

# CLUSTERING JERÁRQUICO

*RECOMENDACIONES DE  
COMPRAS AMAZON*



# CARGAR BASE Y PREPROCESAMIENTO

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
data = pd.read_excel('Amazon.xlsx')
data.head()
```

	Unnamed: 0	Velocidad Entrega	Precio	Durabilidad	Imagen Producto	Valor Educativo	Servicio Retorno	Tamano Paquete	Calidad Producto	Numero Estrellas
0	Adam	205	3	345	235	24	23	26	21	17
1	Anna	9	15	315	33	25	4	42	215	28
2	Bernard	17	26	285	3	43	27	41	26	33
3	Edward	135	5	355	295	18	23	39	195	17
4	Emilia	3	45	48	39	34	46	225	34	43

```
import pandas as pd
from sklearn.preprocessing import normalize

# Separar columnas categoricas y numéricas
numerical_cols = data.select_dtypes(include=['float64', 'int64']).columns
categorical_cols = data.select_dtypes(include=['object', 'category']).columns

# Normalizar solo columnas numéricas
data_numerical = data[numerical_cols]
data_numerical_scaled = normalize(data_numerical)

# Convertir de nuevo a DataFrame conservando los nombres de las columnas
data_numerical_scaled_df = pd.DataFrame(data_numerical_scaled,
                                         columns=numerical_cols,
                                         index=data.index)

data_numerical_scaled_df
```

	Velocidad Entrega	Precio	Durabilidad	Imagen Producto	Valor Educativo	Servicio Retorno	Tamano Paquete	Calidad Producto	Numero Estrellas
0	0.438263	0.006414	0.737565	0.502399	0.051309	0.049171	0.055585	0.044895	0.036344
1	0.023235	0.038725	0.813234	0.085196	0.064542	0.010327	0.108431	0.555065	0.072287

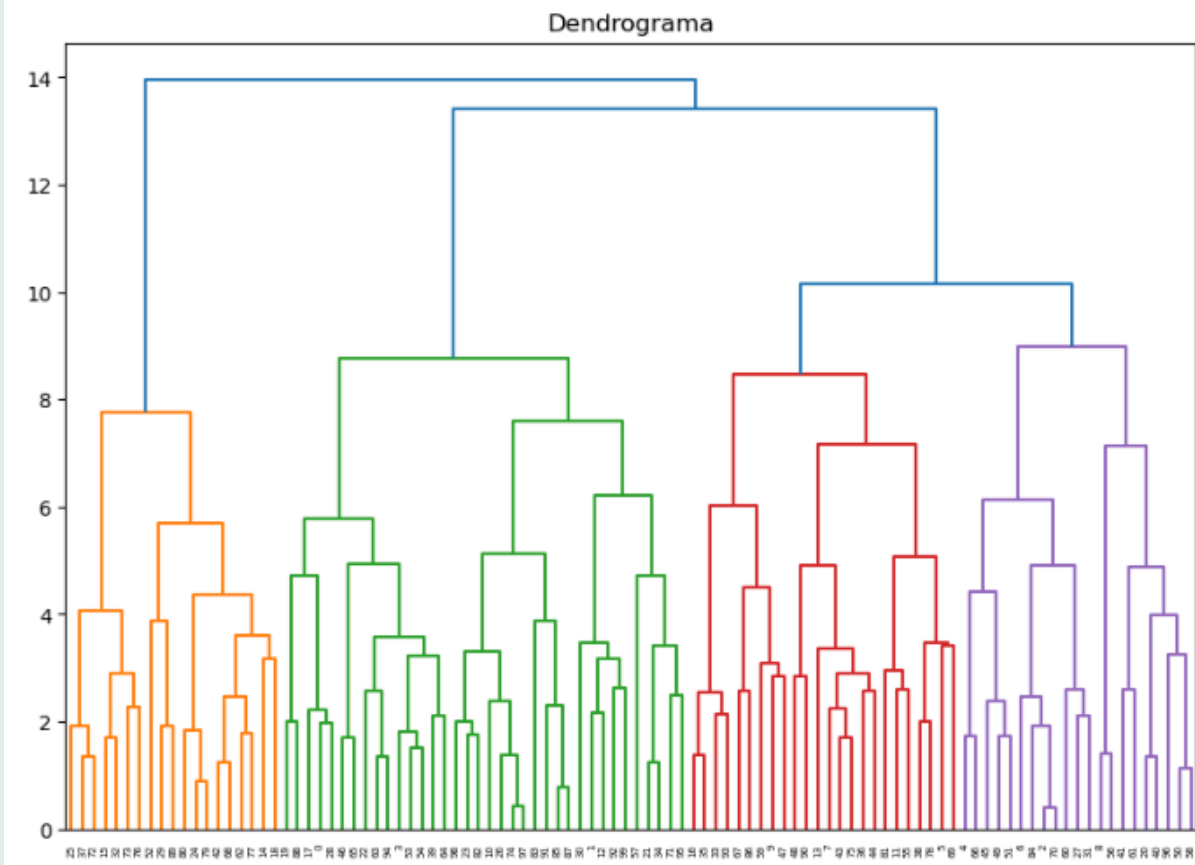
# CLUSTERING JERÁRQUICO

```
# Crear el algoritmo clustering jerárquico
import scipy.cluster.hierarchy as shc
from sklearn.preprocessing import StandardScaler

# Filtrar columnas categóricas y quedarse solo con numéricas
numeric_columns = data.select_dtypes(include=['float64', 'int64']).columns
data_numeric = data[numeric_columns]
```

```
# Escalar datos numéricos
scaler = StandardScaler()
data_scaled = scaler.fit_transform(data_numeric)
```

```
# Crear el dendograma
plt.figure(figsize=(10,7))
plt.title('Dendrograma') # Fixed typo in 'Dentograma'
dend = shc.dendrogram(shc.linkage(data_scaled, method='ward'))
plt.show()
```

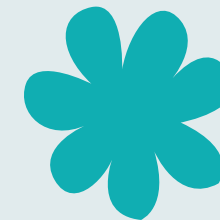
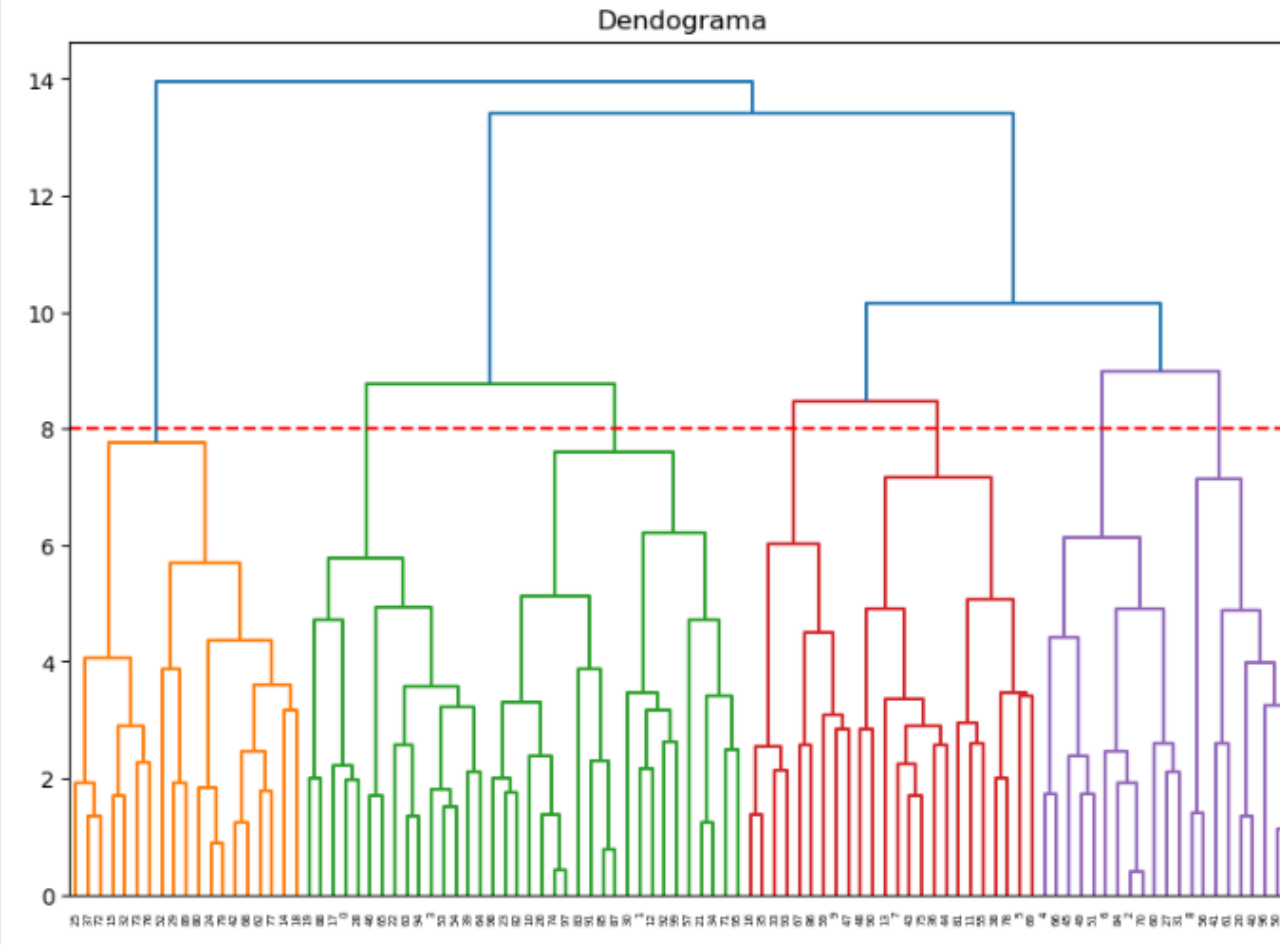


```
# Verificar los colores únicos
colores_unicos = set(dend['color_list'])
num_clusters_optimo = len(colores_unicos)-1
num_clusters_optimo
```

4

```
plt.figure(figsize=(10,7))
plt.title("Dendograma")
dend = shc.dendrogram(shc.linkage(data_scaled, method='ward'))
plt.axhline(y=8, color='r', linestyle='--')
```

<matplotlib.lines.Line2D at 0x2507dd1b750>



# ANÁLISIS PCA

```
# Análisis gráfico con PCA
campos = data_scaled
from sklearn import decomposition
pca = decomposition.PCA(n_components = 2)
pca.fit(campos)
campos = pca.transform(campos)
campos
```

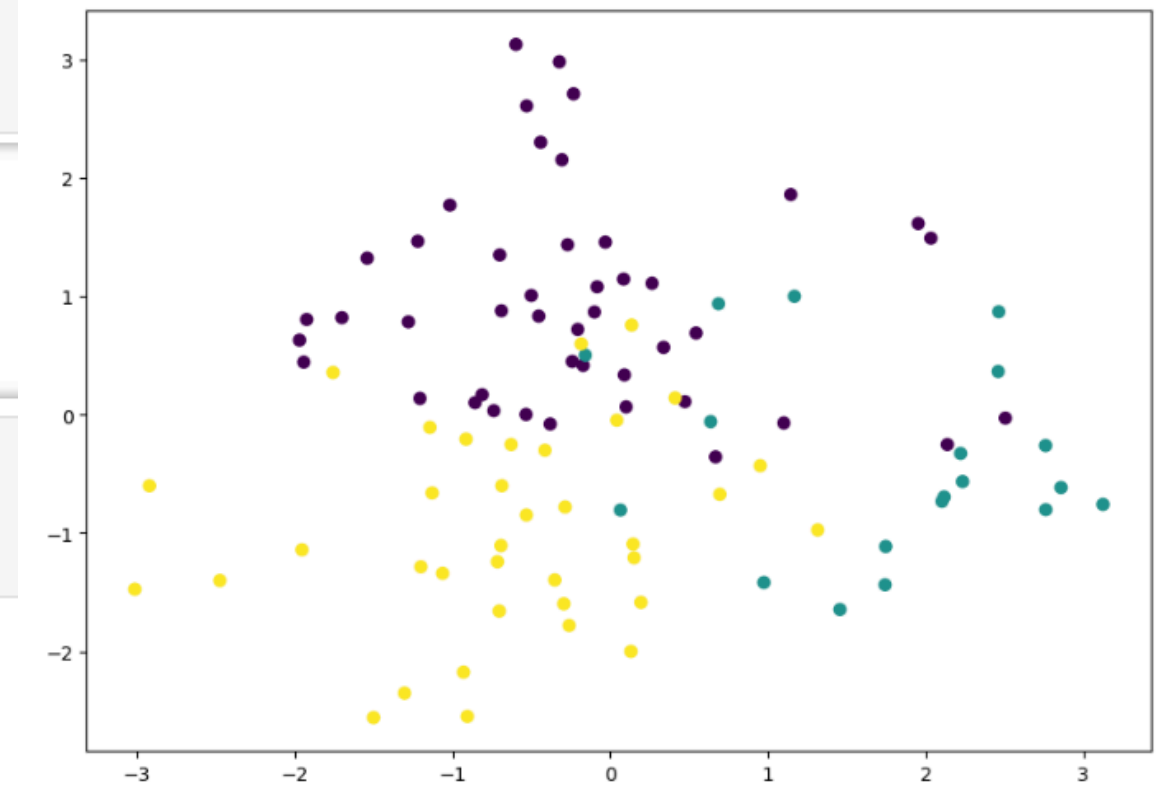
```
array([[ 1.94333789e-01, -1.58560889e+00],
       [ 1.50353542e-01, -1.20841395e+00],
       [ 3.37626029e-01,  5.69333744e-01],
       [ 1.30811356e-01, -2.00000504e+00],
       [-4.41908254e-01,  2.30292971e+00],
       [-1.70228918e+00,  8.21047543e-01],
       [ 8.89555116e-02,  3.36770020e-01],
       [-1.72653845e-01,  4.17650303e-01])
```

```
# Agregar una columna que permita etiquetar ¿qué elemento corresponde a qué grupo?
from sklearn.cluster import AgglomerativeClustering
cluster = AgglomerativeClustering(n_clusters=3, metric='euclidean', linkage='ward')
grupos = cluster.fit_predict(data_scaled)
grupos
```

```
array([[2, 2, 0, 2, 0, 0, 0, 0, 0, 0, 2, 0, 2, 0, 1, 1, 0, 2, 1, 2, 0, 2,
       2, 2, 1, 1, 2, 0, 2, 1, 2, 0, 1, 0, 2, 0, 0, 1, 0, 2, 0, 0, 1, 0,
       0, 0, 2, 0, 0, 0, 0, 0, 1, 2, 2, 0, 0, 2, 0, 0, 0, 0, 1, 2, 2, 2,
       0, 0, 1, 0, 0, 2, 1, 1, 2, 0, 1, 1, 0, 1, 1, 0, 2, 2, 0, 2, 0, 2,
       2, 1, 0, 2, 2, 0, 2, 2, 0, 2, 2, 2])
```

```
plt.figure(figsize = (10,7))
plt.scatter(campos[:,0], campos[:,1], c=cluster.labels_)
```

<matplotlib.collections.PathCollection at 0x2507db2a660>



# GRUPOS Y CONCATENAR DATAFRAMES

```
dataframe = pd.DataFrame(grupos, columns=['grupos'])
dataframe
```

	grupos
0	2
1	2
2	0
3	2
4	0
...	...
95	2
96	0
97	2
98	2
99	2

100 rows × 1 columns

```
# Concatenar dataframe's
dataframe2 = pd.concat([data, dataframe], axis=1, join = 'inner')
dataframe2
```

	Unnamed: 0	Velocidad Entrega	Precio	Durabilidad	Imagen Producto	Valor Educativo	Servicio Retorno	Tamano Paquete	Calidad Producto	Numero Estrellas	grupos
0	Adam	205	3	345	235	24	23	26	21	17	2
1	Anna	9	15	315	33	25	4	42	215	28	2
2	Bernard	17	26	285	3	43	27	41	26	33	0
3	Edward	135	5	355	295	18	23	39	195	17	2
4	Emilia	3	45	48	39	34	46	225	34	43	0
...	...	...	...	...	...	...	...	...	...	...	...
95	Teofan	3	8	32	25	7	21	42	17	1	2
96	Teofil	305	25	46	24	33	28	355	26	45	0

```
# Usar la primera columna como índice
dataframe2 = dataframe2.set_index(dataframe2.columns[0])
dataframe2
```

	Velocidad Entrega	Precio	Durabilidad	Imagen Producto	Valor Educativo	Servicio Retorno	Tamano Paquete	Calidad Producto	Numero Estrellas	grupos
Unnamed: 0										
Adam	205	3	345	235	24	23	26	21	17	2
Anna	9	15	315	33	25	4	42	215	28	2
Bernard	17	26	285	3	43	27	41	26	33	0
Edward	135	5	355	295	18	23	39	195	17	2
Emilia	3	45	48	39	34	46	225	34	43	0
...	...	...	...	...	...	...	...	...	...	...
Teofan	3	8	32	25	7	21	42	17	1	2
Teofil	305	25	46	24	33	28	355	26	45	0



# FILTRAR CLUSTERS PARA SU ANÁLISIS

# Filtrar por grupos 0

```
rows_group_0 = dataframe2[dataframe2['grupos'] == 0]
print(rows_group_0)
```

	Calidad Producto	Numero Estrellas	grupos
Unnamed: 0			
Bernard	26	33	0
Emilia	34	43	0
Fabian	22	3	0
Philip	29	31	0
Frank	215	29	0
Xavier	27	48	0
Gabriel	27	39	0
Henry	25	32	0
Isabelle	25	23	0
Eugene	22	23	0
Evdokia	225	26	0
Florence	29	39	0
Hilary	25	38	0
Isadore	19	32	0
Jacob	18	24	0
Jeremiah	24	29	0
Joshua	105	14	0

# Filtrar por grupos 1

```
rows_group_1 = dataframe2[dataframe2['grupos'] == 1]
print(rows_group_1)
```

	Calidad Producto	Numero Estrellas	grupos
Unnamed: 0			
Isidore	295	39	1
Joseph	235	34	1
Eunice	295	39	1
Flavia	245	34	1
Flora	295	34	1
Helen	255	31	1
Lourdes	305	45	1
Jervis	255	31	1
Judith	295	38	1
Louise	275	35	1
Markian	265	32	1
Maura	215	43	1
Methodius	245	36	1
Michael	255	45	1
Monica	255	27	1
Mukuta	255	32	1

# Filtrar por grupos 2

```
rows_group_2 = dataframe2[dataframe2['grupos'] == 2]
print(rows_group_2)
```

	Calidad Producto	Numero Estrellas	grupos
Unnamed: 0			
Adam	21	17	2
Anna	215	28	2
Edward	195	17	2
Marisol	215	17	2
Irene	22	24	2
Eugenia	28	25	2
Eva	3	4	2
Fedir	165	2	2
Felix	26	4	2
Fialka	185	21	2
Florent	185	21	2
Hannah	27	34	2
Herman	165	28	2
Ivan	205	2	2
John	165	13	2
Lawrence	245	4	2

- FILTRÉ LOS CLUSTERS PARA UBICAR LOS NOMBRES Y CARACTERÍSTICAS, DE MANERA QUE PUDIERA OFRECER LAS CONCLUSIONES SOLICITADAS.

# CONCLUSIONES

“RECOMENDARÍA A MARIANNA LOS MISMOS PRODUCTOS QUE COMPRÓ JOSEPHINE PORQUE PARA AMBAS IMPORTA MUCHO LA VELOCIDAD DE ENTREGA , LA CALIDAD DEL PRODUCTO Y EL TAMAÑO DEL PAQUETE, CALIFICANDO DE MANERA SIMILAR SUS COMPRAS, TIENEN ALTA PROBABILIDAD DE COMPARTIR LAS MISMAS NECESIDADES, INTERESES Y GUSTOS”.

“RECOMENDARÍA A STEPHANIA LOS MISMOS PRODUCTOS QUE COMPRÓ LOURDES PORQUE PARA AMBAS IMPORTA MUCHO LA IMAGEN DEL PRODUCTO, LA DURABILIDAD Y LA CALIDAD DEL PRODUCTO Y POR SER SIMILARES EN SU PUNTAJE TIENEN MUY ALTA PROBABILIDAD DE COMPARTIR LAS MISMAS NECESIDADES, INTERESES Y GUSTOS”.

RECOMENDARÍA A STEPHAN LOS MISMOS PRODUCTOS QUE COMPRÓ HANNAH PORQUE PARA LOS DOS IMPORTA LA VELOCIDAD DE ENTREGA, LA DURABILIDAD Y LA IMAGEN DEL PRODUCTO; PUEDE SER QUE TENGAN SIMILITUD EN NECESIDADES, GUSTOS E INTERESES.”.

