

Grupo 3

Algoritmos de Aprendizaje **No Supervisado**



Tabla de Contenidos

01

Análisis de las Variables

Datos interesantes extraídos del análisis realizado

02

Agrupación de Películas

Utilizando k-means, agrupación jerárquica y Red de Kohonen

03

Predictión de Géneros

Utilizando k-means, agrupación jerárquica y Red de Kohonen

04

Conclusiones

Análisis de conocimientos adquiridos y puntos a destacar



01

Análisis de Variables

Y puntos interesantes a destacar





Pre-Procesamiento del DataSet

Análisis Inicial

- Filas totales: 5505
- Remoción de “imdb_id” no nulos duplicados → 265



Valores Nulos por Característica

- Budget: 38
- Genre: 36
- imdb_id: 45
- Original Title: 35
- Overview: 42
- Popularity: 38
- Production Companies: 27
- Production Countries: 25
- Release Date: 31
- Revenue: 30
- Runtime: 32
- Spoken Languages: 43
- Vote Average: 40
- Vote Count: 37

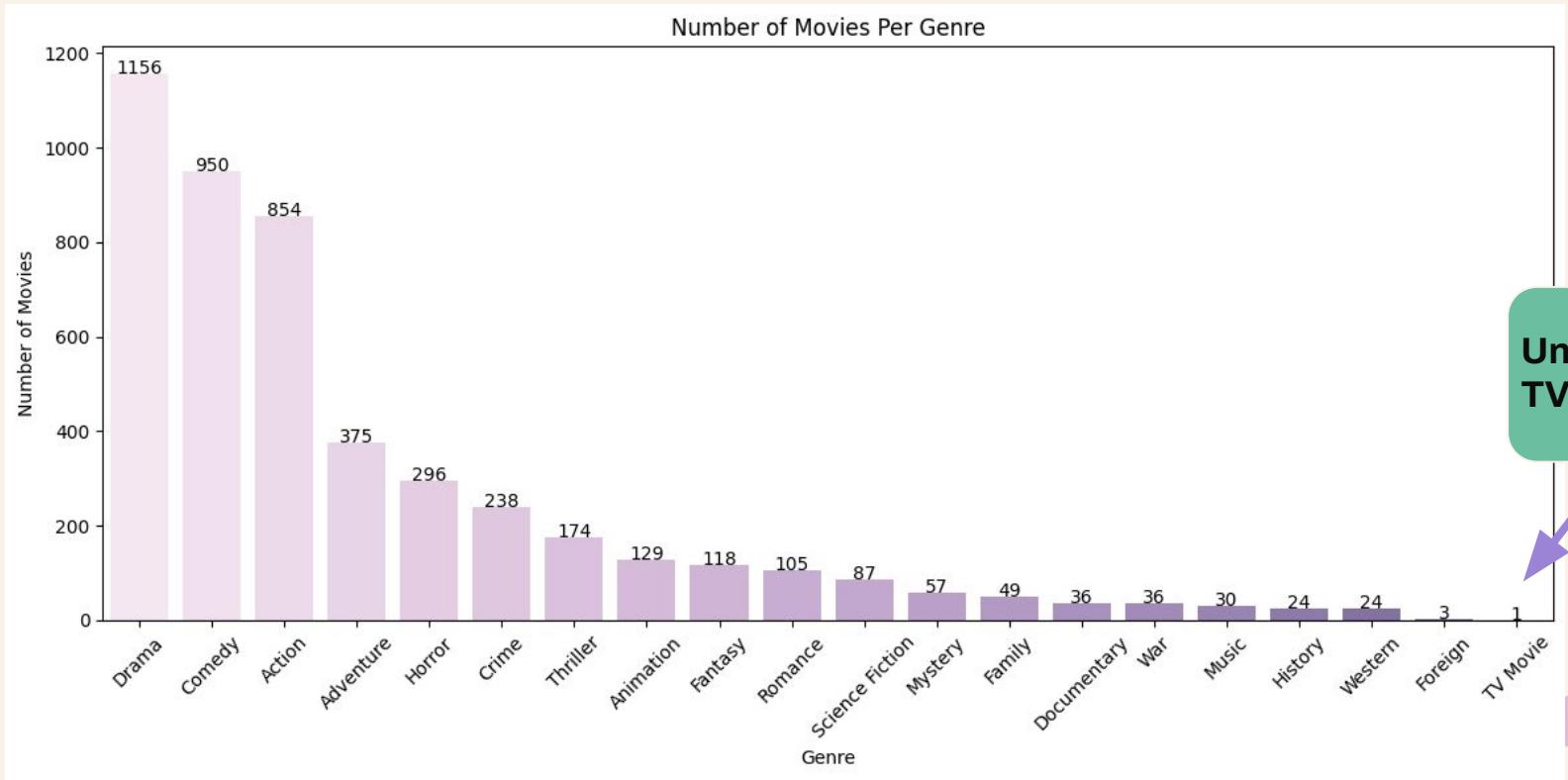
En total, se eliminaron 498 filas con valores nulos.

Cantidad de filas finales: 4742





Análisis de Géneros

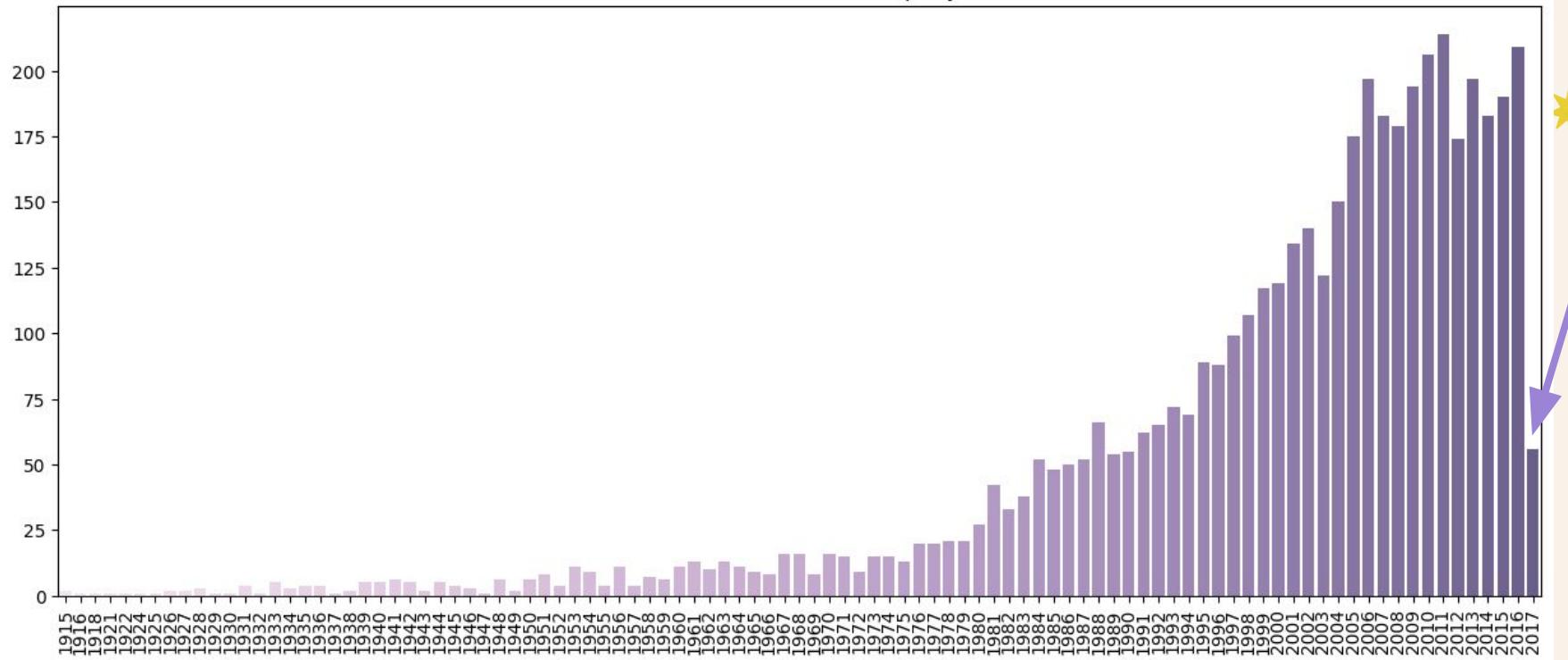




¡Los datos de
2017 están
incompletos!

Análisis de Estrenos por año

Number of movies released per year



Pre-Procesamiento del DataSet

Observaciones

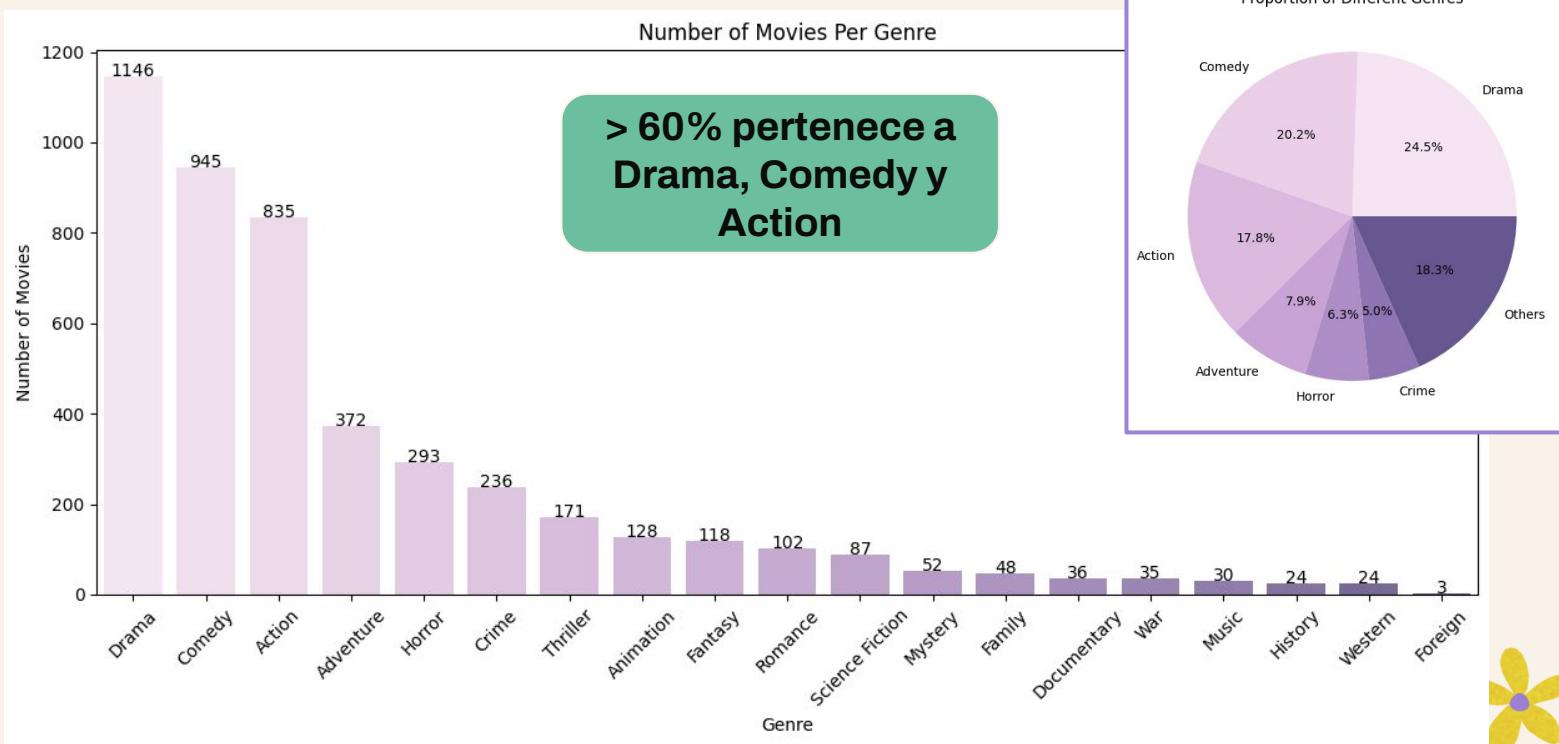
- Hay una única TV-Movie
- El año 2017 se encuentra **incompleto**
- Entre ambas encontramos 57 filas

Decidimos eliminar estos valores dado que
no son significativos en el **dataset** y el **análisis**.

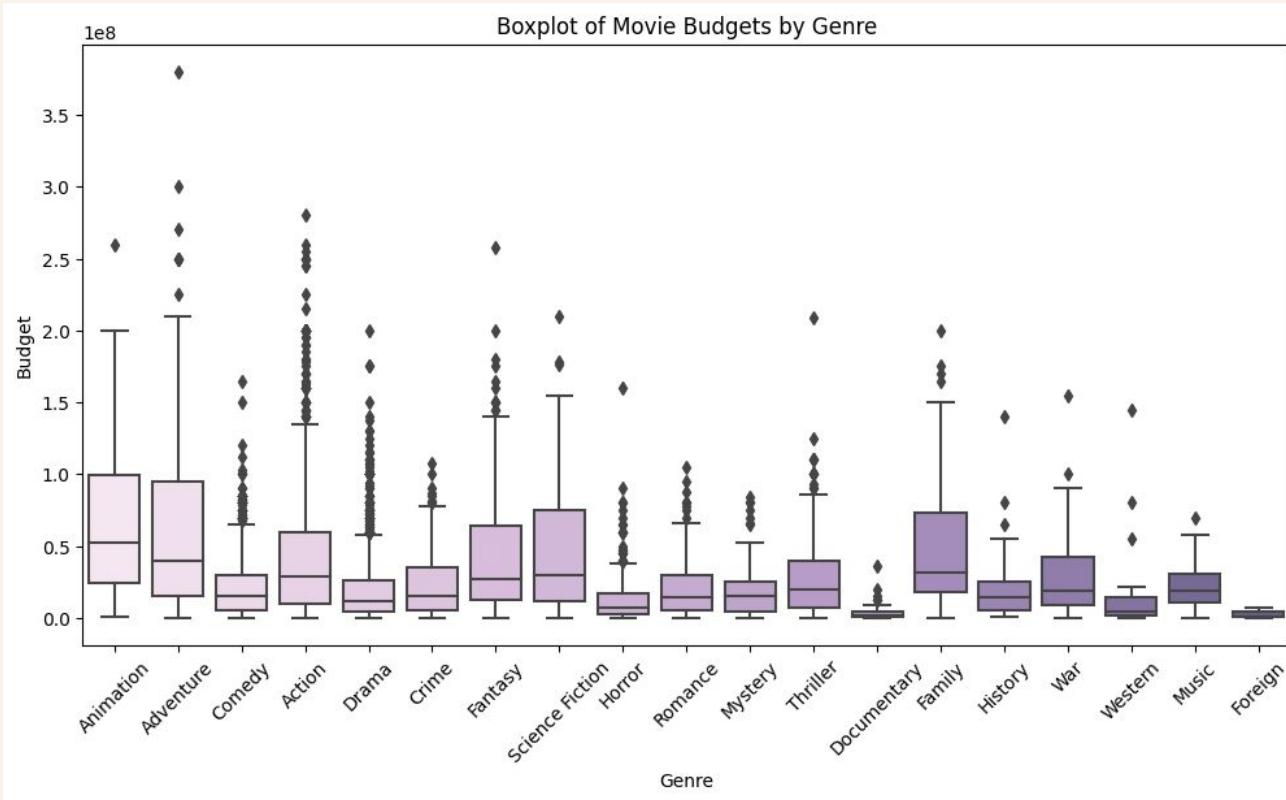
Ejecutamos nuevamente el análisis de variables **sin** estos valores

Cantidad de filas finales: 4685

Análisis del DataSet - Genres



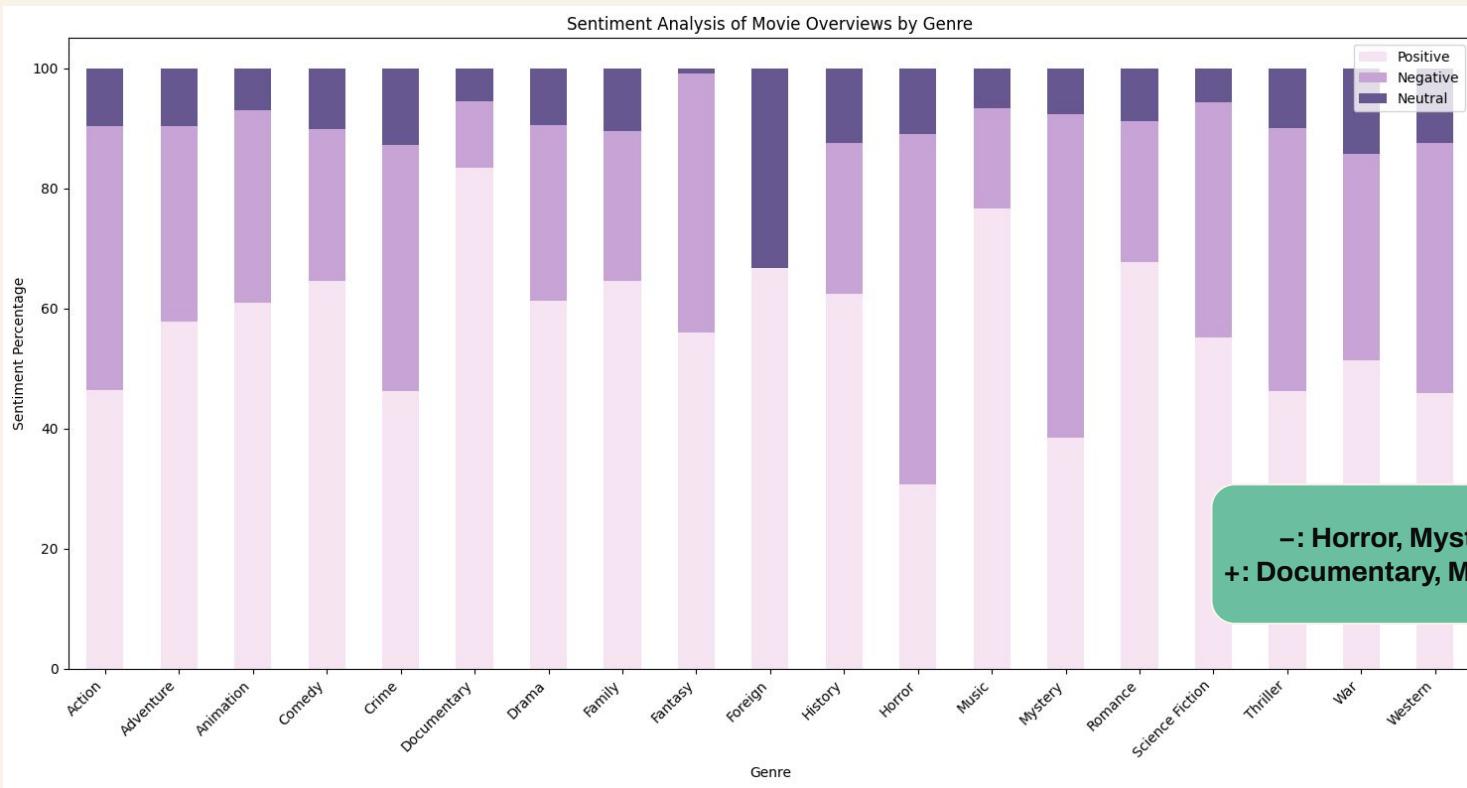
Análisis del DataSet - *Budget*



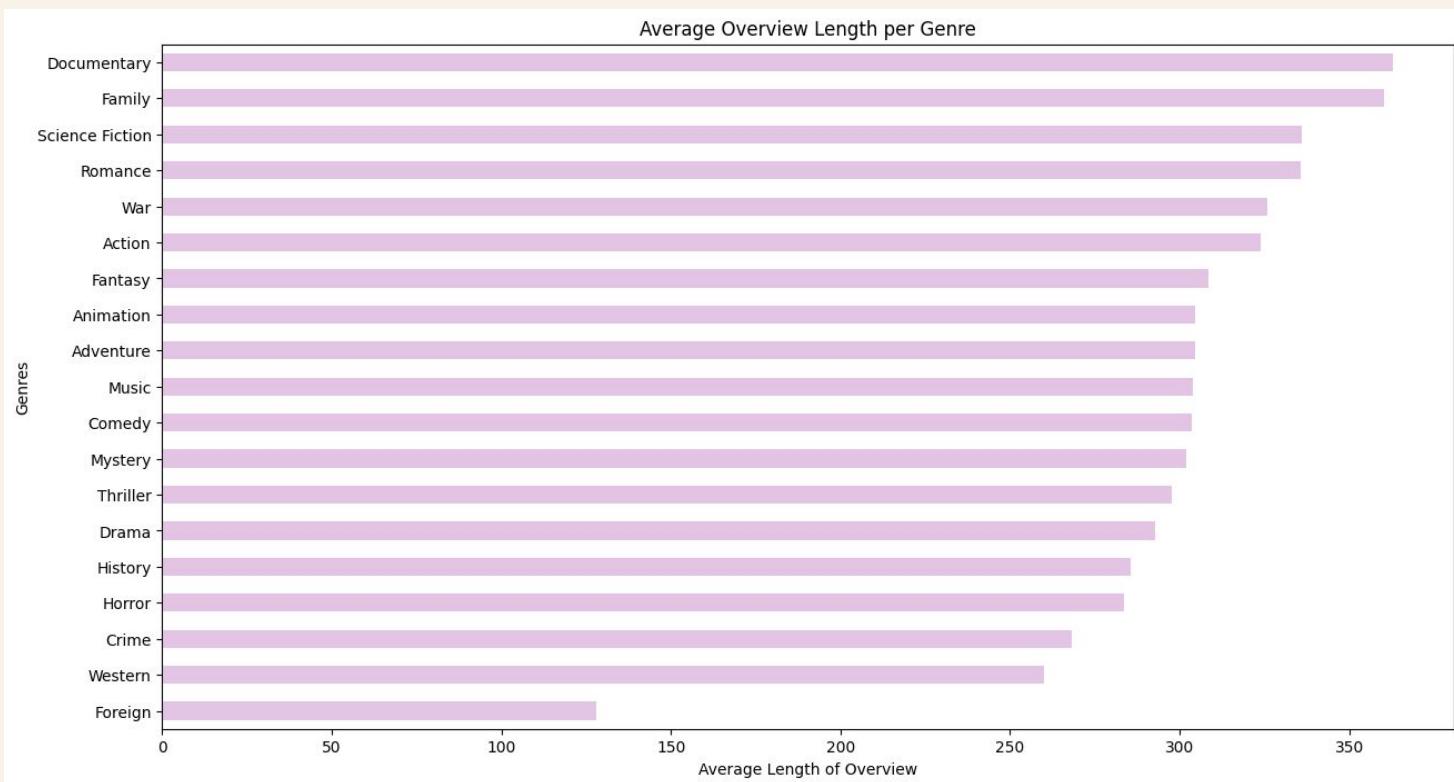
Análisis del DataSet - *Overviews*



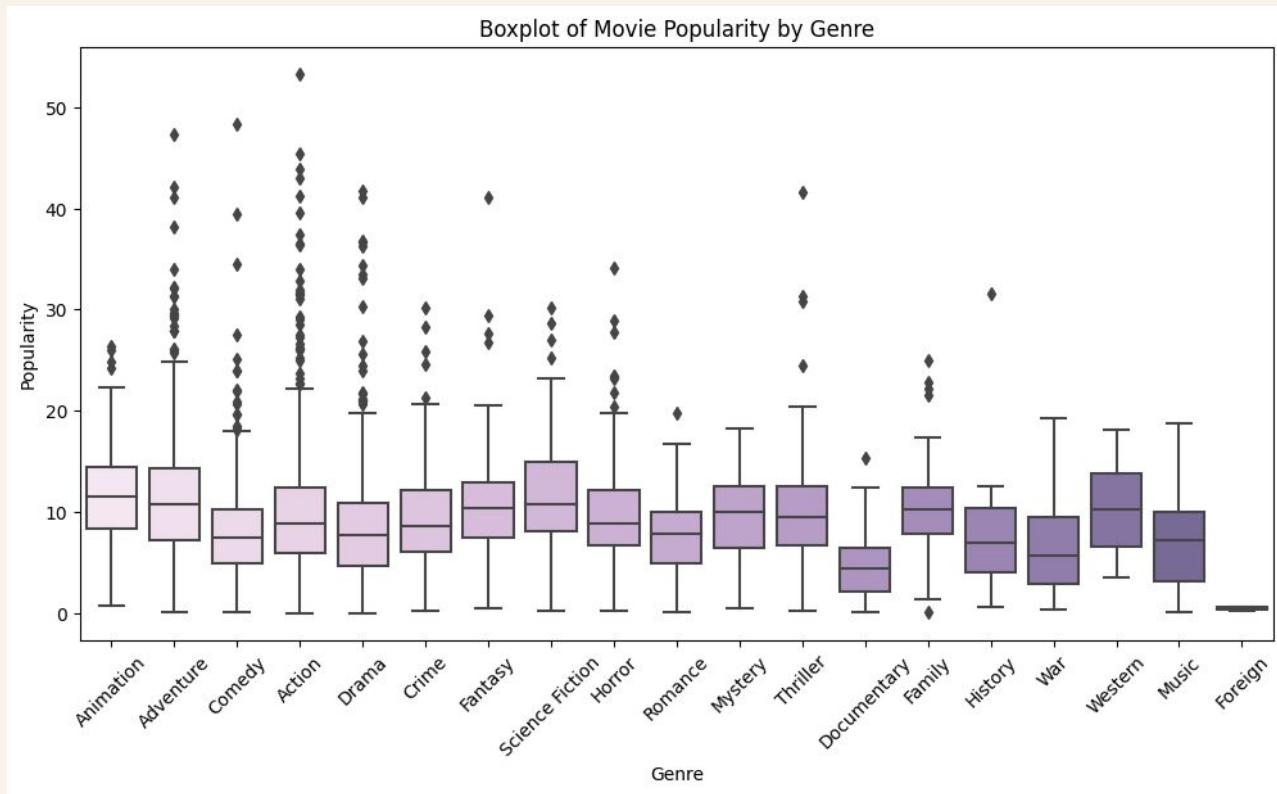
Análisis del DataSet - *Overviews*



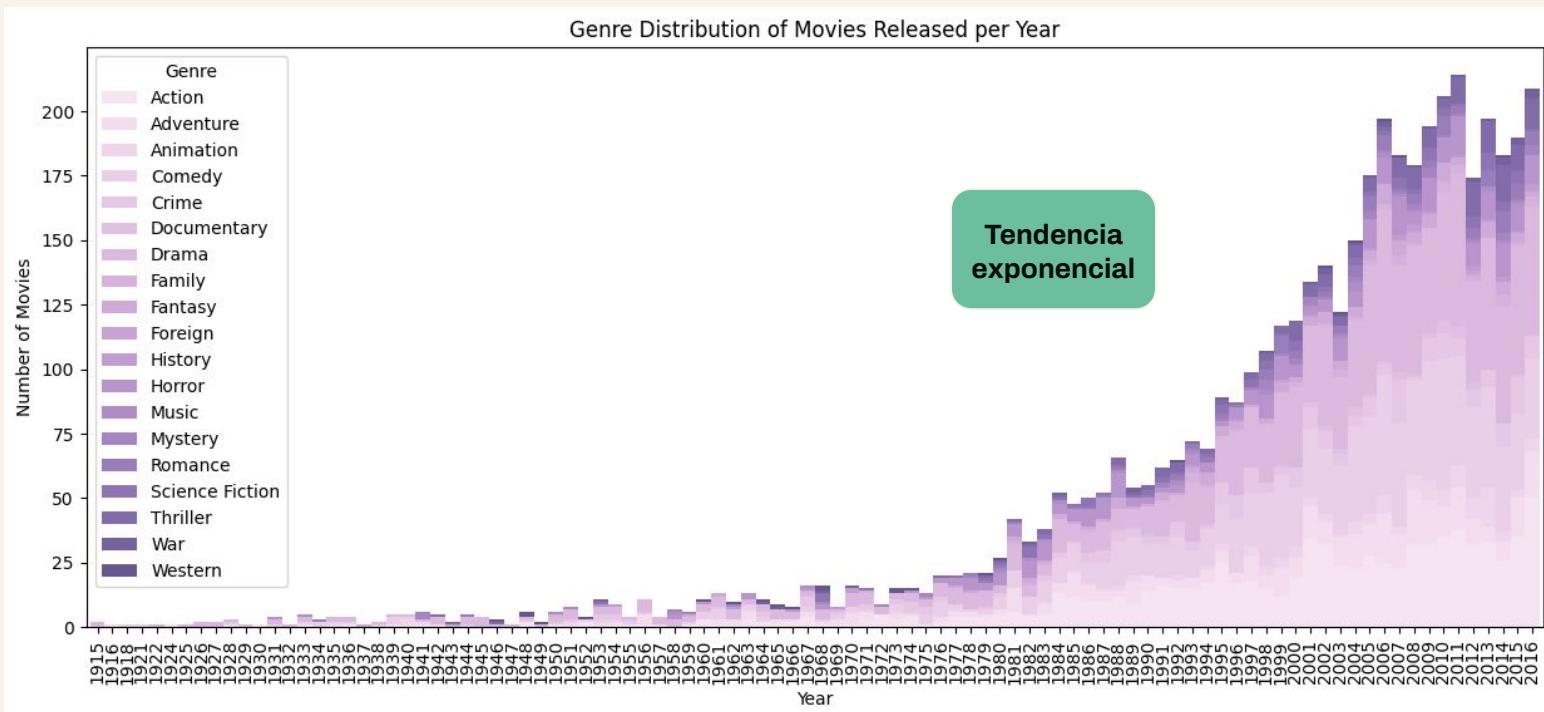
Análisis del DataSet - *Overviews*



Análisis del DataSet - *Popularity*

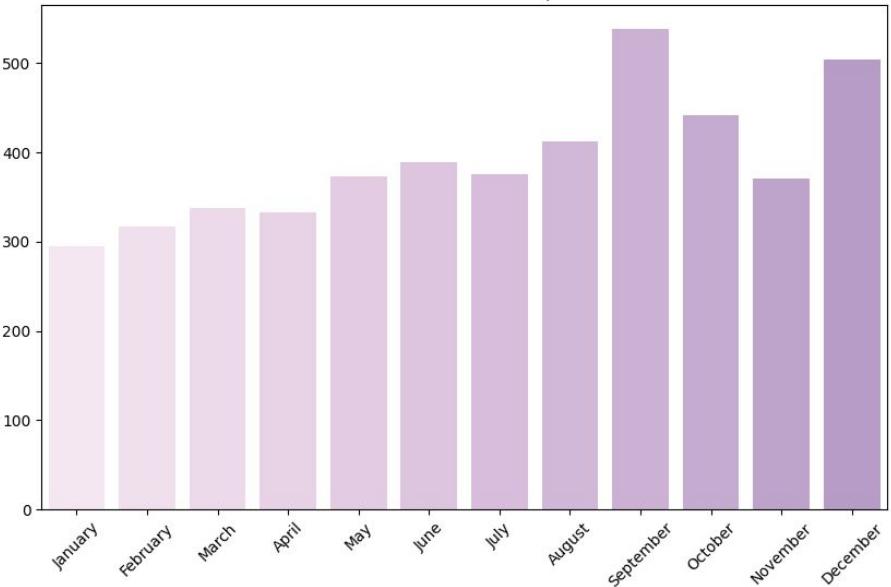


Análisis del DataSet - *Release date*

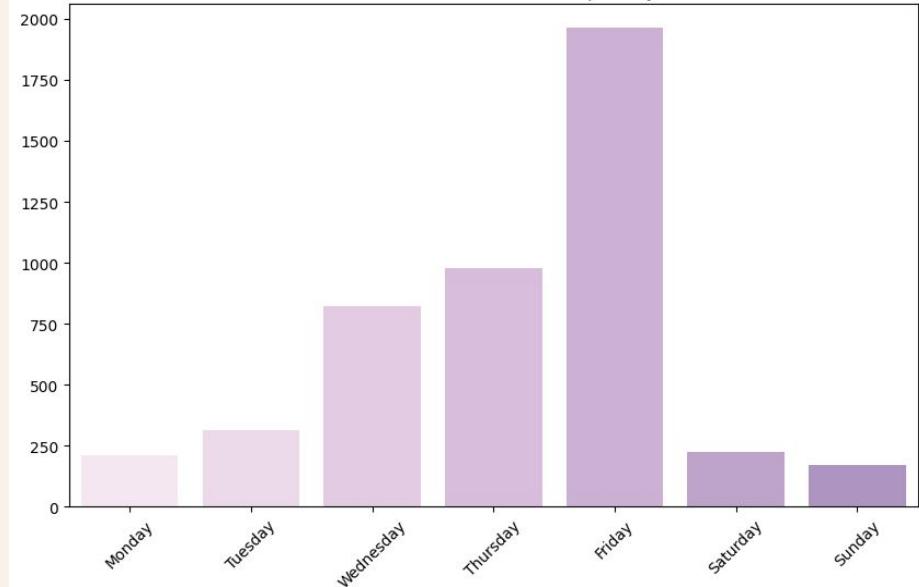


Análisis del DataSet - *Release date*

Number of movies released per month

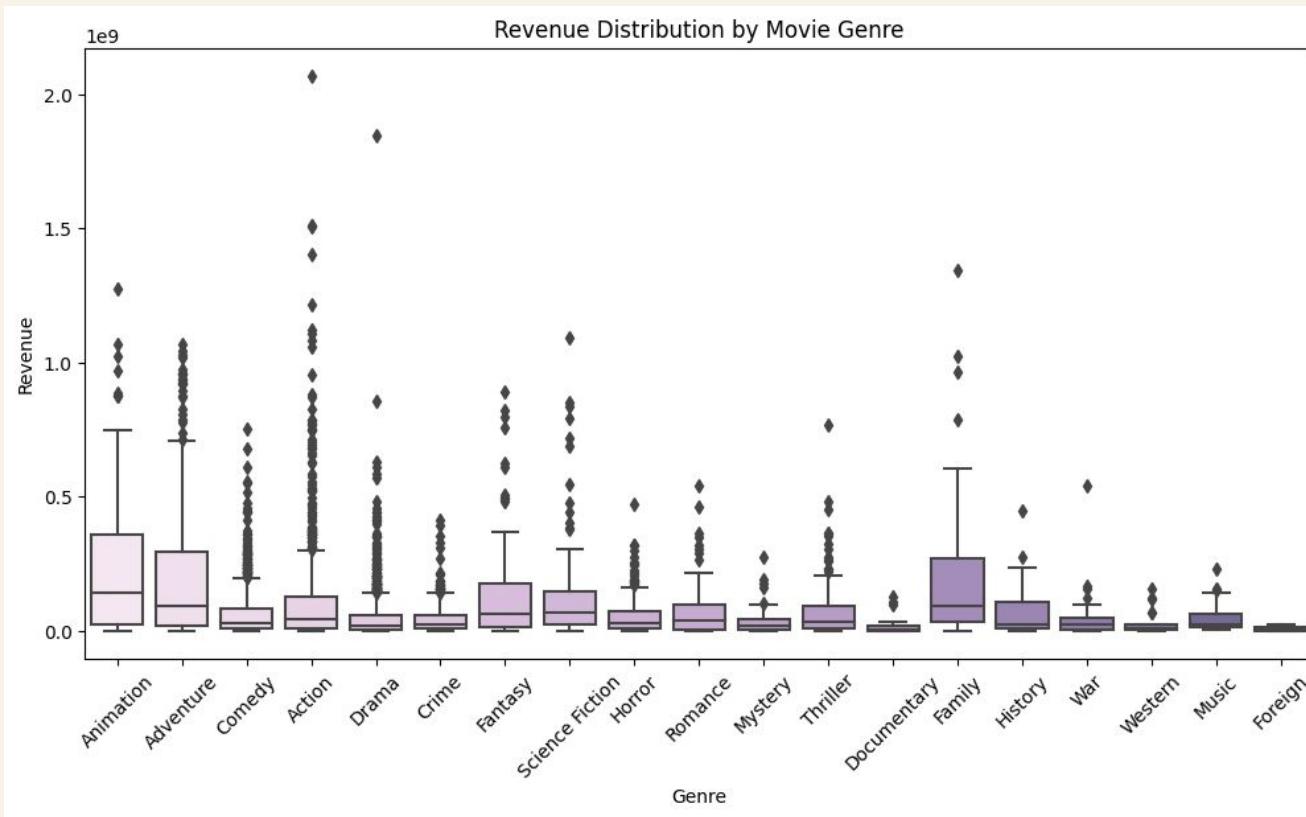


Number of movies released per day



Mes → Septiembre
Día → Viernes

Análisis del DataSet - Revenue



02

Agrupación de Películas

Mediante Métodos No Supervisados





K Means

- Algoritmo **no** supervisado
- Construye una partición de las observaciones en **K conjuntos** que minimiza la distancia de los elementos dentro de cada grupo
 - Agrupa en **K conjuntos** no solapados
 - Un **buen agrupamiento** es aquel que la variación dentro de un mismo clúster es pequeña



¿Cómo medimos la variación?

$$W(C_k) = \frac{1}{|C_k|} \sum_{i,j \in C_k} \sum_{l=1}^p (x_{il} - x_{jl})^2$$

- **Factor W**, es la suma de las distancias al cuadrado de cada punto de datos del centroide.
 - A menor W más cerca se encuentran los puntos. → Lo queremos **minimizar**

¡K Medias es un problema de optimización no lineal!





K Means

Algoritmo

1. Asignamos aleatoriamente un número de 1 a K a cada una de las observaciones
2. Realizar los siguientes pasos hasta que la asignación de clusters se mantenga **estable** entre las iteraciones:
 - a. Para cada clase, se calcula el **centroide**
 - b. Se asigna cada observación al cluster cuyo centroide está más cerca en **distancia euclídea**
3. El algoritmo termina cuando se mantiene estable durante varias iteraciones, se está en presencia de un **mínimo local**, no necesariamente un **mínimo global**

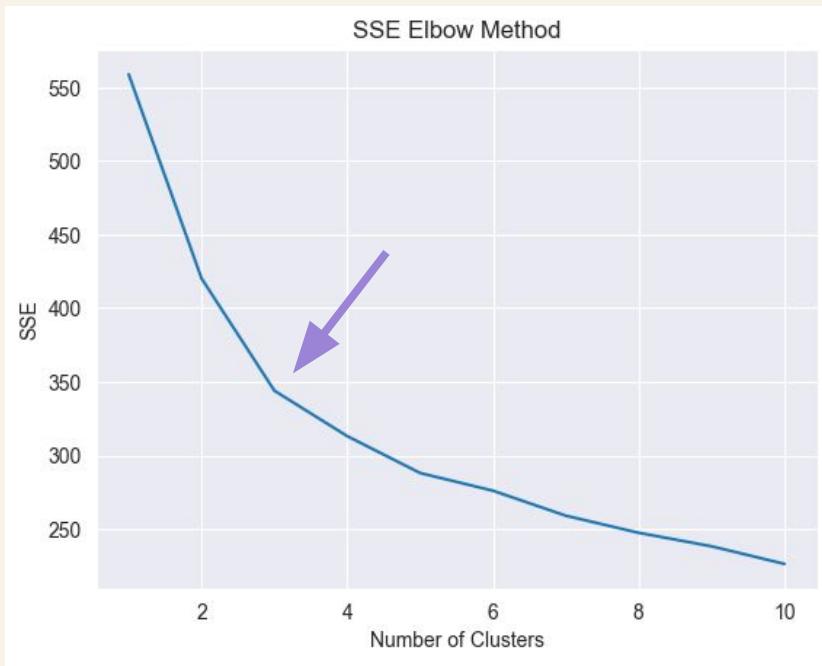
Los resultados dependen mucho de los valores de los centroides elegidos inicialmente!



Hay que definir K a priori teniendo en cuenta esto, ¿cómo?

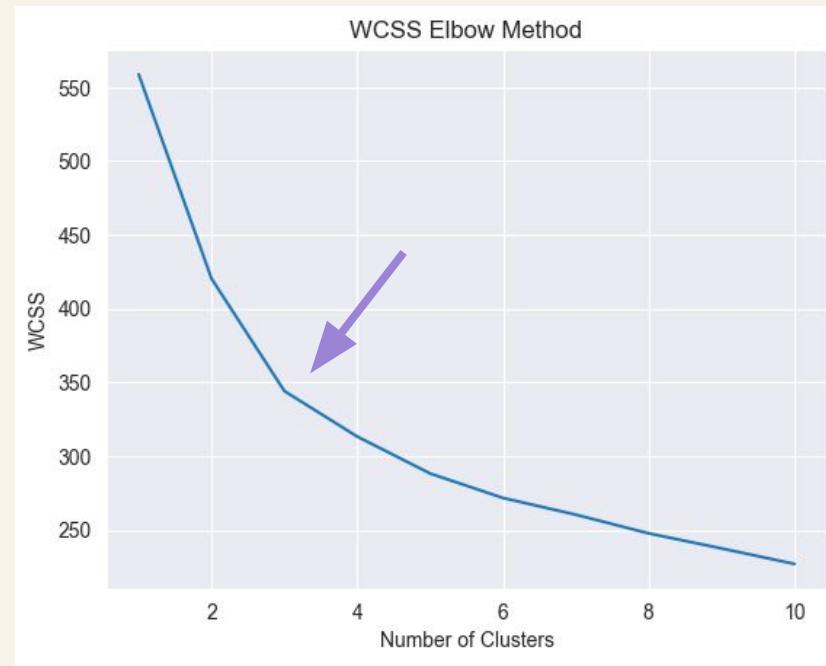


Método del Codo



Diferencia entre nodos y su centroide
Sum of the Square Error

K = 3



Diferencia entre nodos de un clúster
Within-Cluster Sum of the Square Error



Agrupamiento Jerárquico

- Algoritmo **no** supervisado
- Organiza puntos de datos en una jerarquía de clústeres basados en su **similitud** o **distancia**.



Algoritmo

1. Se inicia con **N** grupos (uno por punto)
2. Se calcula la distancia entre cada uno de los clusters/grupos
3. Se **unen** los grupos que tienen la menor distancia entre sí (clústeres)
4. Si queda más de un grupo o de la cantidad de grupos deseada, se vuelve al **paso 2**.



Pero... ¿cómo calculamos la distancia entre clústeres?



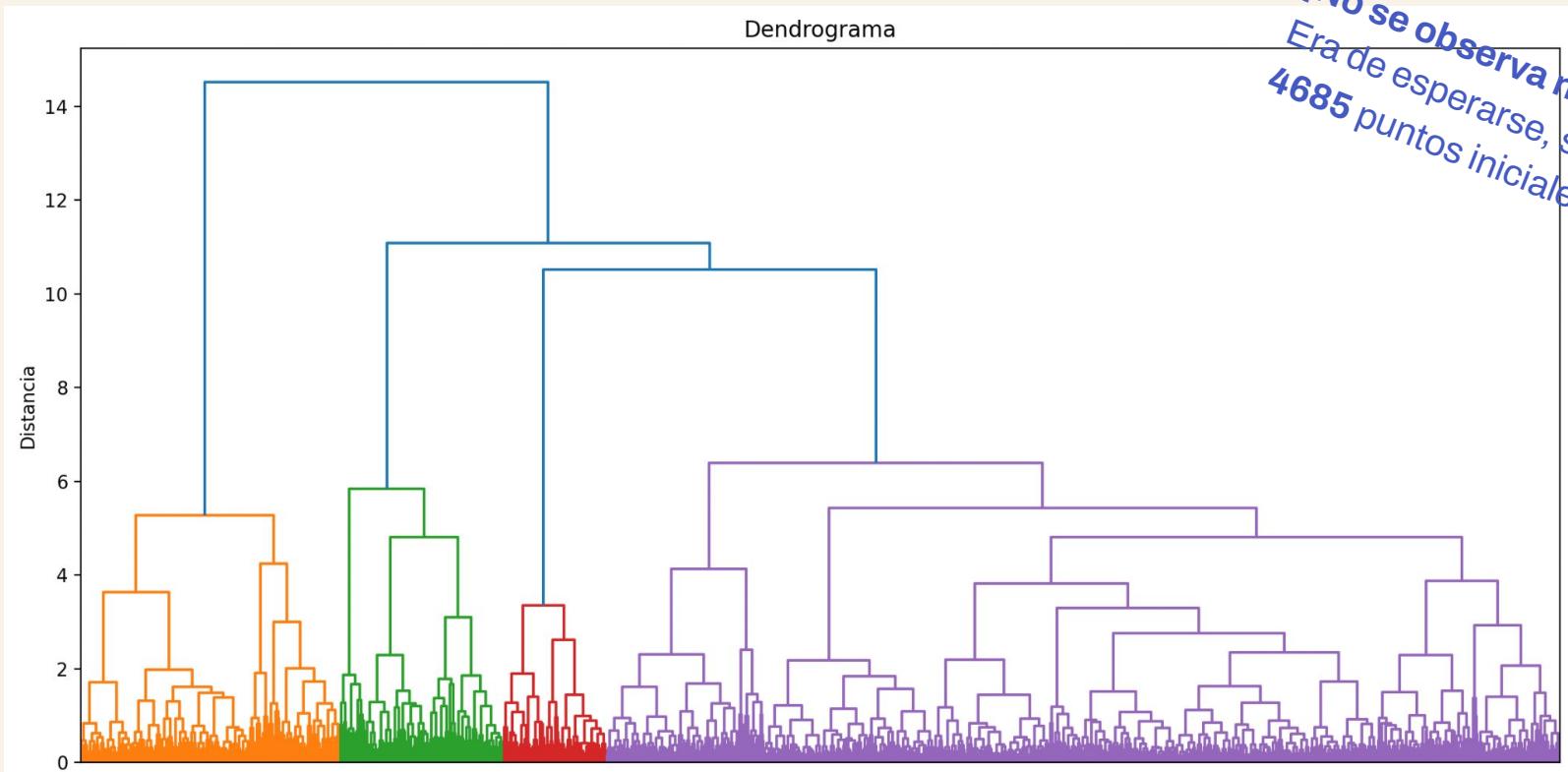
Distancia entre clústeres

- Para el cálculo de distancias llevamos una matriz simétrica de distancias, con diagonales nulas
- Solamente consideramos el triángulo inferior, dado que la misma se encuentra espejada,
- Hay distintas medidas de similitud:
 - Máxima**: la distancia máxima posible entre dos puntos del clúster
 - Mínima**: la distancia mínima posible entre puntos de clúster
 - Promedio**: entre todos los puntos de cada clúster,
 - Centróide**: se toma la distancia entre los puntos medios de los clústeres.
- Estandarizamos las variables para que sean **comparables**
- Utilizamos la **distancia euclídea** para medir la distancia entre los clústeres.

¡la que utilizamos!



Dendrograma Obtenido





Análisis de Agrupamiento Jerárquico

¿Qué pasa en cada nivel? ¿Qué se agrupa?

1. Primer cluster seleccionado aleatoriamente para analizar:

	budget	genres	imdb_id	original_title	...	spoken_languages	vote_average	vote_count	year
16000000.0	Comedy	tt0114885		Waiting to Exhale	...	1.0	6.1	34.0	1995.0
20000000.0	Comedy	tt0120703	How Stella Got Her Groove Back		...	1.0	6.0	24.0	1998.0

2. Se une luego con el clúster

	budget	genres	imdb_id	original_title	...	spoken_languages	vote_average	vote_count	year
12000000.0	Comedy	tt0116414		Girl 6	...	1.0	5.7	17.0	1996.0
15000000.0	Comedy	tt0120772	The Object of My Affection		...	1.0	5.6	75.0	1998.0

3. Y luego con el clúster

	budget	genres	imdb_id	original_title	overview	...	runtime	spoken_languages	vote_average	vote_count	year
30000000.0	Comedy	tt0118798	Bulworth	A suicidally disillusioned liberal politician	108.0	1.0	6.3	66.0	1998.0
31000000.0	Comedy	tt0100140	Mermaids	Fifteen-year-old Charlotte Flax is tired of he...	110.0	1.0	6.5	124.0	1990.0

Y así continúa clusterizando (hasta llegar a un único cluster)

Redes de Kohonen (SOM)

- Algoritmo **no** supervisado
- Grilla de **KxK** neuronas conectadas entre sí
- Asocia los datos del conjunto de entrenamiento a neuronas de la grilla

Algoritmo

1. Se inicializa una grilla de **KxK** con pesos al azar entre -1 y 1 o con datos del conjunto
2. Se define un radio de influencia y una constante de aprendizaje por iteración
3. Se obtiene la neurona cuyos pesos más se asemeje a un ejemplo seleccionado del conjunto de entrenamiento
4. Se actualizan los vecinos de dicha neurona según el radio de influencia



Parámetros Redes de Kohonen (SOM)

Inicialización de pesos

- Entre -1 y 1
- Con datos del conjunto

Tipo de entrenamiento

- Random shuffle
- Estocástico

Tipo de grilla

- Rectangular
- Hexagonal

Radio de vecindad

- Inversamente proporcional a la iteración
- Exponencialmente decae con la iteración

Tasa de aprendizaje

- Fija (0.1)
- Inversamente proporcional a la iteración



Parámetros Redes de Kohonen (SOM)

¿Cómo medimos que tan buena es una Red?

Usamos el siguiente criterio:

1. Suma de las distancias a la neurona ganadora para cada dato del conjunto de entrenamiento

Este es el criterio que intentamos **minimizar** con las diferentes arquitecturas.

Parámetros Redes de Kohonen (SOM)

Inicialización de pesos

- Entre -1 y 1
- Con datos del conjunto

Tipo de entrenamiento

- Random shuffle
- Estocástico

Tipo de grilla

- Rectangular (4x4)

Radio de vecindad

- Inversamente proporcional a la iteración
- Exponencialmente decae con la iteración

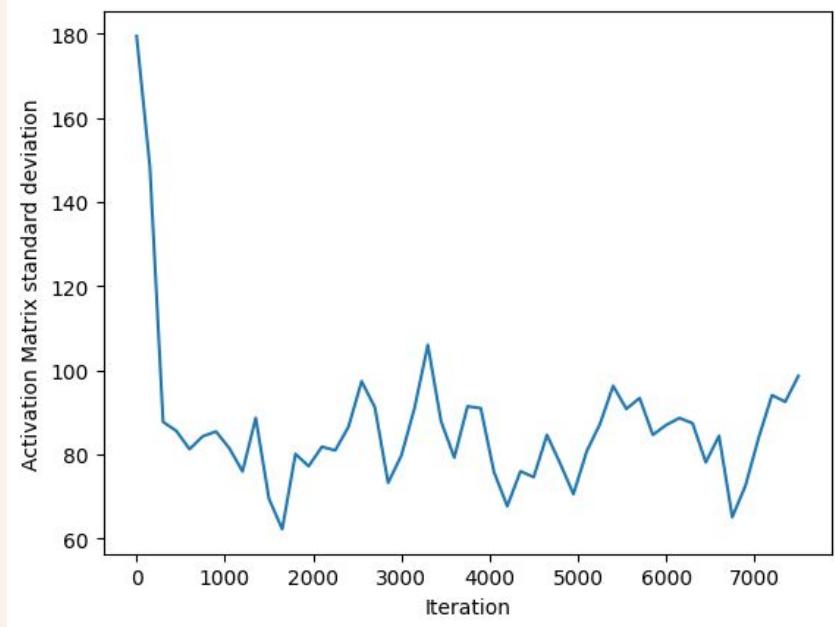
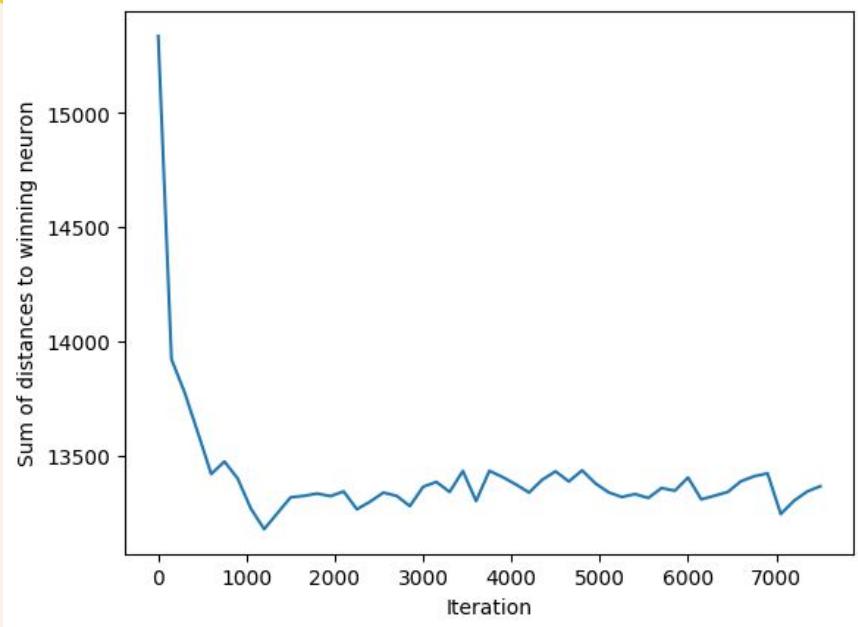
Tasa de aprendizaje

- Fija (0.1)
- Inversamente proporcional a la iteración

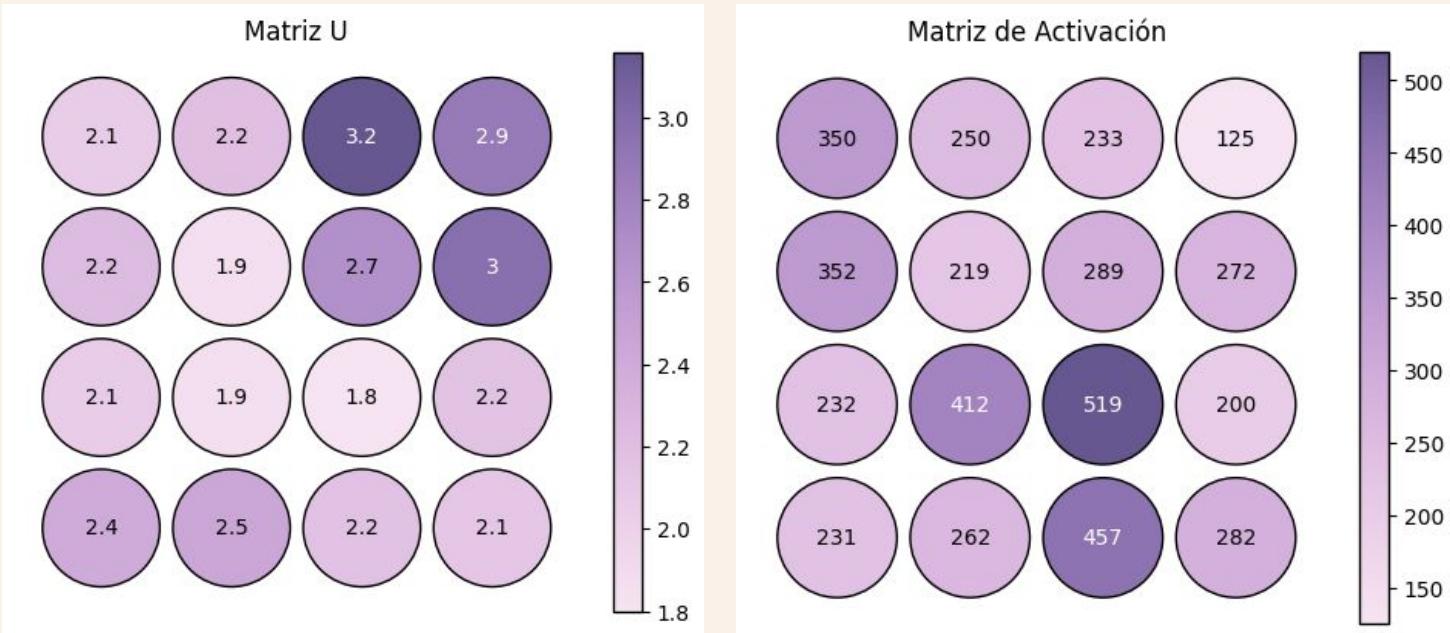
Iters = 15*500

D(min) = 12.373
D(max) = 12.754
D(rango) = 380

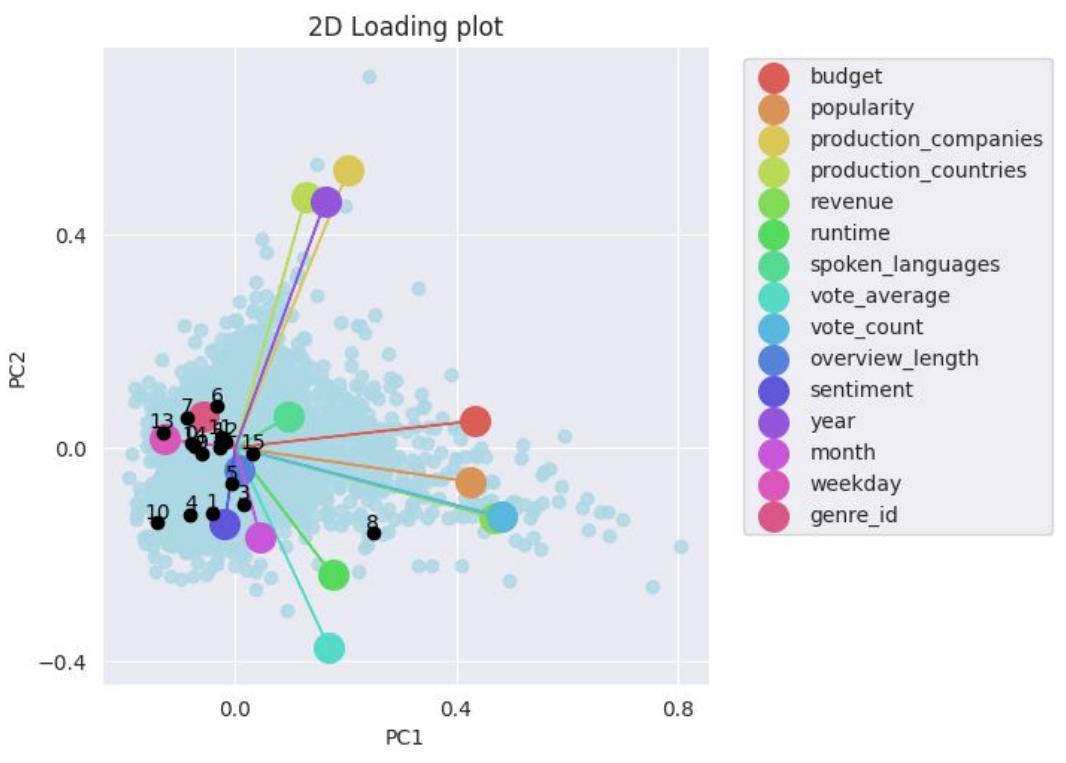
Métricas de agrupamiento



Matriz U y Matriz de Activación



Pesos seleccionados

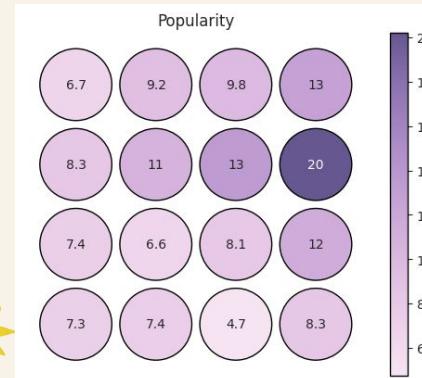
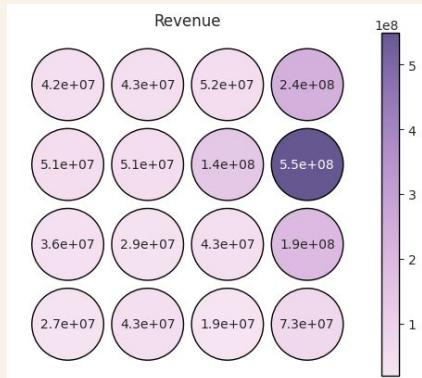
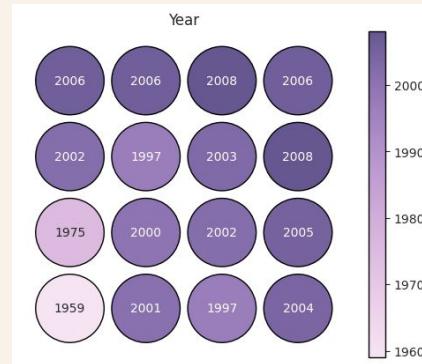
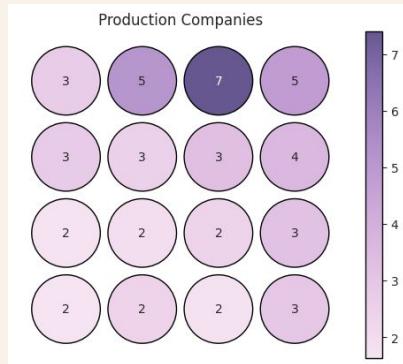
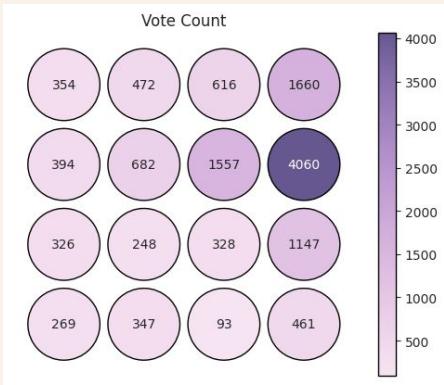


Neuron	
0	(0, 1)
1	(1, 0)
2	(0, 2)
3	(2, 2)
4	(1, 2)
5	(1, 3)
6	(2, 0)
7	(3, 3)
8	(2, 3)
9	(3, 1)
10	(0, 0)
11	(0, 3)
12	(1, 1)
13	(2, 1)
14	(3, 0)
15	(3, 2)

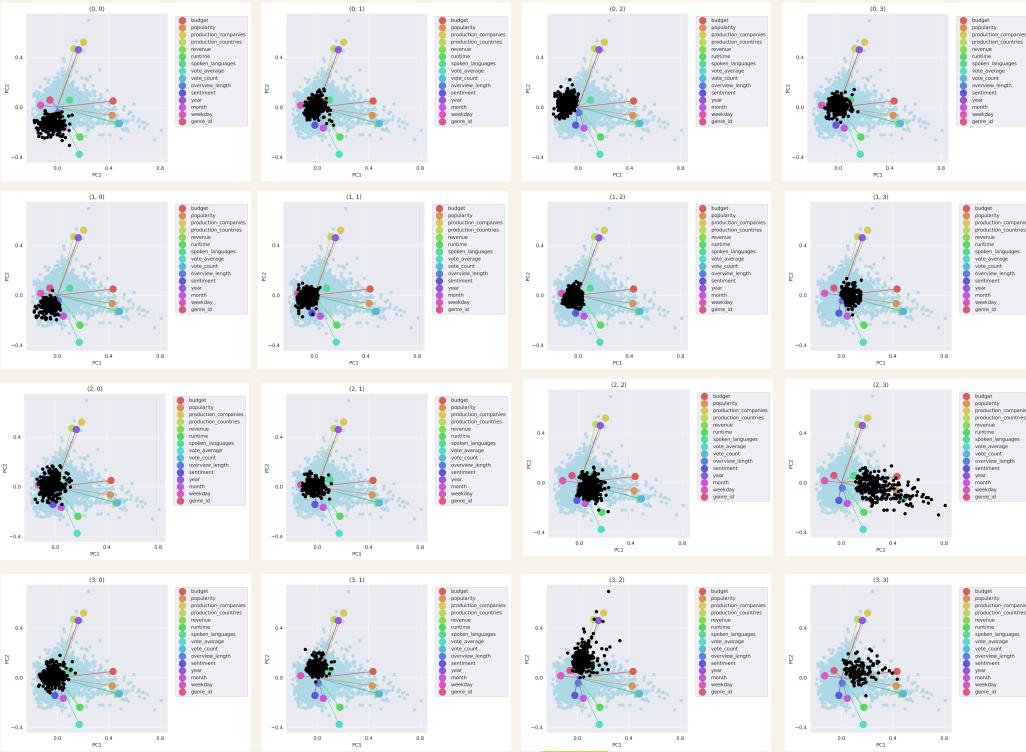
feature_names	PC1	PC2
budget	0.433040	0.049788
popularity	0.426776	-0.064137
production_companies	0.204604	0.520394
production_countries	0.127954	0.469433
revenue	0.469984	-0.131339
runtime	0.175837	-0.238604
spoken_languages	0.095781	0.057398
vote_average	0.170001	-0.376321
vote_count	0.482881	-0.130212
overview_length	0.006855	-0.043471
sentiment	-0.019059	-0.142992
year	0.162720	0.460649
month	0.045935	-0.167526
weekday	-0.129386	0.015921
genre_id	-0.057556	0.057694



Incidencia de las Características



Mapeo de Características



Parámetros Redes de Kohonen (SOM)

Inicialización de pesos

- Entre -1 y 1
- Con datos del conjunto

Tipo de entrenamiento

- Random shuffle
- Estocástico

Tipo de grilla

- Hexagonal (4x4)

Radio de vecindad

- Inversamente proporcional a la iteración
- Exponencialmente decae con la iteración

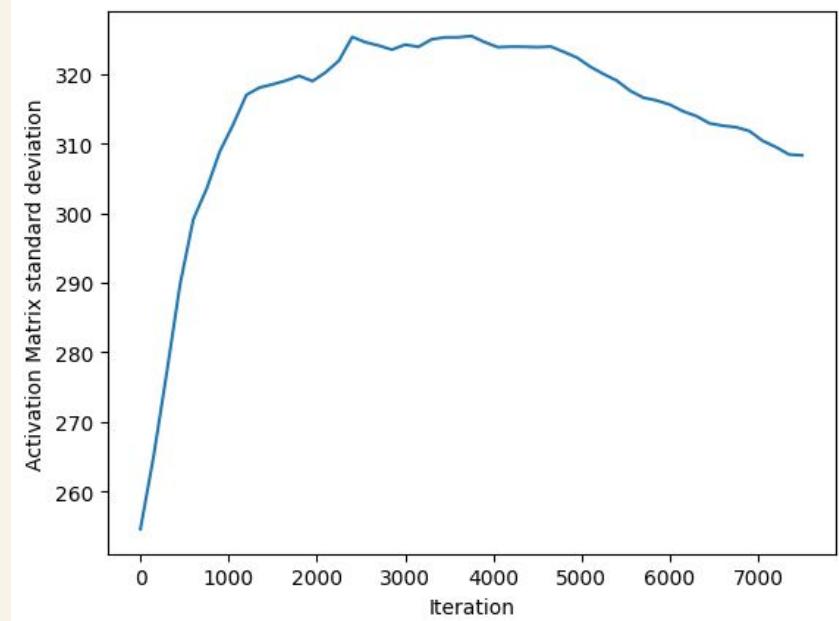
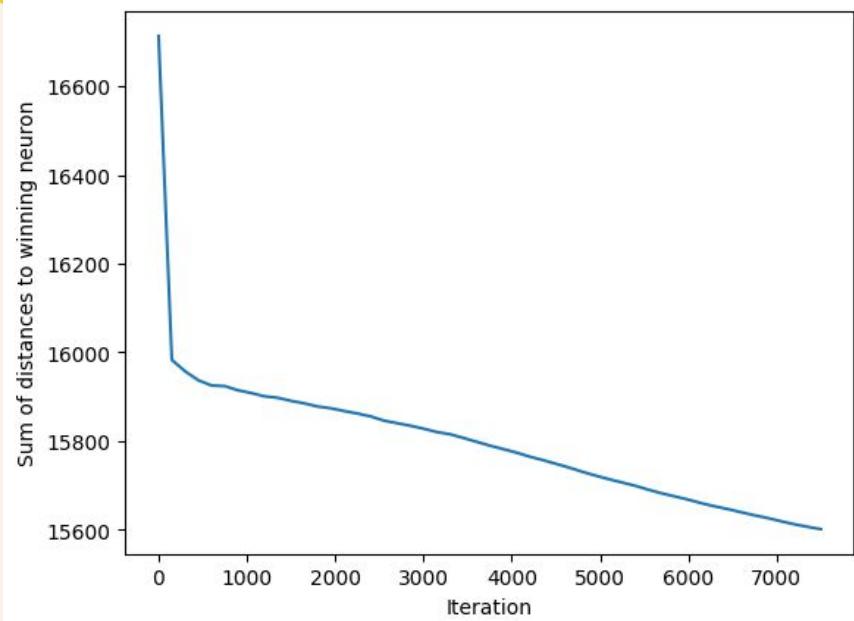
Tasa de aprendizaje

- Fija (0.1)
- Inversamente proporcional a la iteración

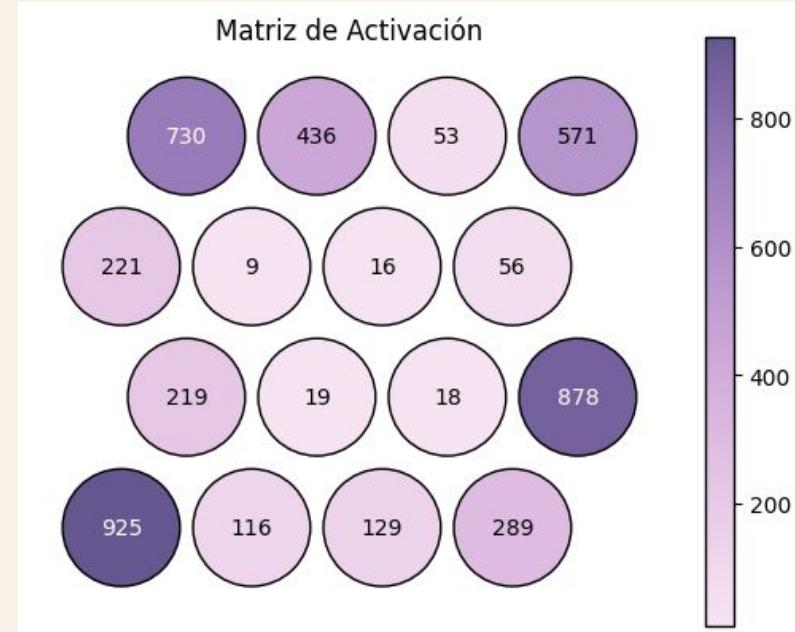
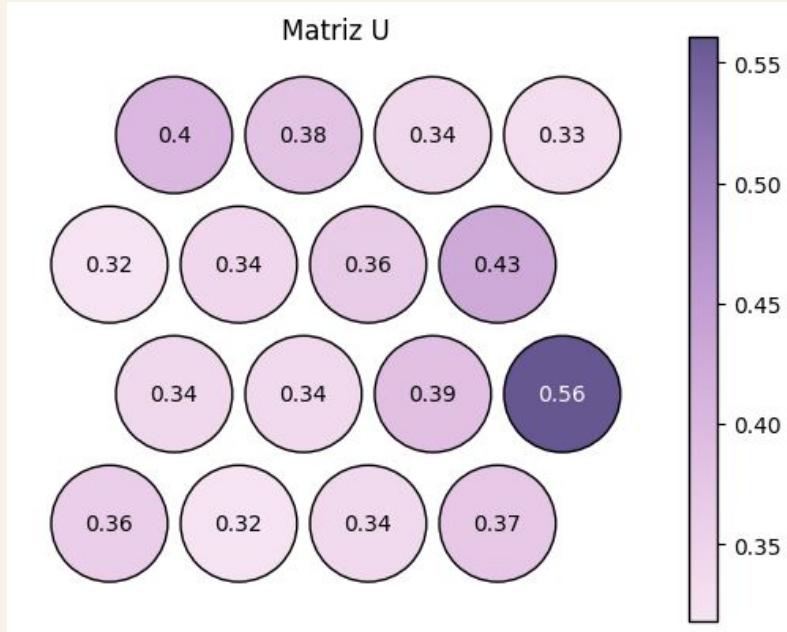
Iters = 15*500

D(min) = 14.633
D(max) = 14.981
D(rango) = 347

Métricas de agrupamiento



Matriz U y Matriz de Activación



03

Predictión de Géneros

Mediante Métodos No Supervisados





Análisis por Género

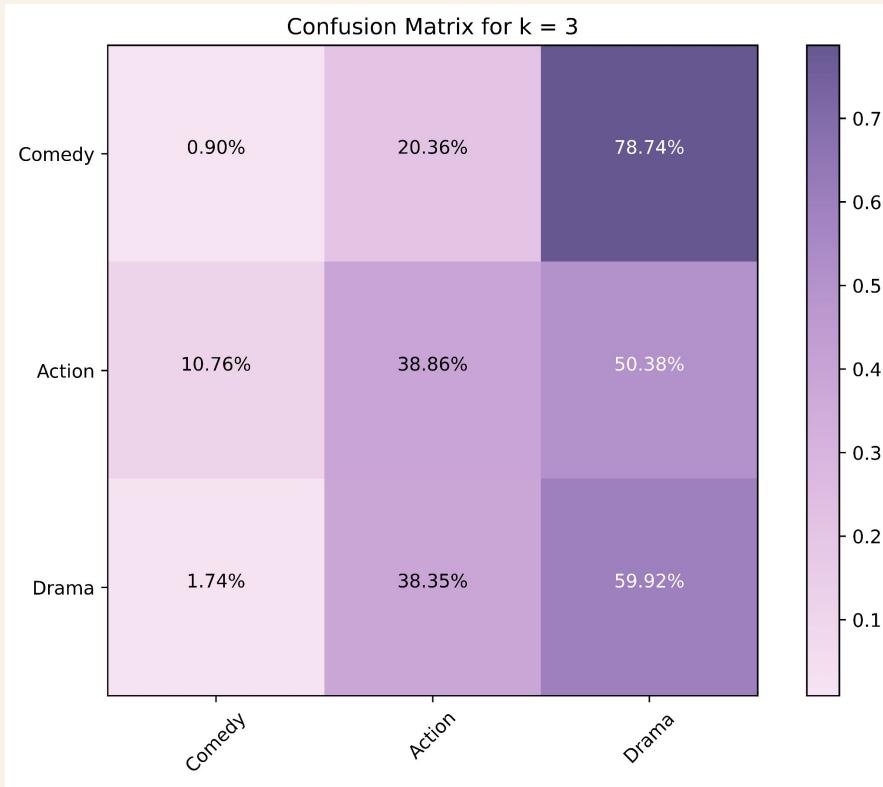
“Action”, “Comedy” y “Drama”

- Se realiza un filtrado del dataset,
- Nos quedamos con los valores de género correspondientes a “Action”, “Comedy” y “Drama”
- En total, **2926** registros.
 - **Drama: 1146**
 - **Acción: 835**
 - **Comedia: 945**





K Means

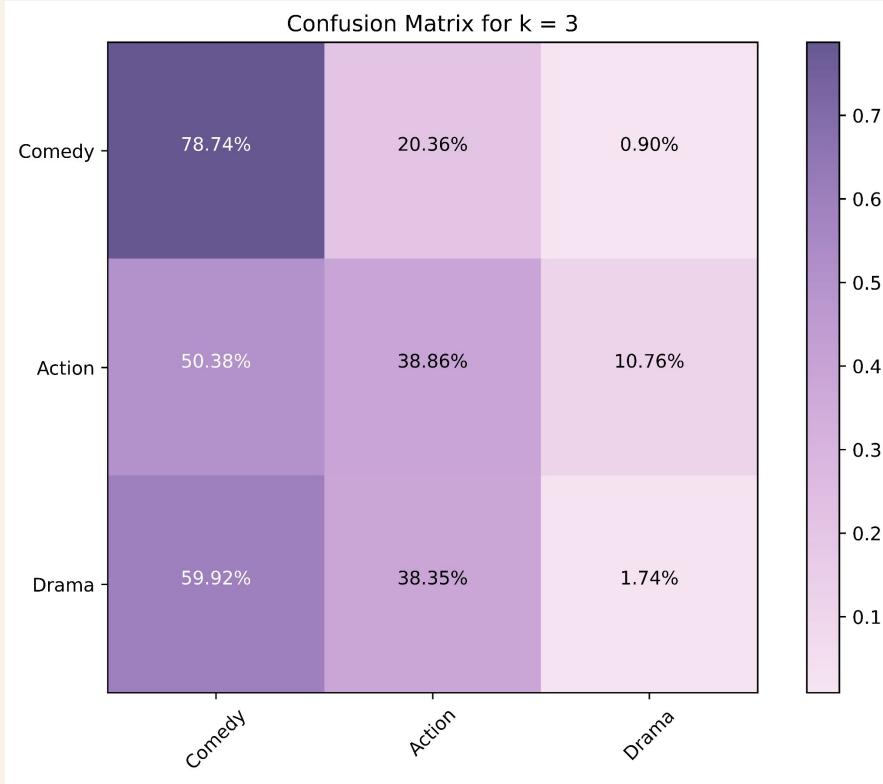


- **Precision:** 0.27,
Error: 0.73
- **Accuracy:** 0.35,
Error: 0.65





K Means

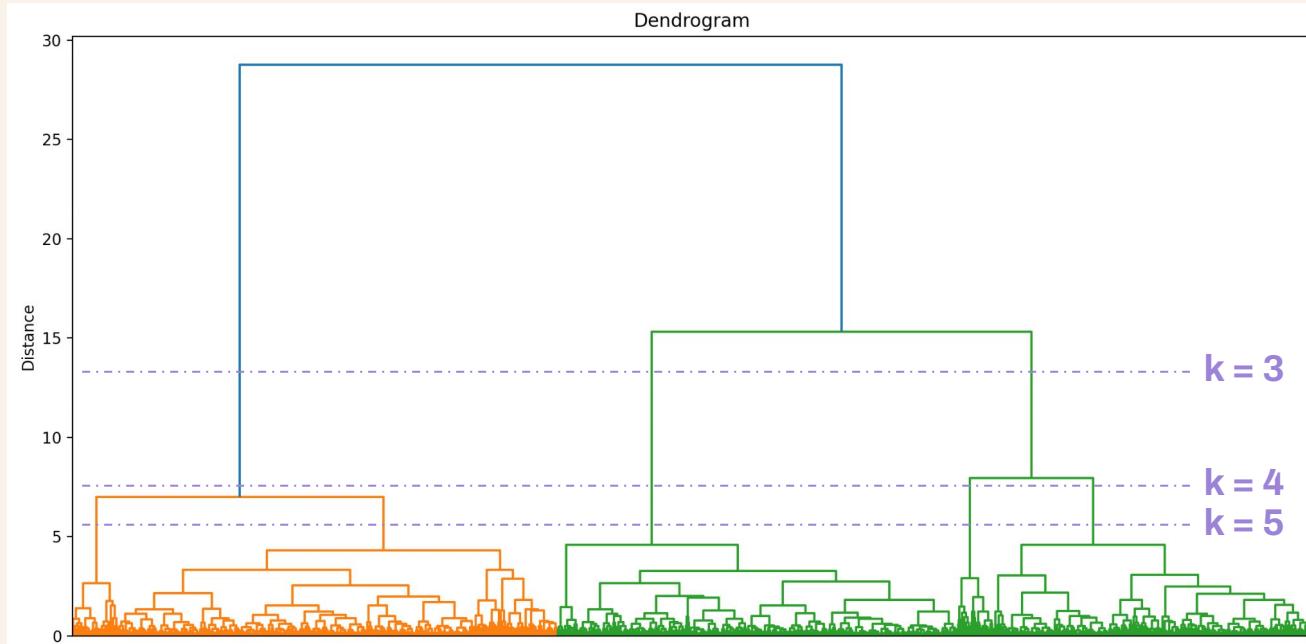


- **Precision:** 0.29,
Error: 0.71
- **Accuracy:** 0.37,
Error: 0.62



Agrupamiento Jerárquico

- Utilizamos $k = 3$, $k = 4$ y $k = 5$
- Vemos cómo clasifica en ambos escenarios
 - “emulamos” un aprendizaje supervisado → predecimos el dataset completo





Agrupamiento Jerárquico

Analizamos la probabilidad de que los miembros de un cluster pertenezcan a una categoría.

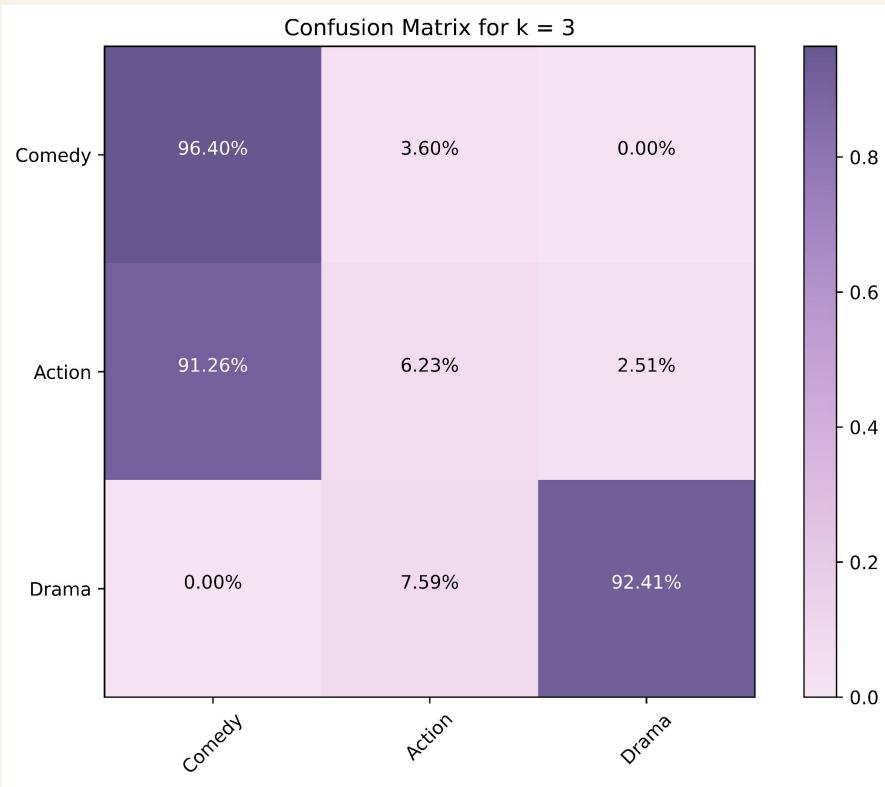
k = 3

	Acción	Comedia	Drama	Registros Totales
Cluster 1	0,00%	33,11%	66,89%	1193
Cluster 2	47,21%	32,33%	20,46%	1701
Cluster 3	100,00%	0,00%	0,00%	32

- Nuestra idea: Establecer el género predominante como género **total** del cluster e intentar clasificar en base a esto
- No hay ningún cluster en donde prevalezca “**Comedia**”, pero dado que el Cluster 2 se acerca, lo tomamos como bucket de este género.
 - Cluster 1: “Drama”,
 - Cluster 2: “Comedia”,
 - Cluster 3: “Acción”



Matriz de Confusión con k = 3



- Gran error en la clasificación de Acción por Comedia, debido al **Cluster 2**
- **Comedia** y **Drama** obtuvieron muy buenos resultados
- **Precision:** 0.65, Error: 0.35
- **Accuracy:** 0.69, Error: 0.31



Agrupamiento Jerárquico

Analizamos la probabilidad de que los miembros de un cluster pertenezcan a una categoría.

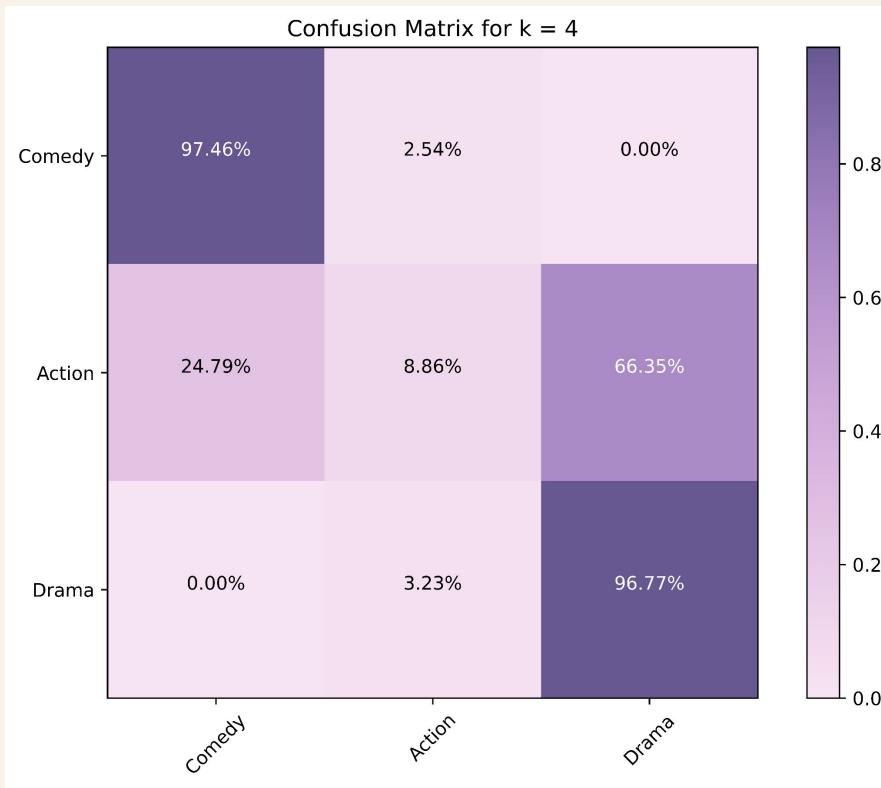
k = 4

	Acción	Comedia	Drama	Registros Totales
Cluster 1	0,00%	33,11%	66,89%	1193
Cluster 2	57,40%	42,52%	0,09%	1176
Cluster 3	24,38%	9,52%	66,10%	525
Cluster 4	100,00%	0,00%	0,00%	32

- Vemos que es posible establecer un género por Clúster
 - Cluster 1: “Drama”
 - Cluster 2: “Comedia” (caemos en el mismo caso que **k = 3**)
 - Cluster 3: “Drama”
 - Cluster 4: “Acción”



Matriz de Confusión con $k = 4$



- Sigue habiendo un error dado que el Cluster 2 debería pertenecer a las películas de **Acción** y no **Comedia**.
- Buena clasificación de **Comedia** y **Drama**
- **Precision:** 0.68,
Error: 0.32
- **Accuracy:** 0.72,
Error: 0.28



Agrupamiento Jerárquico

Analizamos la probabilidad de que los miembros de un cluster pertenezcan a una categoría.

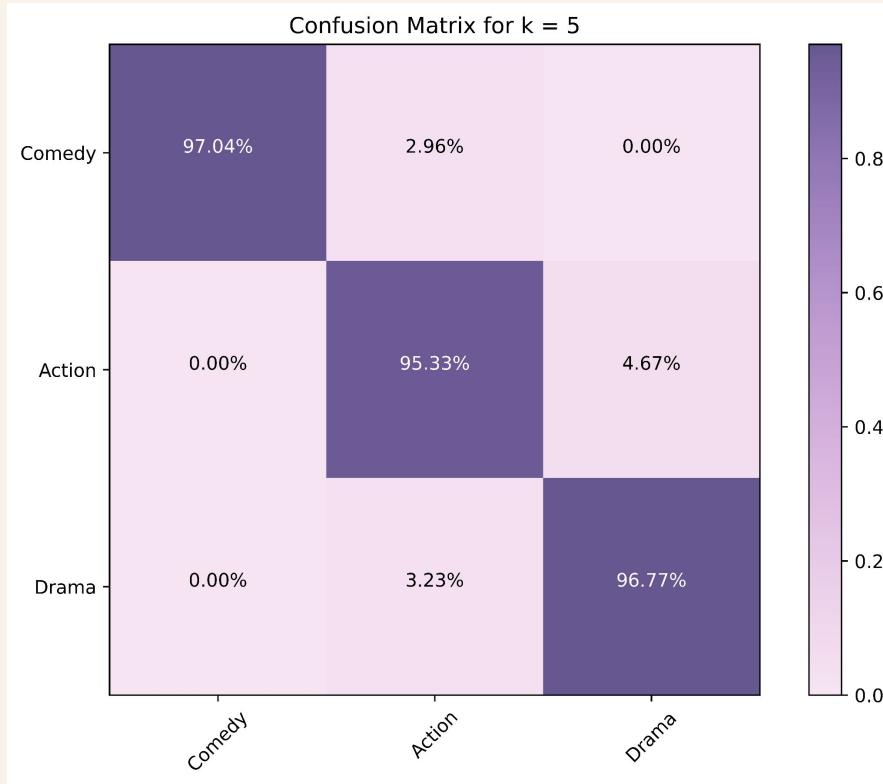
k = 5

	Acción	Comedia	Drama	Registros Totales
Cluster 1	0,00%	0,00%	100,00%	690
Cluster 2	57,40%	42,52%	0,09%	1176
Cluster 3	24,38%	9,52%	66,10%	525
Cluster 4	100,00%	0,00%	0,00%	32
Cluster 5	0,00%	78,53%	21,47%	503

- Vemos que es posible establecer un género por Clúster, sin los problemas anteriores
 - Cluster 1: “Drama”
 - Cluster 2: “Acción”
 - Cluster 3: “Drama”
 - Cluster 4: “Acción”
 - Cluster 5: “Comedia”



Matriz de Confusión con $k = 5$



- Mejora **significativa** de la clasificación
- Agregar un clúster más permite que la agrupación generalice mejor
- No hicieron falta asignar etiquetas incorrectas → **MUCHO** mejor resultado
- **Precision:** 0.97,
Error: 0.03
- **Accuracy:** 0.96,
Error: 0.04



Kohonen

Otorgamos a cada neurona un género según el género que predomine en la misma

k = 10

- Tenemos en la neurona (0, 0)
 - Drama
 - Drama
 - Acción
- Todos los ejemplos de esa neurona serán clasificados como Drama



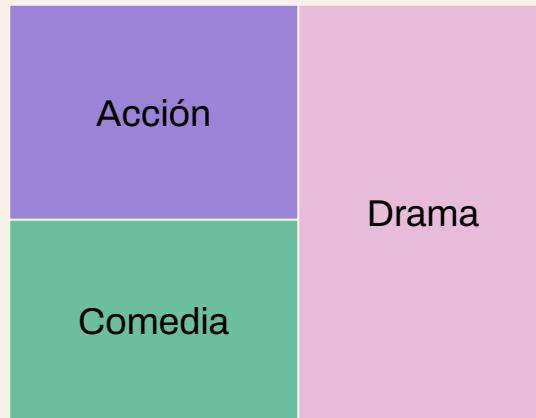


Kohonen

Otorgamos a cada neurona un género según el género que predomine en la misma

k = 10

Usamos una función de inicialización de pesos especial que divide la grilla en 3 cuadrantes y asigna pesos aleatorios de una categoría específica a esas neuronas



Parámetros Redes de Kohonen (SOM)

Inicialización de pesos

- Entre -1 y 1
- Con datos del conjunto

Tipo de entrenamiento

- Random shuffle
- Estocástico

Tipo de grilla

- Rectangular (10x10)

Radio de vecindad

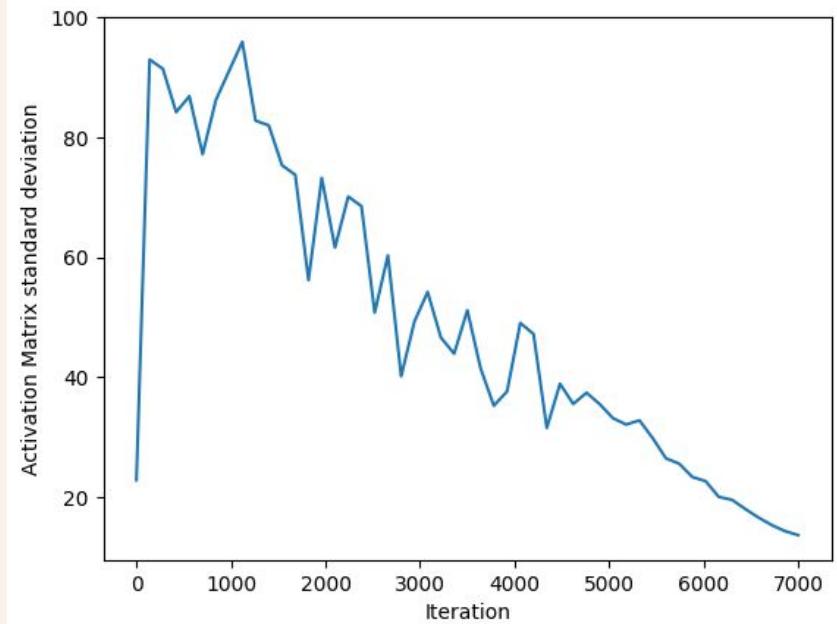
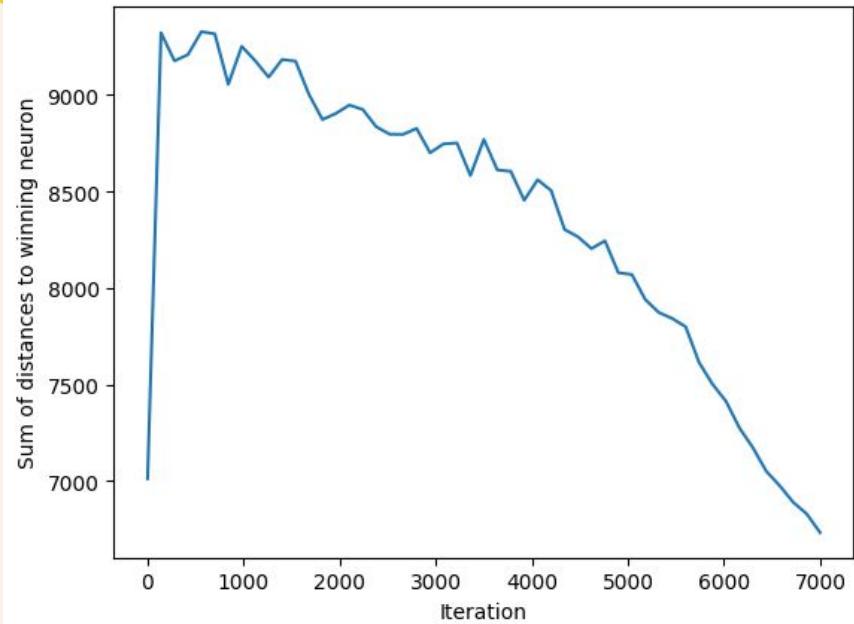
- Inversamente proporcional a la iteración
- Exponencialmente decae con la iteración

Tasa de aprendizaje

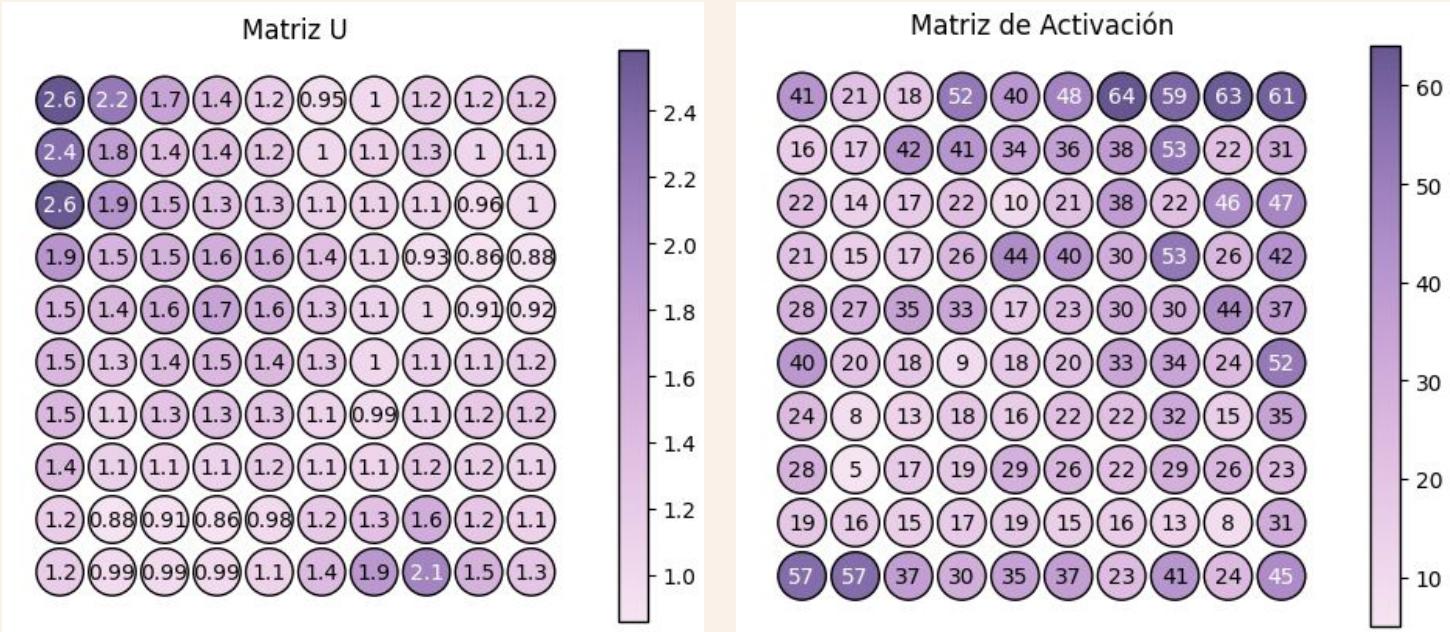
- Fija (0.1)
- Inversamente proporcional a la iteración

Iters = 15*500

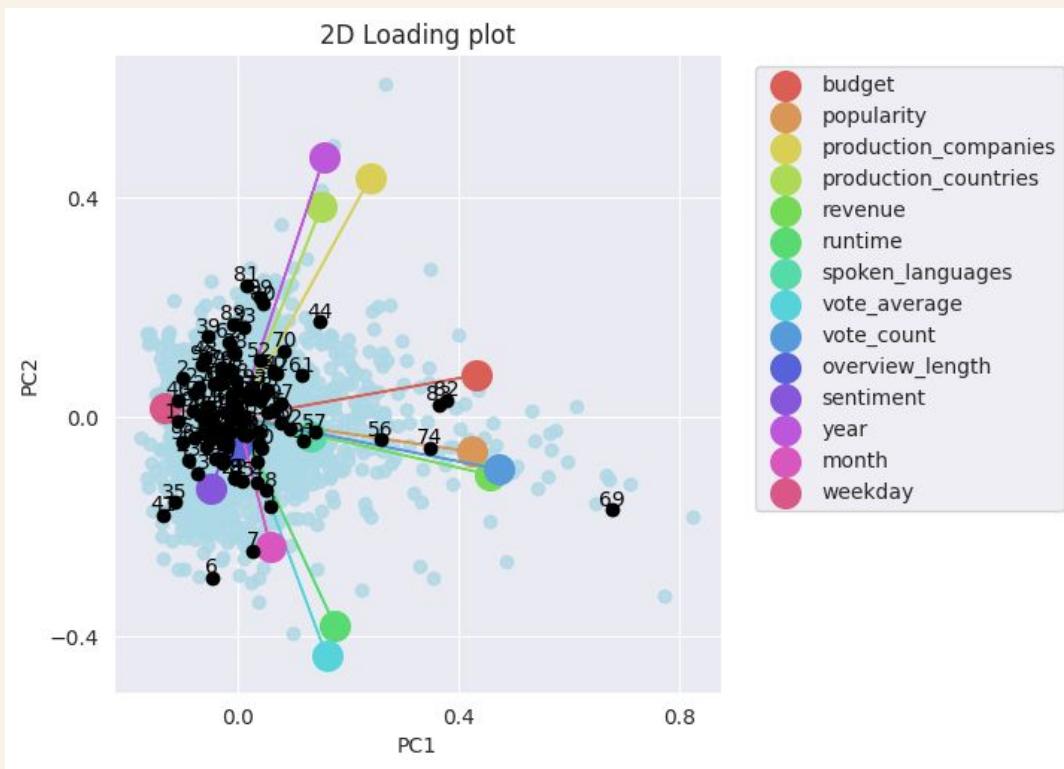
Métricas de agrupamiento



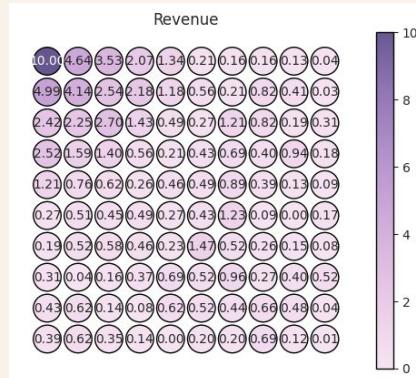
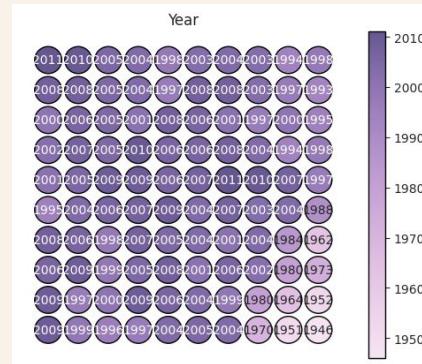
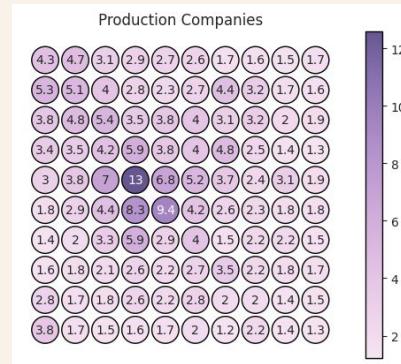
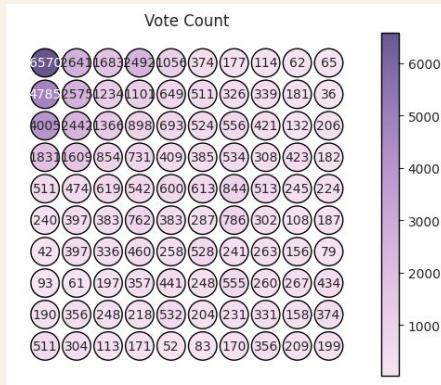
Matriz U y Matriz de Activación



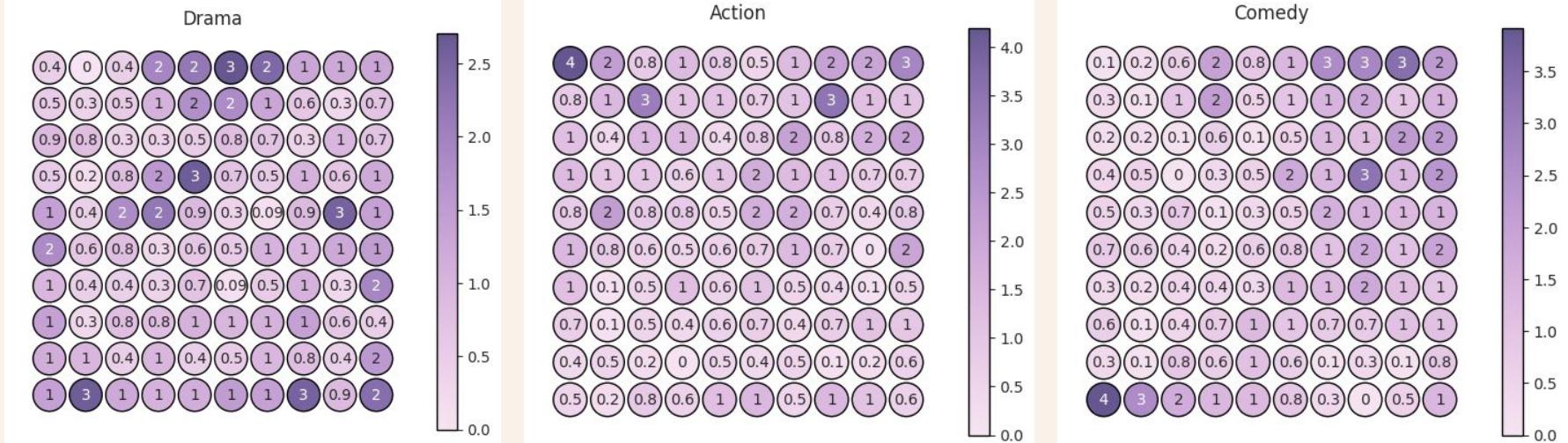
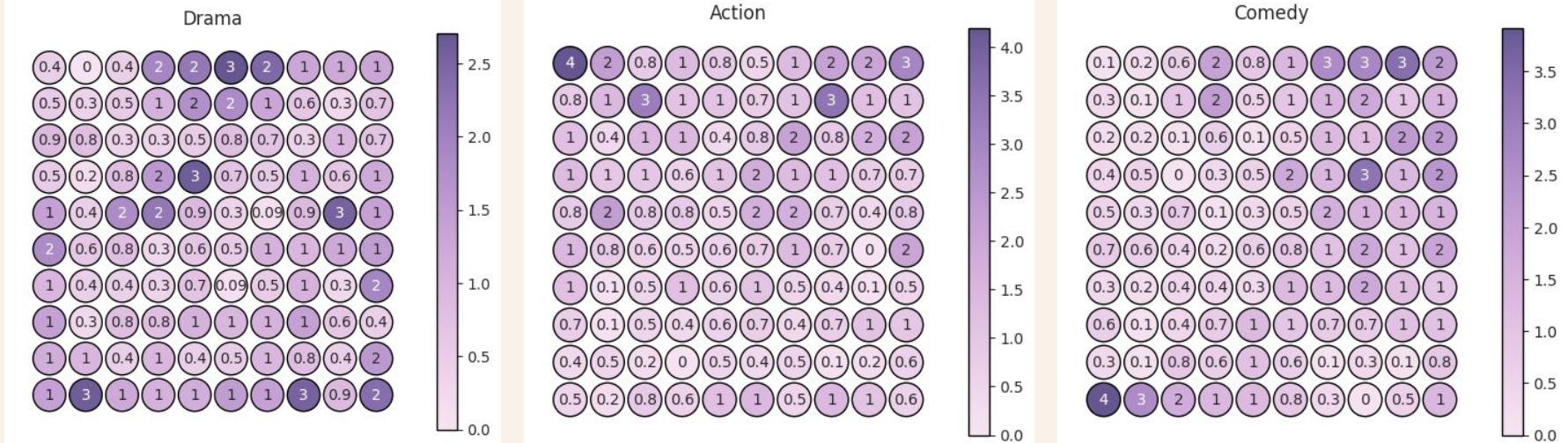
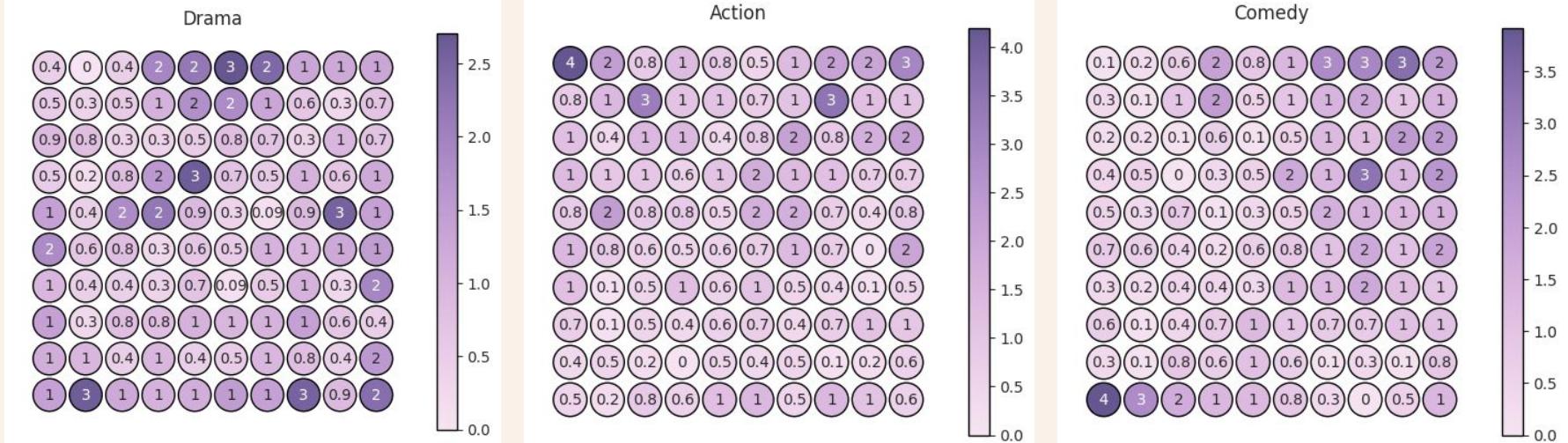
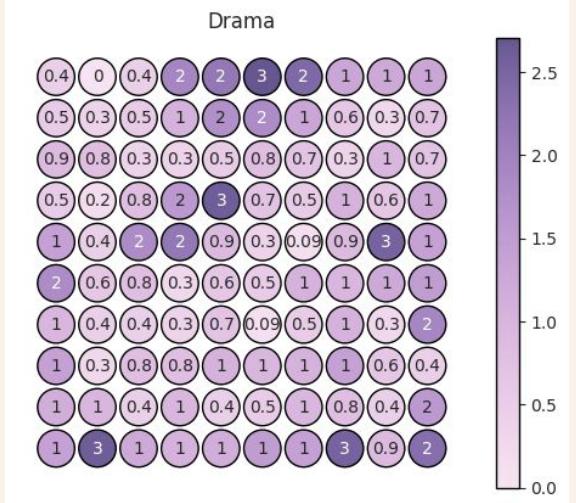
Pesos seleccionados



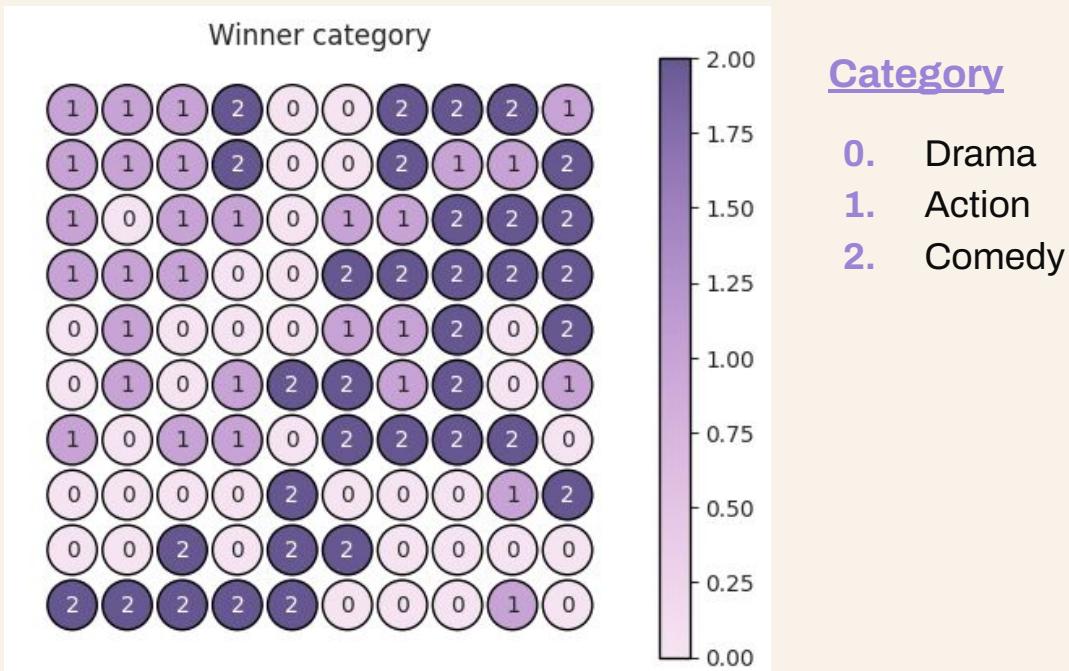
Incidencia de las Características



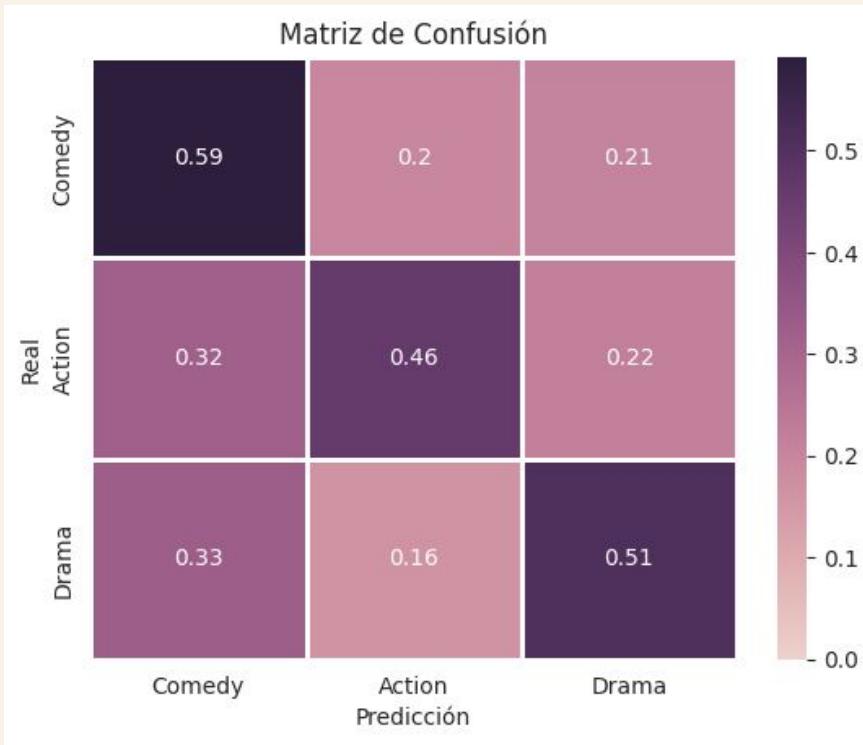
Incidencia del Género



Categorías Ganadoras



Categorías Ganadoras



- **Comedy**
 - **Precision:** 0.47,
 - **F1 Score:** 0.52
 - **Recall:** 0.59,
 - **Accuracy:** 0.65,
- **Action**
 - **Precision:** 0.50,
 - **F1 Score:** 0.48,
 - **Recall:** 0.46,
 - **Accuracy:** 0.72,
- **Drama**
 - **Precision:** 0.60,
 - **F1 Score:** 0.55,
 - **Recall:** 0.51,
 - **Accuracy:** 0.68,

04

Nuestras Conclusiones

Y puntos interesantes a destacar



Conclusiones

- Cada método requiere su proceso particular de pre-procesamiento:
 - Kohonen requiere **normalización**
 - Tanto a **hierarchical clustering** como a **k-means** les basta con **estandarización**
- **Agrupamiento Jerárquico** no es óptimo al momento de tratar con grandes volúmenes de información:
 - Si no hay un criterio de interés para analizar, es muy difícil interpretar un gráfico tan grande
 - A mayor cantidad de datos, más lento el tiempo de cómputo
- En **agrupamiento jerárquico** hay que probar con distintas cantidades de **clusters** finales:
 - no necesariamente una cantidad de clusters igual a la cantidad de etiquetas a clasificar es lo mejor





Conclusiones

- Un biplot permite observar cómo mapea Kohonen los datos a cada neurona
- No se obtienen buenos resultados en Kohonen:
 - Para clasificación
 - Con una tasa de aprendizaje inversamente proporcional de manera directa a la iteración
- Conviene utilizar las características con mayor incidencia en las componentes principales para entrenar la red
- Siempre es una mejor idea inicializar los pesos con datos del **conjunto de entrenamiento**
- Para este conjunto en particular, la **grilla rectangular** dio marginalmente mejores resultados





Conclusiones

- 
- 
- KMeans depende mucho de la inicialización de los centroides
 - La similitud entre **WCSS** y **SSE** es consecuencia de la previa eliminación de outliers (no hay datos muy alejados del centroide)
 - La naturaleza del dataset para géneros parece no ser clusterizable, lo que causa los malos resultados al hacer agrupaciones de 3 en **KMeans**
 - Las películas de acción son las que generan más Revenue, las más Populares y que tienen más votos. Para las comedias es al revés (**Kohonen**)



¡Gracias!

¿Preguntas?

Credits: This presentation template was created by [Slidesgo](#), including icons by [Flaticon](#) and infographics & images by [Freepik](#)