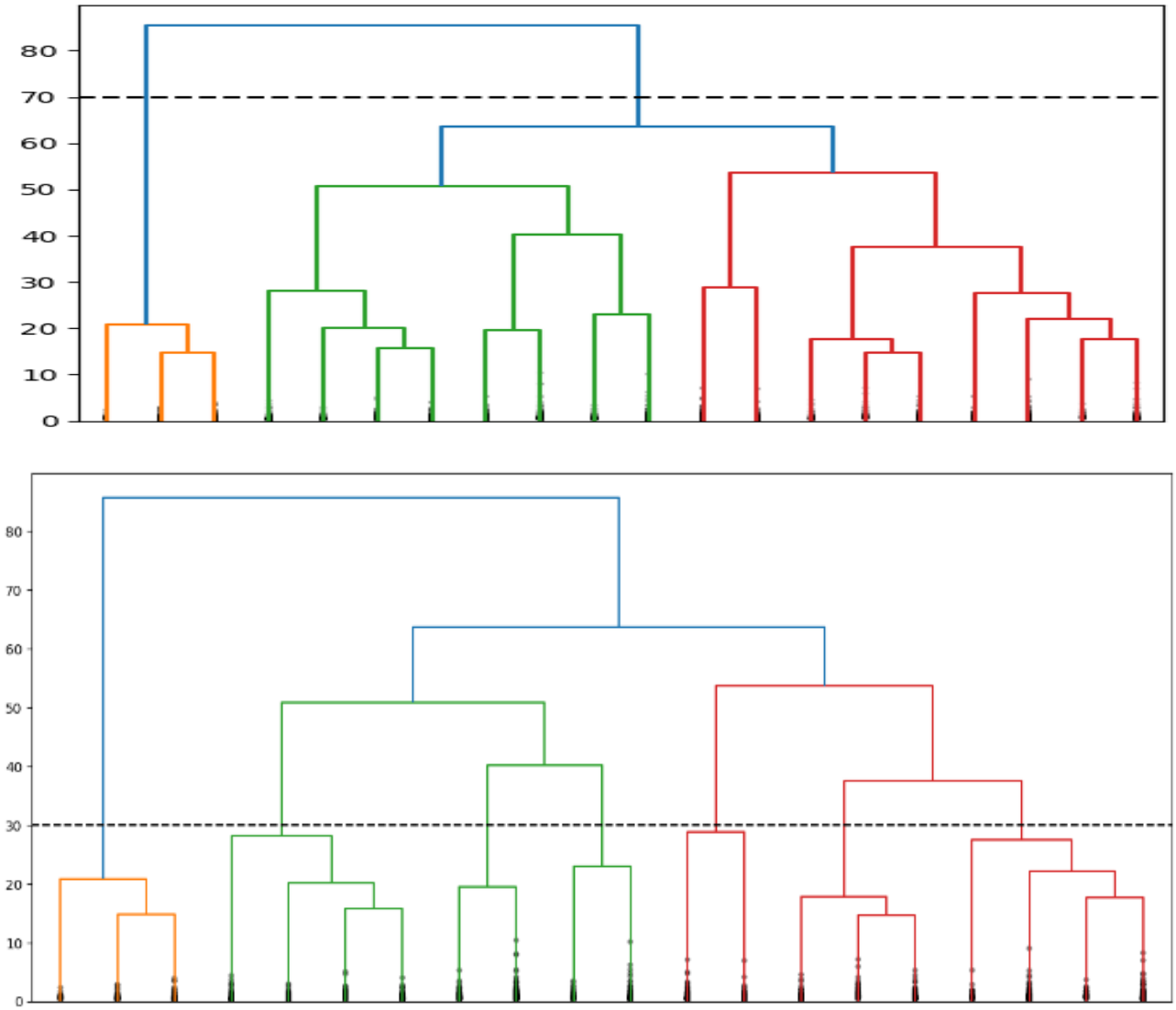


PARTE VIII: CLUSTERING JERÁRQUICO

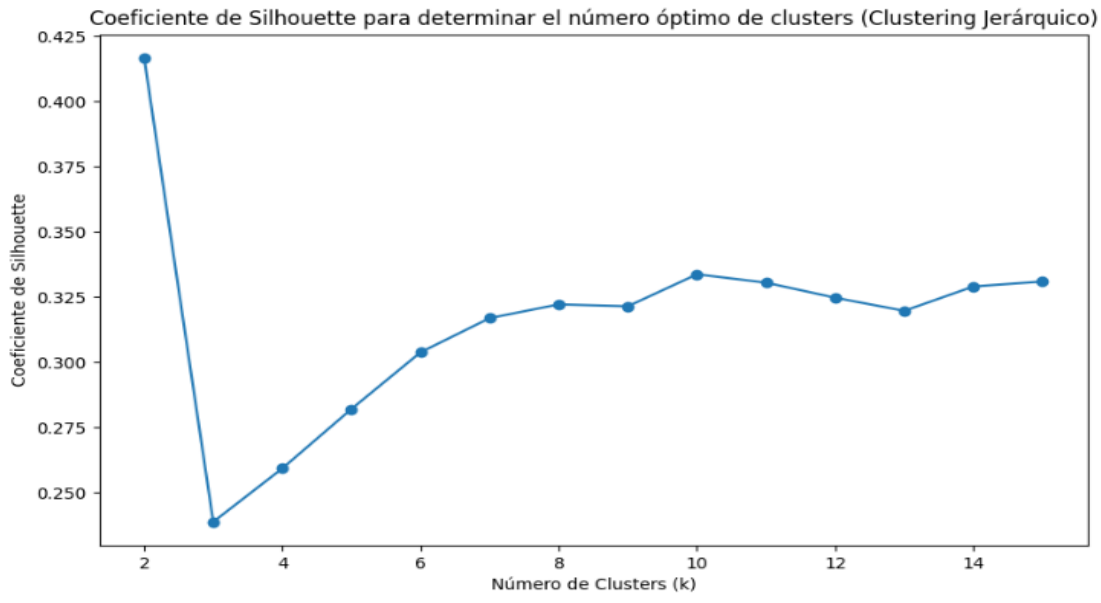
Aplicación de la técnica

Se obtiene el dendograma que muestra las agrupaciones que genera el algoritmo. Se adjunta, además, el dendograma truncado con un valor de altura p=20.




80
70
60
50
40
30
20
10
0

- Silhouette


Coeficiente de Silhouette para determinar el número óptimo de clusters (Clustering Jerárquico)
Coeficiente de Silhouette
0.425
0.400
0.375
0.350
0.325
0.300
0.275
0.250
2
4
6
8
10
12
14
Número de Clusters (k)

```
[ ]  1  n_clusters = 2
     2  clustering = AgglomerativeClustering(n_clusters=n_clusters)
     3
     4  cluster_assignments = clustering.fit_predict(X_scaled)
     5
     6  df['Cluster'] = cluster_assignments
```

```
[ ]  1  df['Cluster'].value_counts()
```

```
0    2000
1     200
Name: Cluster, dtype: int64
```

```
[ ]  1  from sklearn.metrics import silhouette_score,silhouette_samples
     2  silhouette_avg = silhouette_score(X_scaled, cluster_assignments)
     3  silhouette_avg
```

```
0.4168458131187345
```

- DBSCAN

Parametros DBSCAN (Density-Based Spatial Clustering of Applications with Noise):

eps (Epsilon): Este parámetro controla la distancia máxima entre dos muestras para que una sea considerada parte del mismo clúster. En otras palabras, establece la distancia máxima entre puntos vecinos que se agruparán juntos. Un valor pequeño de eps puede hacer que el algoritmo encuentre más clústeres, mientras que un valor grande puede fusionar más puntos en un solo clúster.

min_samples: Este parámetro define el número mínimo de muestras (puntos de datos) en un vecindario para que un punto sea considerado núcleo (core point). Los puntos núcleo son aquellos que tienen al menos min_samples puntos dentro de una distancia de eps. Los puntos que no son núcleo, pero están dentro del vecindario de un punto núcleo, se consideran puntos límite o de borde.

```
Número de clústeres identificados por DBSCAN: 4
```

- HDBSCAN

Parámetros HDBSCAN (Hierarchical Density-Based Spatial Clustering of Applications with Noise):

min_cluster_size: Este parámetro establece el tamaño mínimo del clúster. Los clústeres con menos puntos que min_cluster_size no se considerarán como clústeres válidos y se etiquetarán como ruido (-1 en las etiquetas). Aumentar este valor tiende a producir menos clústeres.

```
Número de clústeres identificados por HDBSCAN: 2
```

REPOSITORIOS

GitHub

Todos los archivos correspondientes a la resolución de este Trabajo Práctico se encuentran alojados en un repositorio público en GitHub.

https://github.com/MiguelMussi/Mineria_TP1

Google Colab

Se ofrece el enlace al cuaderno de Google Colab para un completo análisis de las ejecuciones y visualizaciones

https://colab.research.google.com/drive/1pxpMOUctDN0go9OrFGLCxU4NOK6RToSt