Trabajo Práctico
Trabajo Final Integrador
Integración de contenidos
Bases de Datos Masivas – 11088

Alejandro F. Dunogent
alejandro.dunogent@gmail.com

Índice

1. Propuesta de trabajo final

A partir de un dataset que contiene los datos recolectados de una página de recomendaciones de películas, en donde se encuentran documentadas un millón de calificaciones realizadas, las cuales incluyen datos del usuario (Género, edad, ocupación y código postal), datos de la película (Titulo, año y género) y datos sobre la calificación (Puntaje y fecha), realizar los pasos del proceso de descubrimiento de conocimiento con el objetivo de encontrar patrones validos que aporten conocimiento nuevo.

Dataset usado: http://grouplens.org/datasets/movielens/1m/

2. Problemática abordada

Partiendo de los datos provistos por una página de análisis de películas que contiene un conjunto de calificaciones hechas por los usuarios, se desea poder encontrar patrones validos que aporten nuevo conocimiento.

3. Objetivos

Los objetivos propuestos para este trabajo son:

- Realizar un análisis de datos inicial que incluya:
  - Cuales son los géneros con mayor porcentaje de votos positivos en cada uno de los estados de Estados Unidos.
  - Qué temáticas se votaron positivamente según el genero de la persona.
  - Cuales son los géneros mas votados según la profesión de la persona.
  - Los géneros mas populares para distintos rangos de edad.
- A partir de la cantidad de películas de cada genero que los usuarios del dataset calificaron y del promedio de todas las calificaciones que realizo cada uno, determinar si tienden a votar de manera positiva o negativa y con esa información generar un árbol de decisión que permita determinar que condiciones provocan que un usuario tienda a calificar de una u otra manera.
- Realizar un análisis mas profundo en el cual tomando en cuenta la cantidad de películas de cada genero que los usuarios del dataset calificaron, se los divida en un cierto numero de grupos mediante la técnica de clustering y luego se realice un análisis de las características de cada cluster obtenido (Como se distribuye el genero de las personas, las edades y profesiones, las temáticas mas populares y las regiones de Estados Unidos a las cuales dichos usuarios pertenecen).

4. Metodología

4.1. Selección

Las fuentes de datos con las que se trabajara en este trabajo práctico son:

Un dataset provisto por la página web “movielens.com” que contiene un millón de

calificaciones realizadas por sus usuarios a películas de distintas temáticas.

Un dataset con los estados y ciudades asociados a los códigos postales de Estados Unidos usado para poder incorporar información sobre la ubicación del usuario a cada calificación realizada. La fuente de este dataset es https://www.aggdata.com/node/86

4.2. Preprocesamiento y transformación

La información sobre la cual es necesario aplicar los algoritmos de data mining no se encuentra empaquetada en un único dataset, sino que esta segmentada en varios archivos, los cuales a su vez contienen los datos de manera comprimida para reducir el tamaño de los mismos. Por lo tanto es necesario realizar varias tareas para acomodar la información antes de poder usarla.

Las herramientas usadas en esta etapa son “Spoon” de la suite de herramientas “Pentaho” y “Rstudio”.

Los pasos que se realizaran en esta etapa son:

- Transformar el archivo de usuarios usando la herramienta “Spoon” en uno nuevo que contenga la siguiente estructura: UserID, Genero, Edad, Ocupación, Cod Postal, Ciudad, Estado y Sigla del Estado.
- Transformar el archivo de películas mediante “Spoon” en uno nuevo que contenga la siguiente estructura: Movie_ID, Titulo, Género y Año. Luego cargar el dataset en R con el objetivo de unificar todos los géneros en una matriz binaria que indique con 0 la ausencia del genero en la película y con 1 la presencia de este.
- Usando “Spoon”, transformar el archivo de calificaciones para pasar el timestamp a formato de fecha.
- Utilizando la herramienta “Spoon”, unificar los datos obtenidos de cada una de las transformaciones realizadas previamente. Luego cargar el dataset resultante en R para añadir el tipo de valoración, se toma como calificación positiva a las valoraciones superiores a 3 estrellas.

4.3. Análisis de Datos

4.3.1. Objetivo: Determinar los géneros de películas mas populares por genero

Para resolver este objetivo se realizaran dos análisis usando R, uno para las personas de genero masculino y otro para las personas de genero femenino.

Para cada genero la metodología a usar sera: Calcular la cantidad de votos positivos para cada temática de las películas, determinar los porcentajes de popularidad de cada una de ellas y determinar cuales son las mas populares, añadiendo un gráfico que muestre los distintos porcentajes.

4.3.2. Objetivo: Determinar los géneros de películas mas populares por estado

En este objetivo, los pasos a realizar usando R son: Dividir el dataset en varios subgrupos, uno por cada estado. Luego por cada uno de ellos, calcular la cantidad de votos positivos de cada genero y sus correspondientes porcentajes. Una vez hecho esto, determinar cual es el genero mas popular en cada estado y mostrarlo mediante un mapa de Estados Unidos que muestre con distintos colores los géneros mas populares.

4.3.3. Objetivo: Determinar los géneros de películas mas populares por profesión

En este objetivo también se hará uso de R, los pasos a seguir son: Para cada una de las profesiones, contar la cantidad de valoraciones positivas que tiene cada genero presente en el dataset. Luego de tener todos los géneros de todas las profesiones calculados, determinar cual es el genero mas popular para cada una de ellas.

4.3.4. Objetivo: Determinar los géneros de películas mas populares por edad

Este objetivo es muy similar al anterior pero para el caso de las edades. Para cada uno de los rangos de edades del dataset, se contara usando R la cantidad de valoraciones positivas que tiene cada genero y luego se determinara cual es el genero mas popular para cada caso.

4.3.5. Objetivo: Predecir la tendencia a calificar de manera positiva o negativa de los usuarios

Para cumplir este objetivo se pasaran los datos a una base de datos MySQL (Usando la herramienta "Spoon") en donde se agrupara por cada usuario la cantidad de películas que vio por genero y su tendencia de voto.

El criterio que se usara para determinar si un usuario tiende a votar de manera positiva es cuando el promedio de sus calificaciones resulta ser mayor o igual a 4.

Luego de realizar el agrupamiento, se pasaran los datos de vuelta a un archivo .csv usando "Spoon".

Una vez hecho esto, usando R se separara el dataset en dos grupos, un grupo con el 80% de los datos para usarlos en el entrenamiento del árbol y otro con el 20% para usarlos posteriormente como observaciones de testeo del árbol.

Para generar los arboles se usara Weka con el algoritmo de clasificación J48.

Se realizaran dos análisis:

- Primero usando validación cruzada se buscara cual es el mejor modelo que se puede generar.
- Luego se usaran los parámetros obtenidos para entrenar el árbol con el 80% de los datos y testearlo con el 20% separado previamente.

También se realizaran los pasos anteriores usando el algoritmo de RandomForest en Weka.

4.3.6. Objetivo: Dividir a los usuarios en clusters según la cantidad de películas calificadas de cada genero y analizar cada uno de ellos

Para este objetivo vamos a considerar dos subconjuntos de datasets posibles, uno que para cada usuario tenga la cantidad de calificaciones que realizo para cada genero y otro que ademas tenga la cantidad de calificaciones positivas y negativas realizadas por genero. La idea de esta división es ver si el agrupamiento realizado mediante clustering difiere al tener esos datos adicionales.

Como resulta demasiado costoso realizar el agrupamiento de cantidades de calificaciones por temática y por usuario utilizando R, mediante "Spoon" se pasara la información a una base de datos de MySQL y se generara allí el agrupamiento mediante consultas SQL. Luego se pasaran los datos de vuelta a un archivo .csv para continuar su análisis.

Una vez obtenidos los archivos, usando R se realizara el clustering para distintas cantidades de grupos (Entre 2 y 8 agrupaciones) y se calculara el coeficiente de silueta para cada uno de ellos para determinar cual es la cantidad optima de grupos.

Finalmente, para cada uno de los grupos obtenidos, se realizaran los siguientes análisis en R:

- Determinar cual es el porcentaje de usuarios de genero femenino y masculino
- Determinar cual es el porcentaje de usuarios de cada rango de edades del dataset
- Determinar el porcentaje de popularidad de cada profesión
- Determinar el porcentaje de usuarios que hay en cada estado
- Determinar el porcentaje de usuarios que hay en cada región
- Determinar el porcentaje de popularidad de cada uno de los géneros de las películas
- Determinar cuales son los géneros mas populares por estado y región

5. Resultados

5.1. Preprocesamiento y transformación

5.1.1. Transformando el archivo de usuarios

El archivo “Users.dat” se encuentra estructurado de la siguiente manera:

```
UserID::Genero::Edad::Ocupación::Codigo-Postal
```

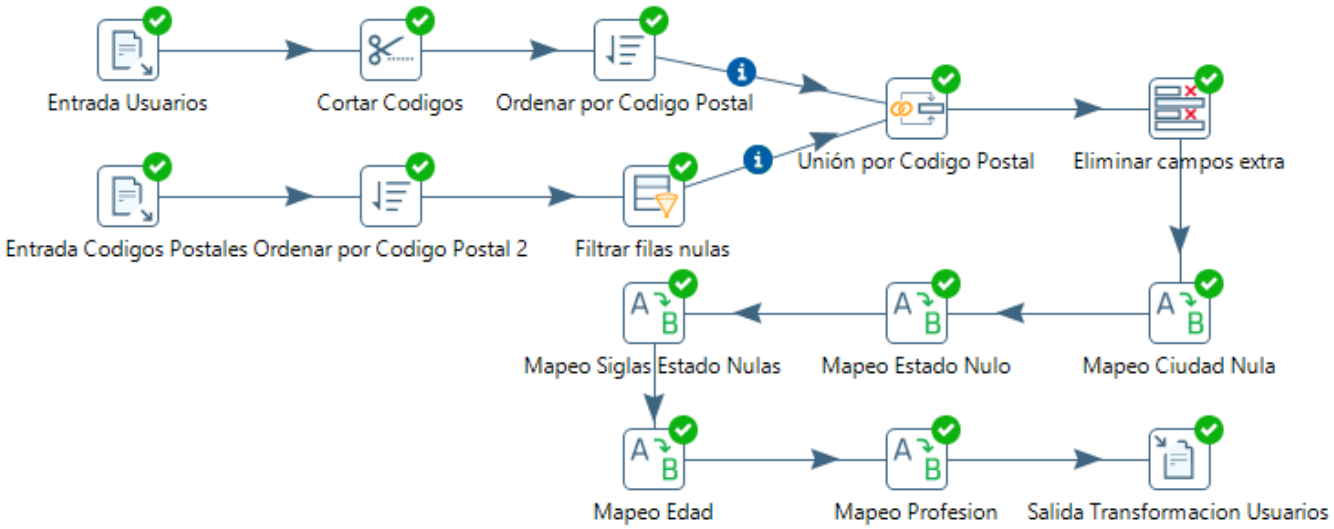
- El genero esta denotado con una F para femenino y M para masculino.
- El valor edad indica el rango de edades al que pertenece la persona.
  - 1: "Menor de 18"

    - 18: "18 a 24"
    - 25: "25 a 34"
    - 35: "35 a 44"
    - 45: "45 a 49"
    - 50: "50 a 55"
    - 56: "56 o mas"
- El valor de ocupación indica la profesión de la persona.
    - 0: "otro" o no especificado
    - 1: "academic/educator"
    - 2: "artist"
    - 3: "clerical/admin"
    - 4: "college/grad student"
    - 5: "customer service"
    - 6: "doctor/health care"
    - 7: "executive/managerial"
    - 8: "farmer"
    - 9: "homemaker"
    - 10: "K-12 student"
    - 11: "lawyer"
    - 12: "programmer"
    - 13: "retired"
    - 14: "sales/marketing"
    - 15: "scientist"
    - 16: "self-employed"
    - 17: "technician/engineer"
    - 18: "tradesman/craftsman"
    - 19: "unemployed"
    - 20: "writer"
- El código postal esta disponible directamente.

También se usa el archivo “us_postal_codes.csv” con las ubicaciones relacionadas a los códigos postales, el cual tiene la siguiente estructura:

```
Cod Postal,Ciudad,Estado,Siglas Estado,Condado,Latitud,Longitud
```

Utilizando “Spoon” se genero el correspondiente archivo csv con la información disponible de manera explicita.



Los pasos realizados son:

- Se lee el archivo “Users.dat”
- Se recortan los códigos postales a los primeros 5 dígitos
- Se lee el archivo “us_postal_codes.csv” (Con las ubicaciones relacionadas a los códigos postales)
- Se ordenan ambos archivos por el código postal
- Se filtran las filas nulas del archivo de códigos postales
- Se realiza la unión de los archivos por el código postal
- Se eliminan los atributos que se consideran innecesarios
- Se mapean las ubicaciones faltantes como “Unknown”
- Se realiza el mapeo de las edades y profesiones a sus valores correspondientes
- Se almacena el archivo .csv resultante

Luego de la transformación, el archivo de usuarios pasa a tener la siguiente estructura:

```
UserID,Genero,Edad,Ocupacion,Cod Postal,Ciudad,Estado,Sigla Estado
```

5.1.2. Transformando el archivo de películas

El archivo “movies.dat” tiene la siguiente estructura:

```
MovieID::Titulo::Géneros
```

- El titulo incluye el año entre paréntesis
- El valor de géneros es una cadena textual con los géneros separados por el carácter “|”
- Los géneros establecidos son:
  - Action
  - Adventure
  - Animation
  - Children's

- Comedy
- Crime
- Documentary
- Drama
- Fantasy
- Film-Noir
- Horror
- Musical
- Mystery
- Romance
- Sci-Fi
- Thriller
- War
- Western

Mediante “Spoon” se genero el archivo .csv con los datos reestructurados

Los pasos son:

- Se carga el fichero .csv con las películas
- Se separa el año del titulo como una columna independiente
- Se divide cada fila en una fila por cada genero de la película
- Se guarda la salida de la transformación

Luego de la transformación, el archivo de películas pasa a tener la siguiente estructura:

`Movie_ID,Titulo,Género,Año`

Imagen de ejemplo

	Movie_ID	Movie_Title	Movie_Genre	Movie_Year
1	1	Toy Story (1995)	Animation	1995
2	1	Toy Story (1995)	Children's	1995
3	1	Toy Story (1995)	Comedy	1995
4	2	Jumanji (1995)	Adventure	1995
5	2	Jumanji (1995)	Children's	1995
6	2	Jumanji (1995)	Fantasy	1995
7	3	Grumpier Old Men (1995)	Comedy	1995
8	3	Grumpier Old Men (1995)	Romance	1995

Luego se cargo el dataset en R para con el objetivo de unificar todos los géneros en

una matriz binaria que indica con 0 la ausencia del genero en la película y con 1 la presencia de este.

Código en Anexo I: 6.1 Transformando el archivo de películas

El archivo de salida pasa a tener la siguiente estructura:

```
"Movie_ID","Titulo","Año","Action","Adventure","Animation","Children's","Comedy","Cr
ime","Documentary","Drama","Fantasy","Film-
Noir","Horror","Musical","Mystery","Romance","Sci-Fi","Thriller","War","Western"
```

Imagen de ejemplo

	Movie_ID	Movie_Title	Movie_Year	Action	Adventure	Animation	Childr
1	1	Toy Story (1995)	1995	0	0	1	
2	2	Jumanji (1995)	1995	0	1	0	
3	3	Grumpier Old Men (1995)	1995	0	0	0	
4	4	Waiting to Exhale (1995)	1995	0	0	0	
5	5	Father of the Bride Part II (1995)	1995	0	0	0	
6	6	Heat (1995)	1995	1	0	0	
7	7	Sabrina (1995)	1995	0	0	0	
8	8	Tom and Huck (1995)	1995	0	1	0	
9	9	Sudden Death (1995)	1995	1	0	0	

5.1.3. Transformando el archivo de calificaciones

El archivo “ratings.dat” tiene la siguiente estructura:

```
UserID::MovieID::Rating::Timestamp
```

- El rating es una calificación entre 1 y 5
- El timestamp esta establecido en segundos

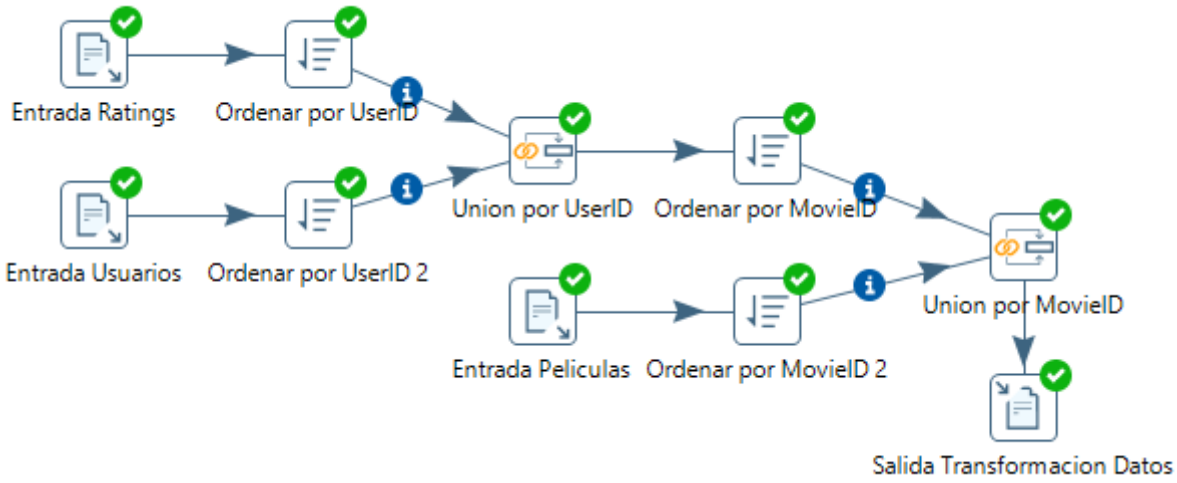
Se realiza la carga del archivo en “Spoon”

Los pasos realizados son:

- Cargar el archivo de calificaciones
- Multiplicar el timestamp por mil para pasarlo a milisegundos
- Cambiar el timestamp a formato fecha de tipo día/mes/año
- Guardar la salida de la transformación

5.1.4. Unificando los datos

Utilizando la herramienta “Spoon” se procede a unificar los datos obtenidos de cada una de las transformaciones realizadas previamente



Los pasos realizados son:

- Se cargan las calificaciones y los usuarios, se ordenan por el ID del usuario y se unen por dicho ID
- Se cargan las películas, se ordenan por el ID de la película y se unen con los datos del paso anterior por dicho ID
- Se guarda la salida de la transformación

Luego de realizar la unión de los datos, procedemos a cargar el dataset resultante en R para añadir el tipo de valoración, se toma como calificación positiva a las valoraciones superiores a 3 estrellas.

Código en Anexo I: 6.2 Unificando los datos

5.2. Análisis de Datos

5.2.1. Objetivo: Determinar los géneros de películas mas populares por genero

5.2.1.1. <u>Calificaciones realizadas por personas de genero masculino</u>

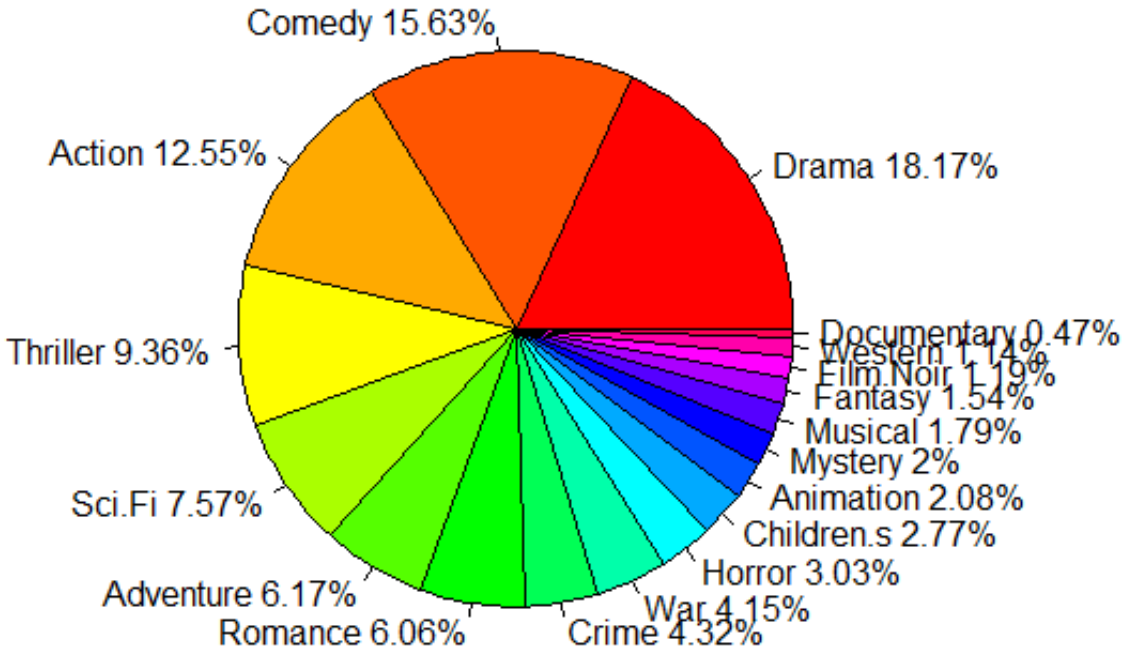
Código en Anexo I: 6.3 Calificaciones realizadas por personas de genero masculino

Salida

	Cantidad de valoraciones positivas
Drama	165059
Comedy	142014
Action	114061
Thriller	85045
Sci.Fi	68754
Adventure	56085
Romance	55047
Crime	39234
War	37746
Horror	27555

Children.s	25172
Animation	18863
Mystery	18142
Musical	16237
Fantasy	14000
Film.Noir	10849
Western	10379
Documentary	4309

Gráfico



Evaluación e interpretación

Los 5 géneros con mayor popularidad entre las personas de genero masculino son el drama, la comedia, la acción el suspenso y la ciencia ficción. El resto de los casos no alcanza a superar el 7% de popularidad. Los tres géneros mas populares abarcan el 46,35% de los casos.

5.2.1.2. Calificaciones realizadas por personas de genero femenino

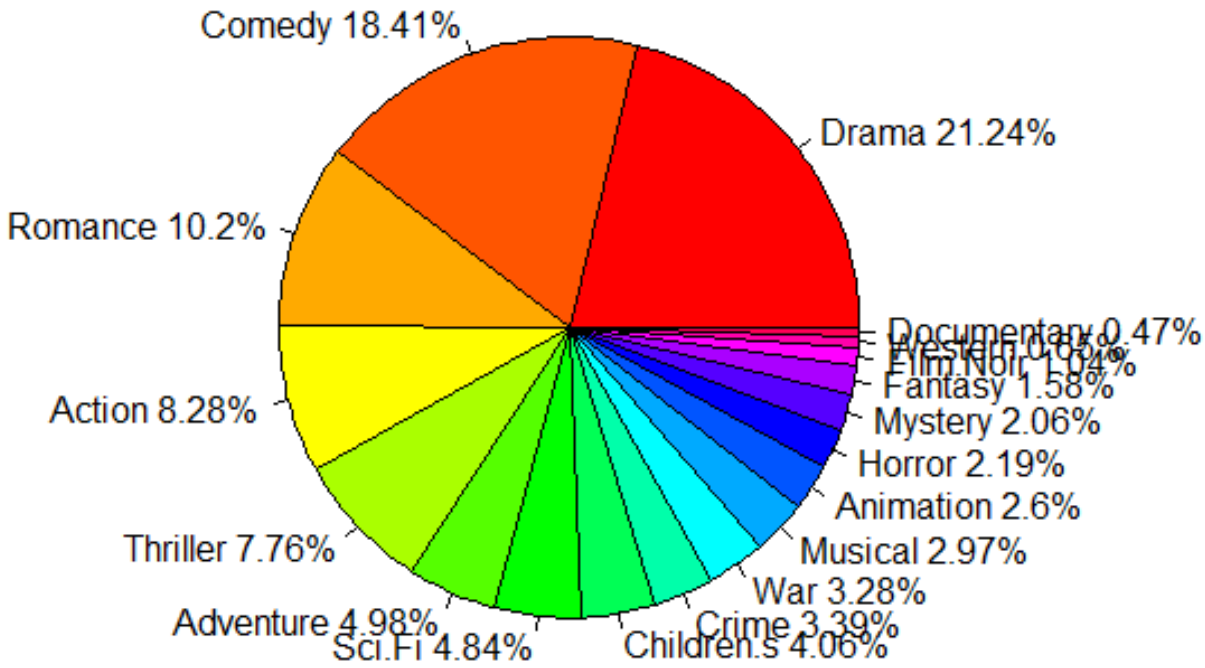
Código en Anexo I: 6.4 Calificaciones realizadas por personas de genero femenino

Salida

	Cantidad de valoraciones positivas
Drama	63381
Comedy	54931
Romance	30452
Action	24705
Thriller	23171
Adventure	14866
Sci.Fi	14443

Children.s	12106
Crime	10113
War	9796
Musical	8874
Animation	7773
Horror	6533
Mystery	6158
Fantasy	4714
Film.Noir	3093
Western	1932
Documentary	1407

Gráfico



Evaluación e interpretación

Los 5 géneros con mayor popularidad entre las personas de genero femenino son el drama, la comedia, el romance, la acción y el suspenso. Como puede verse, el resultado es muy parecido al obtenido para los varones, pero tiene dos diferencias apreciables a simple vista:

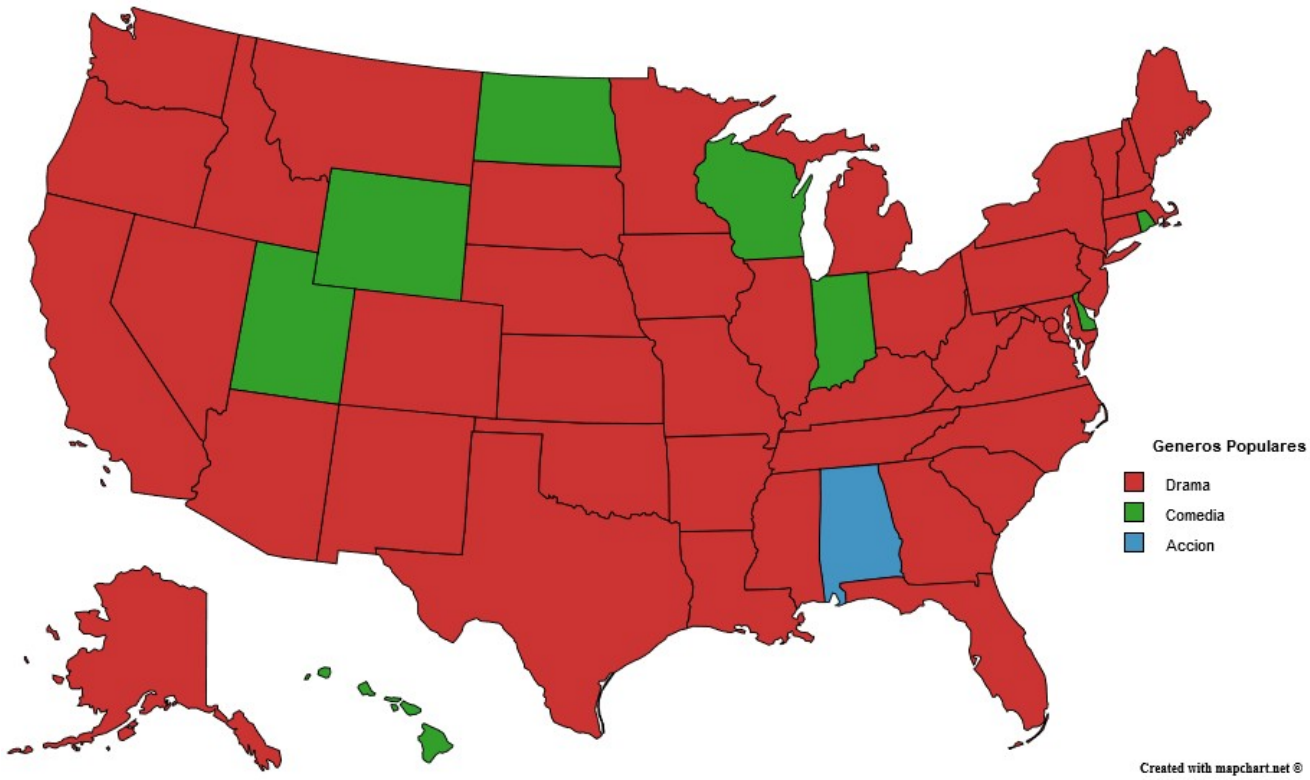
- Los géneros de drama y comedia abarcan porcentajes mucho mayores al resto de los géneros analizados, juntando entre los dos el 39,65% de las valoraciones positivas
- El genero del romance, el cual estaba en el séptimo puesto en el caso de los hombres, escalo hasta el tercero en el caso de las mujeres.

5.2.2. Objetivo: Determinar los géneros de películas mas populares por estado

Código en Anexo I: 6.5 Determinar los géneros de películas mas populares por estado

Salida

ESTADO: GENERO MAS POPULAR
AL: Action
AK: Drama
AZ: Drama
AR: Drama
AE: Action
CA: Drama
CO: Drama
CT: Drama
DE: Comedy
DC: Drama
FL: Drama
GA: Drama
HI: Comedy
ID: Drama
IL: Drama
IN: Comedy
IA: Drama
KS: Drama
KY: Drama
LA: Drama
ME: Drama
MD: Drama
MA: Drama
MI: Drama
MN: Drama
MS: Drama
MO: Drama
MT: Drama
NE: Drama
NV: Drama
NH: Drama
NJ: Drama
NM: Drama
NY: Drama
NC: Drama
ND: Comedy
OH: Drama
OK: Drama
OR: Drama
PA: Drama
PR: Drama
RI: Comedy
SC: Drama
SD: Drama
TN: Drama
TX: Drama
XX: Drama
UT: Comedy
VT: Drama
VA: Drama
WA: Drama
WV: Drama
WI: Comedy
WY: Comedy



Evaluación e interpretación

Podemos ver que en la gran mayoría de los estados la preferencia es por las películas de drama, con la excepción de 8 estados en donde logro superarlo la comedia y un estado, en este caso Alabama, en donde el ganador fue el genero de acción.

Comparándolo con el análisis hecho previamente a los géneros mas populares entre las personas de genero masculino, podemos notar que los tres géneros que se destacan en el mapa son los mismos tres que encabezan el ranking.

5.2.3. Objetivo: Determinar los géneros de películas mas populares por profesión

Código en Anexo I: 6.6 Determinar los géneros de películas mas populares por profesión

Salida

Top 3 Géneros mas populares por profesión

K-12 student
1. Comedy - 18.33%
2. Drama - 13.54%
3. Action - 11.6%

homemaker
1. Comedy - 19.91%
2. Drama - 18.96%
3. Romance - 11.87%

programmer
    1. Drama - 16.63%
    2. Comedy - 15.25%
    3. Action - 12.99%
technician/engineer
    1. Drama - 15.86%
    2. Comedy - 14.93%
    3. Action - 13.86%
academic/educator
    1. Drama - 22.32%
    2. Comedy - 16.56%
    3. Action - 9.42%
clerical/admin
    1. Drama - 19.16%
    2. Comedy - 17.56%
    3. Action - 10.16%
self-employed
    1. Drama - 19.63%
    2. Comedy - 15.55%
    3. Action - 11.62%
other
    1. Drama - 19.06%
    2. Comedy - 16.64%
    3. Action - 11.17%
executive/managerial
    1. Drama - 19.47%
    2. Comedy - 15.17%
    3. Action - 12.73%
college/grad student
    1. Drama - 17.87%
    2. Comedy - 16.96%
    3. Action - 11.93%
writer
    1. Drama - 20.95%
    2. Comedy - 17.18%
    3. Action - 9.13%
retired
    1. Drama - 24.31%
    2. Comedy - 14.53%
    3. Action - 9.34%
scientist
    1. Drama - 18.54%
    2. Comedy - 15.82%
    3. Action - 12.14%
artist
    1. Drama - 20.68%
    2. Comedy - 16.6%
    3. Action - 10.2%
customer service
    1. Comedy - 16.31%
    2. Drama - 15.63%
    3. Action - 13.38%
sales/marketing
    1. Drama - 18.52%
    2. Comedy - 16.53%
    3. Action - 12.24%
doctor/health care
    1. Drama - 21.18%
    2. Comedy - 16.21%
    3. Action - 10.57%

unemployed
1. Drama - 18.18%
2. Comedy - 17.73%
3. Action - 11.11%

lawyer
1. Drama - 20.11%
2. Comedy - 17.24%
3. Action - 10.63%

farmer
1. Comedy - 16.03%
2. Drama - 15.81%
3. Action - 14.11%

tradesman/craftsman
1. Drama - 16.8%
2. Comedy - 15.58%
3. Action – 12.04%

Evaluación e interpretación

En la mayoría de los casos, los tres géneros mas populares para cada profesión fueron el drama, la comedia y la acción, con la acción siempre saliendo tercera y el drama y la comedia intercambiando posiciones según la profesión.

Una cosa interesante de este resultado es la clasificación de la profesión “Homemaker / Ama de casa”, ya que es la única en la cual el genero de acción fue desplazado por el romance, lo cual tiene sentido si consideramos que dicha profesión tiene mayor tendencia a abarcar un mayor porcentaje de personas de genero femenino.

5.2.4. Objetivo: Determinar los géneros de películas mas populares por edad

Código en Anexo I: 6.7 Determinar los géneros de películas mas populares por edad

Salida

Top 3 Géneros mas populares por edad

Under 18
1. Comedy - 18.56%
2. Drama - 14.7%
3. Action – 10.89%

18-24
1. Comedy - 17.29%
2. Drama - 16.95%
3. Action – 12.25%

25-34
1. Drama - 18.71%
2. Comedy - 16.62%
3. Action - 11.91%

35-44
1. Drama - 18.98%
2. Comedy - 15.91%
3. Action – 11.43%

45-49
1. Drama - 20.36%
2. Comedy - 15.58%

3. Action – 10.29%

50-55

1. Drama - 21.47%
2. Comedy - 14.78%
3. Action – 10.53%

56+

1. Drama - 24.28%
2. Comedy - 14.21%
3. Action - 9.41%

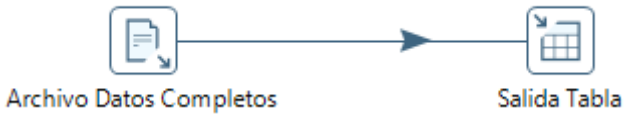
Evaluación e interpretación

Como se vio en los análisis previos, el drama, la comedia y la acción son los géneros de películas mas populares. El resultado muestra que la comedia es la temática mas popular en las personas que tienen hasta 25 años, edad a partir de la cual el drama comienza a ser el mas valorado y cuyo porcentaje es cada vez mayor conforme aumenta la edad.

5.2.5. Objetivo: Predecir la tendencia a calificar de manera positiva o negativa de los usuarios

5.2.5.1. Generando los archivos de entrenamiento y testeo

Partiendo del archivo .csv con los datos completos del dataset, se realiza la copia de estos a una base de datos MySQL utilizando la herramienta “Spoon”.



SQL para crear la tabla destino (Es el mismo código que el usado mas adelante para la transformación del segundo cluster):

Código en Anexo I: 6.9.3.1 Crear la tabla destino

SQL para crear la tabla con la cantidad de películas que vio cada usuario por genero y la tendencia de voto que tiene:

Código en Anexo I: 6.8.1 Crear la tabla con la cantidad de películas que vio cada usuario por genero y la tendencia de voto que tiene

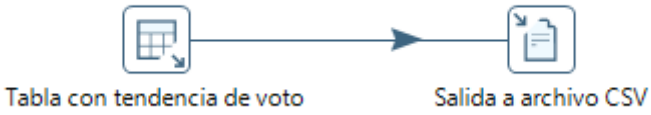
SQL para agrupar por cada usuario la cantidad de películas vistas por genero, añadirle la tendencia de voto y guardar el resultado en la tabla anterior:

Código en Anexo I: 6.8.2 Agrupar por cada usuario la cantidad de películas vistas por genero y añadirle la tendencia de voto

Como se puede ver en el código, el criterio usado para determina si un usuario tiende a

votar de manera positiva es cuando el promedio de sus calificaciones resulta ser mayor o igual a 4.

Transformación de la tabla resultante a .csv



Luego usando R separamos un 20% de los datos para usarlos posteriormente como observaciones de testeo del árbol, de manera que nos quedan dos archivos .csv, el de entrenamiento y el de prueba.

Código en Anexo I: 6.8.3 Separar un 20% de los datos para testeo del árbol

5.2.5.2. Generando los arboles de decisión

Para generar los arboles se uso la herramienta “Weka” con el algoritmo de clasificación J48.

En los siguientes pasos se hará lo siguiente:

- Realizar cross-validation limitando las hojas a 60 elementos en donde se obtiene el mejor balance posible entre: Un tamaño del árbol razonablemente legible y accuracy y área bajo la curva ROC cercanos al valor máximo posible.
- Luego, usando ese mismo parámetro, calcular el árbol sin k-fold usando el 20% separado previamente para hacer el testing.

5.2.5.2.1. Usando validación cruzada con 10 k-folds

```
=== Run information ===

Scheme:weka.classifiers.trees.J48 -C 0.25 -M 60
Relation:     training_extra
Instances:    4832
Attributes:   20
              UserID
              F_Action
              F_Adventure
              F_Animation
              F_Children
              F_Comedy
              F_Crime
              F_Documentary
              F_Drama
              F_Fantasy
              F_FilmNoir
              F_Horror
              F_Musical
              F_Mystery
              F_Romance
```

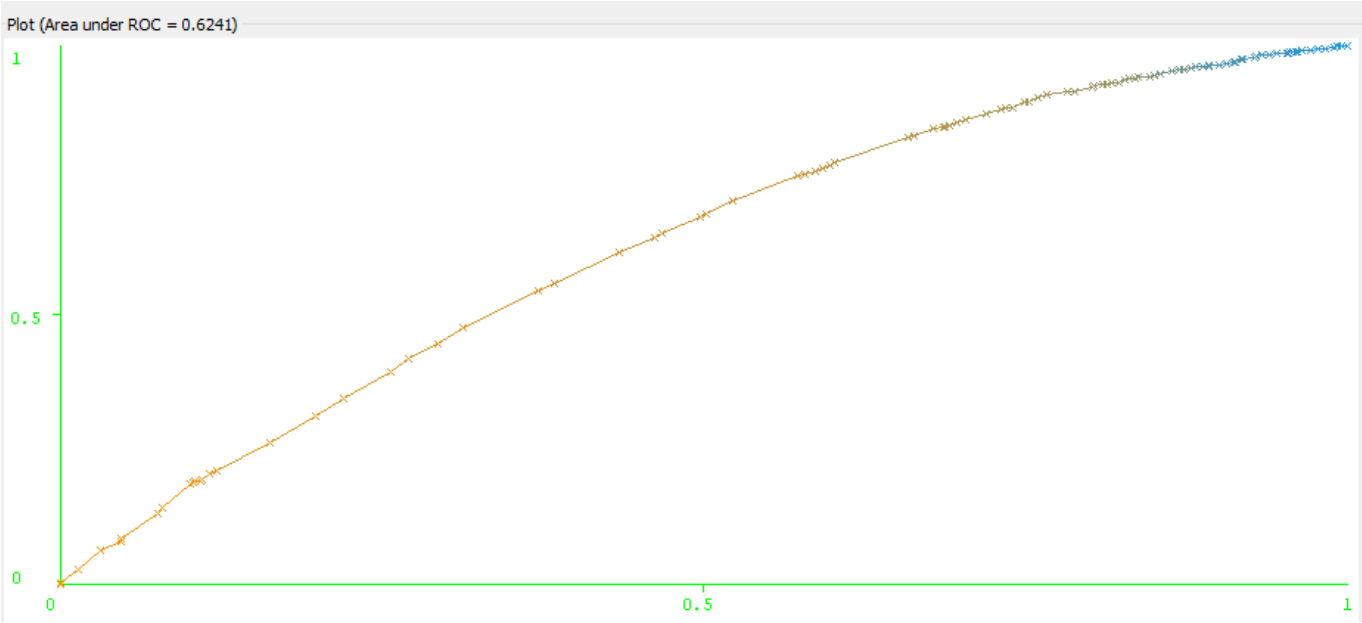
```
              F_SciFi
              F_Thriller
              F_War
              F_Western
              Tendencia_Voto
Test mode:10-fold cross-validation

=== Classifier model (full training set) ===

J48 pruned tree
------------------

F_Horror <= 1
|   F_Comedy <= 42
|   |   F_FilmNoir <= 2
|   |   |   F_War <= 4: N (638.0/214.0)
|   |   |   F_War > 4
|   |   |   |   F_Children <= 3: P (185.0/86.0)
|   |   |   |   F_Children > 3: N (84.0/28.0)
|   |   F_FilmNoir > 2: P (139.0/58.0)
|   F_Comedy > 42: N (131.0/24.0)
F_Horror > 1
|   F_War <= 5: N (1214.0/245.0)
|   F_War > 5
|   |   F_Action <= 14
|   |   |   F_FilmNoir <= 1: N (85.0/32.0)
|   |   |   F_FilmNoir > 1: P (126.0/45.0)
|   |   F_Action > 14: N (2230.0/423.0)

Number of Leaves  :     9

Size of the tree :      17


Time taken to build model: 0.09 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        3600               74.5033 %
Incorrectly Classified Instances      1232               25.4967 %
Kappa statistic                          0.1041
Mean absolute error                      0.3571
Root mean squared error                  0.4271
Relative absolute error                 94.2386 %
Root relative squared error             98.1242 %
Total Number of Instances             4832

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.958     0.879      0.762     0.958     0.849      0.624    N
                 0.121     0.042      0.492     0.121     0.194      0.624    P
Weighted Avg.    0.745     0.667      0.693     0.745     0.682      0.624

=== Confusion Matrix ===

    a    b   <-- classified as
 3452  153 |    a = N
 1079  148 |    b = P
```

El árbol generado tiene un porcentaje de clasificaciones correctas del 74,5% y se puede

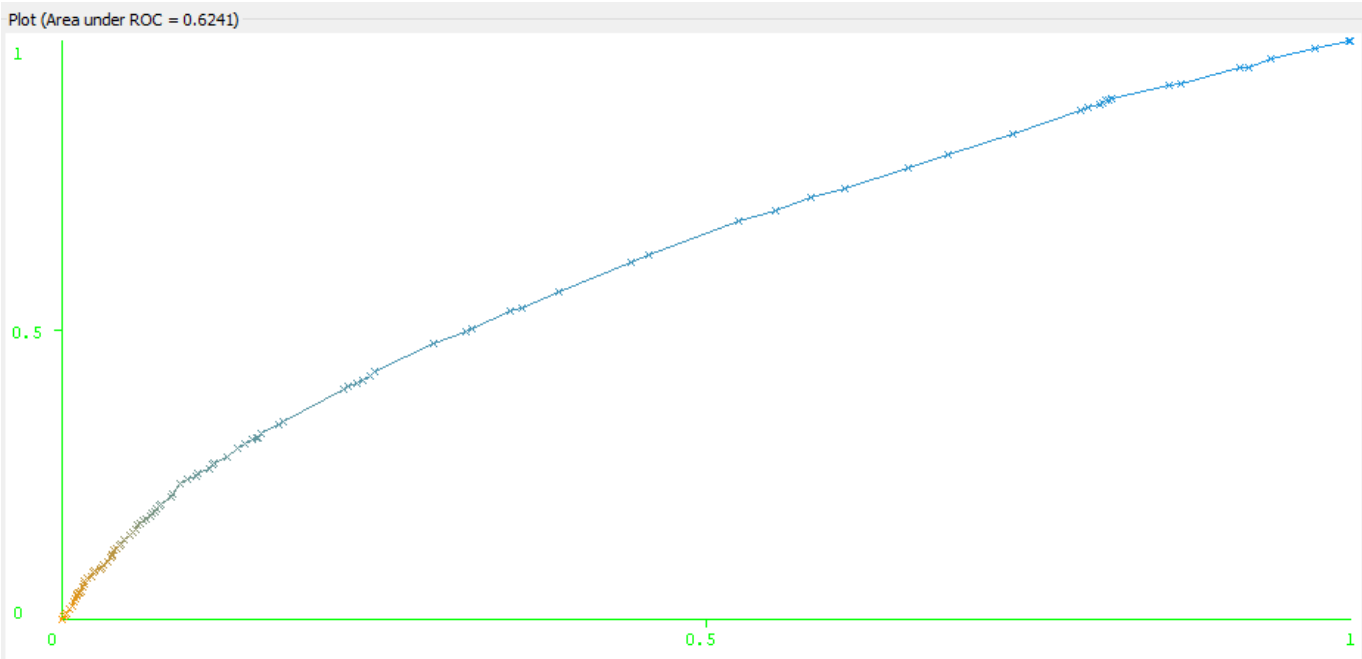
describir de la siguiente manera:

El usuario tendera a calificar las películas de manera positiva si vio mas de una película de horror, mas de 5 de guerra, menos de 15 de acción y mas de una de cine negro. En caso de que no haya visto mas de una de horror, tendera a calificar positivo si vio menos de 43 de comedia y mas de dos de cine negro. Por ultimo, si no vio mas de dos de cine negro, aun puede que tienda a calificar positivamente si vio mas de cuatro de guerra y menos de 4 infantiles.

Curva ROC para Negativos



Curva ROC para Positivos



En el análisis puede observarse que el área bajo la curva resulta ser de 0.6241, lo cual

nos indica que la clasificación no resulto ser demasiado buena, ya que se acerca bastante a la diagonal de 0.5

5.2.5.2.2. Validando el modelo obtenido previamente con los datos reservados

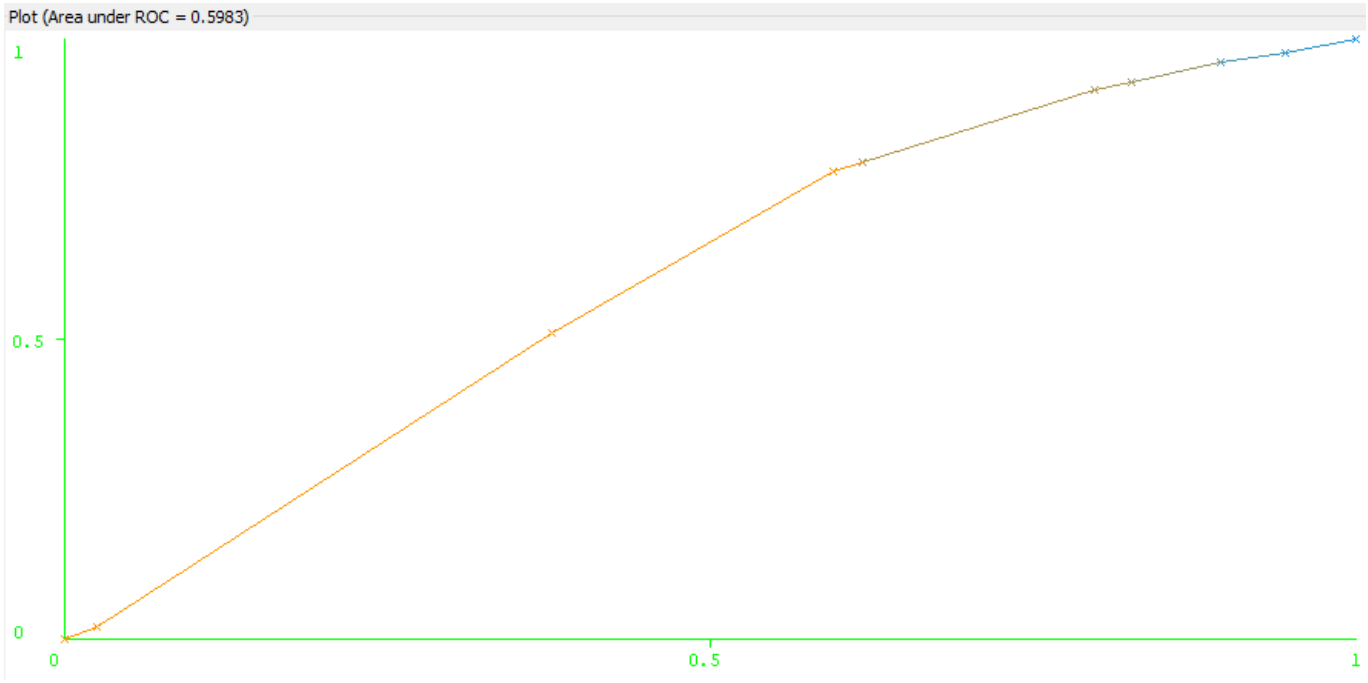
En este paso se realiza la validación del modelo generado previamente con el 20% de los datos que se habían reservado para el testeo.

```
=== Run information ===

Scheme:weka.classifiers.trees.J48 -C 0.25 -M 60
Relation:     training_extra
Instances:    4832
Attributes:   20
              UserID
              F_Action
              F_Adventure
              F_Animation
              F_Children
              F_Comedy
              F_Crime
              F_Documentary
              F_Drama
              F_Fantasy
              F_FilmNoir
              F_Horror
              F_Musical
              F_Mystery
              F_Romance
              F_SciFi
              F_Thriller
              F_War
              F_Western
              Tendencia_Voto
Test mode:user supplied test set:     1208instances

=== Classifier model (full training set) ===

J48 pruned tree
------------------

F_Horror <= 1
|   F_Comedy <= 42
|   |   F_FilmNoir <= 2
|   |   |   F_War <= 4: N (638.0/214.0)
|   |   |   F_War > 4
|   |   |   |   F_Children <= 3: P (185.0/86.0)
|   |   |   |   F_Children > 3: N (84.0/28.0)
|   |   F_FilmNoir > 2: P (139.0/58.0)
|   F_Comedy > 42: N (131.0/24.0)
F_Horror > 1
```

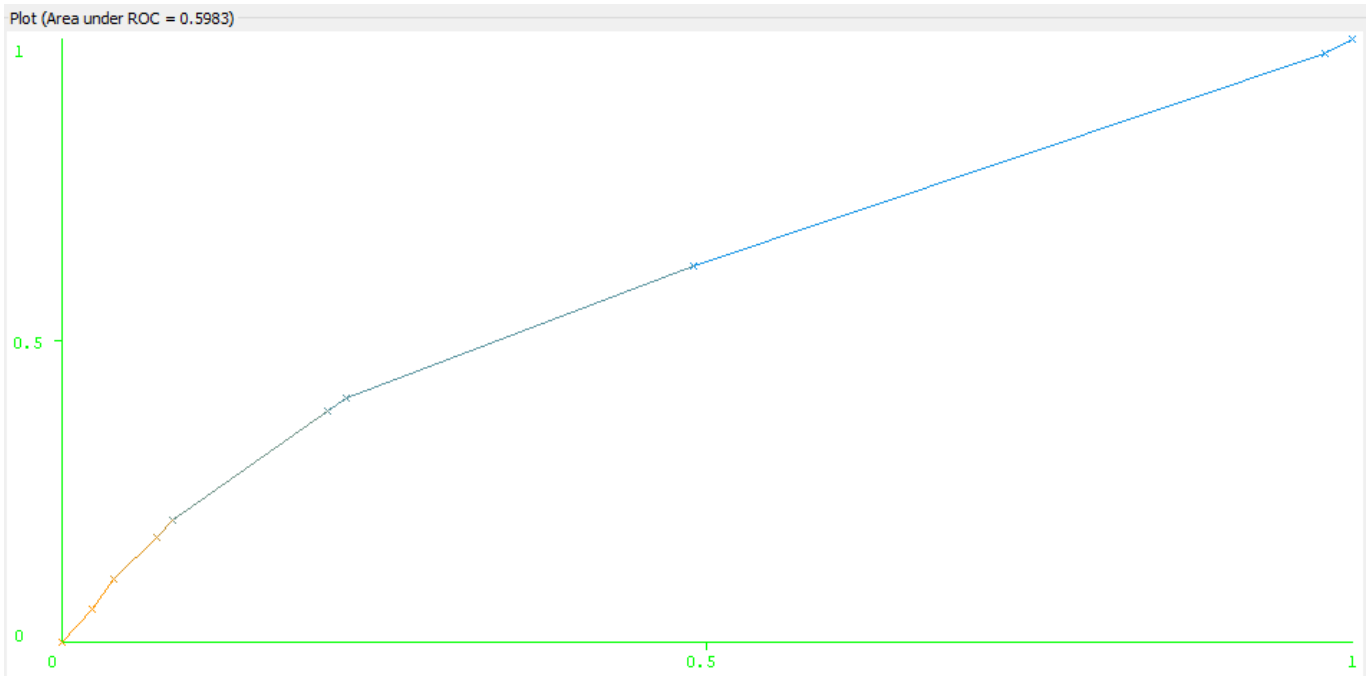
```
|   F_War <= 5: N (1214.0/245.0)
|   F_War > 5
|   |   F_Action <= 14
|   |   |   F_FilmNoir <= 1: N (85.0/32.0)
|   |   |   F_FilmNoir > 1: P (126.0/45.0)
|   |   F_Action > 14: N (2230.0/423.0)

Number of Leaves  :     9

Size of the tree :      17


Time taken to build model: 0.09 seconds

=== Evaluation on test set ===
=== Summary ===

Correctly Classified Instances         880               72.8477 %
Incorrectly Classified Instances       328               27.1523 %
Kappa statistic                          0.1242
Mean absolute error                      0.3647
Root mean squared error                  0.4343
Relative absolute error                 95.1987 %
Root relative squared error             98.6883 %
Total Number of Instances             1208

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.926     0.826      0.759     0.926     0.834      0.598    N
                 0.174     0.074      0.455     0.174     0.251      0.598    P
Weighted Avg.    0.728     0.629      0.679     0.728     0.681      0.598

=== Confusion Matrix ===

   a   b   <-- classified as
 825  66 |   a = N
 262  55 |   b = P
```

En este caso hubo una disminución del porcentaje de instancias clasificadas correctamente, se obtuvo un 72,84% frente al 74,5% del caso anterior.

Curva ROC para Negativos



Curva ROC para Positivos



El área bajo la curva obtenida en este caso también resulto ser peor a la del caso anterior, paso de 0.6241 a 0.598.

5.2.5.3. <u>Usando Random Forest para hacer la clasificación</u>

Para realizar la clasificación se uso la herramienta “Weka” con el algoritmo de clasificación RandomForest.

5.2.5.3.1. Usando validación cruzada con 10 k-folds

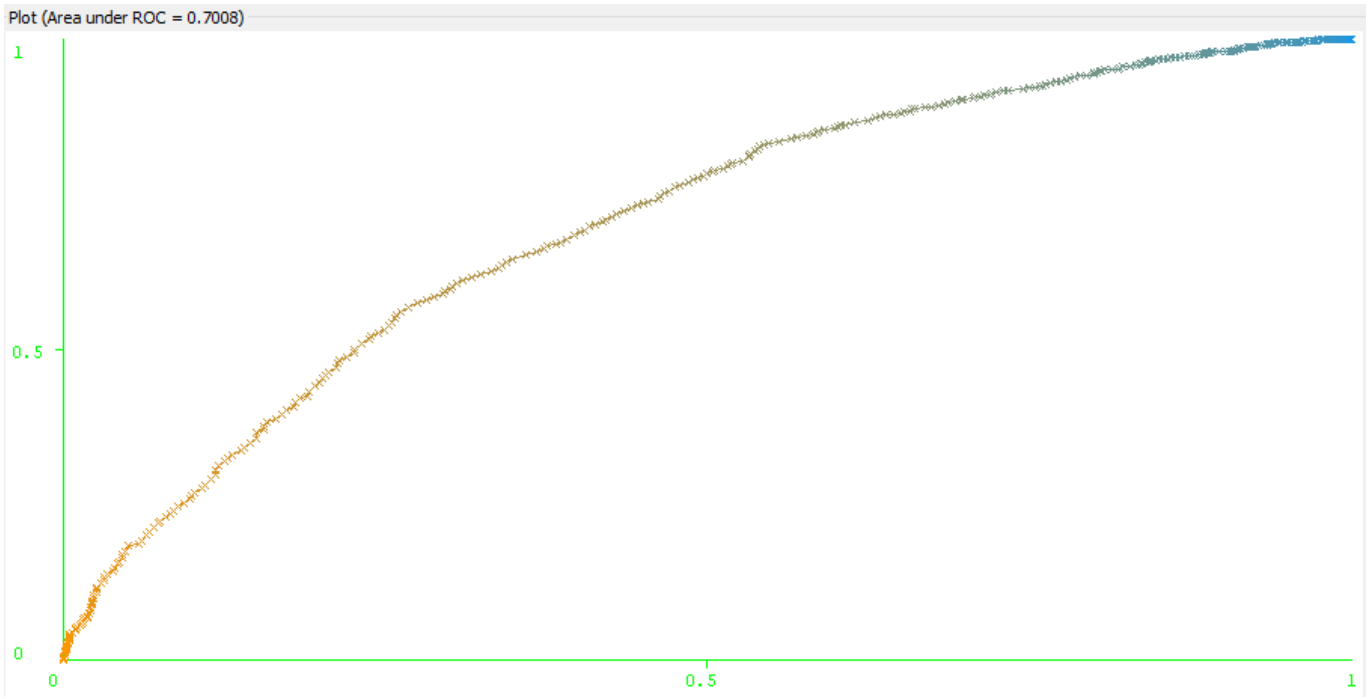
```
=== Run information ===
```

```
Scheme:weka.classifiers.trees.RandomForest -I 500 -K 0 -S 1
Relation:     training_extra
Instances:    4832
Attributes:   20
              UserID
              F_Action
              F_Adventure
              F_Animation
              F_Children
              F_Comedy
              F_Crime
              F_Documentary
              F_Drama
              F_Fantasy
              F_FilmNoir
              F_Horror
              F_Musical
              F_Mystery
              F_Romance
              F_SciFi
              F_Thriller
              F_War
              F_Western
              Tendencia_Voto
Test mode:10-fold cross-validation

=== Classifier model (full training set) ===

Random forest of 500 trees, each constructed while considering 5 random features.
Out of bag error: 0.2432



Time taken to build model: 20.07 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        3657               75.6829 %
Incorrectly Classified Instances      1175               24.3171 %
Kappa statistic                          0.1597
Mean absolute error                      0.3412
Root mean squared error                  0.4125
Relative absolute error                 90.0419 %
Root relative squared error             94.7611 %
Total Number of Instances             4832

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.959     0.837      0.771     0.959     0.855      0.701    N
                 0.163     0.041      0.575     0.163     0.254      0.701    P
Weighted Avg.    0.757     0.635      0.721     0.757     0.702      0.701

=== Confusion Matrix ===

    a    b   <-- classified as
 3457  148 |    a = N
 1027  200 |    b = P
```

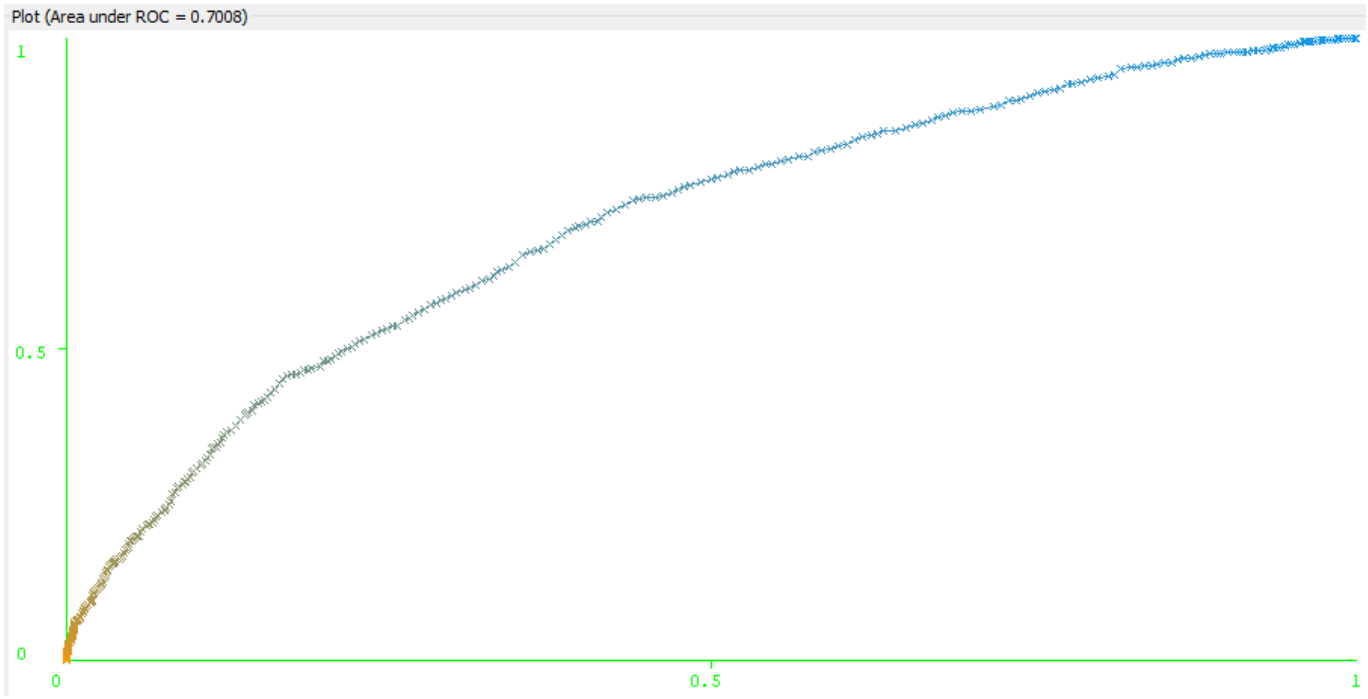
El porcentaje de clasificación correcta fue un poco mejor que en los casos anteriores,

alcanzando un 75,68%.

Curva ROC para Negativos



Curva ROC para Positivos



El área bajo la curva también resulto ser un poco mayor que en los casos anteriores con un valor de 0.701.

5.2.5.3.2. Validando el modelo obtenido previamente con los datos reservados

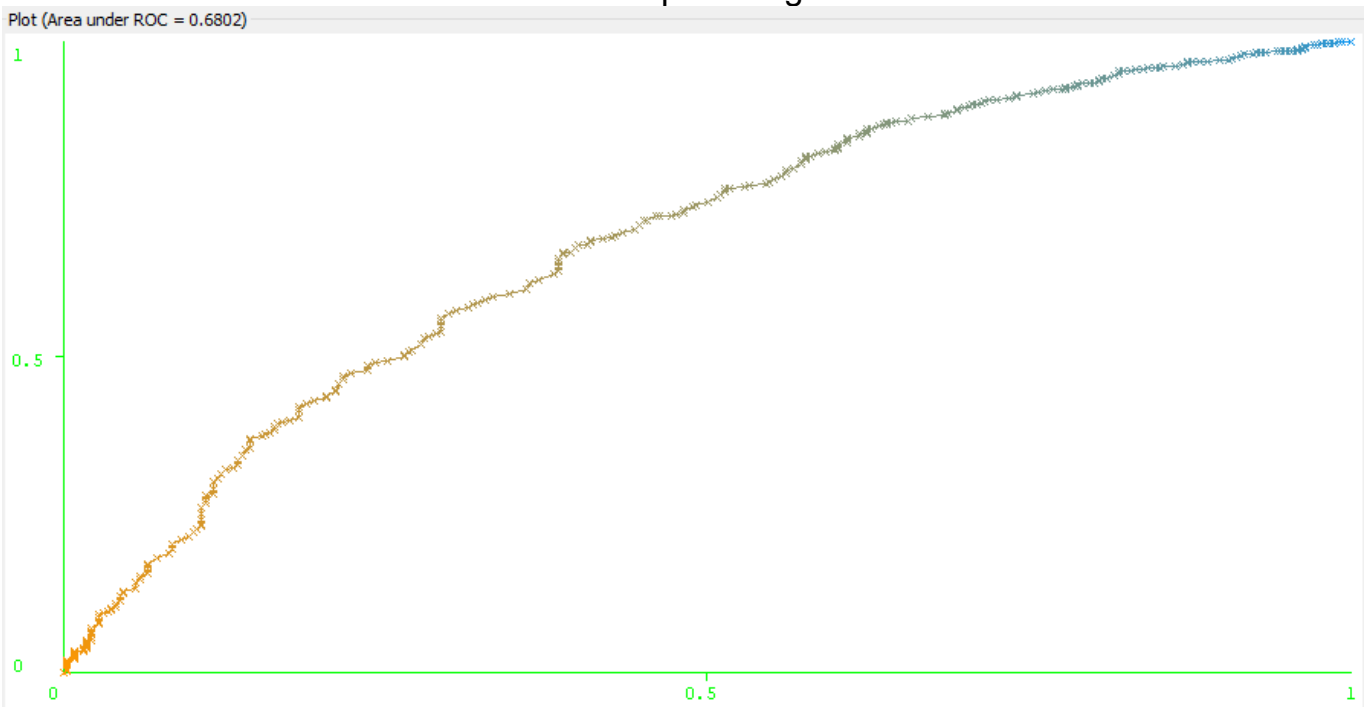
```
=== Run information ===
```

```
Scheme:weka.classifiers.trees.RandomForest -I 500 -K 0 -S 1
Relation:     training_extra
Instances:    4832
Attributes:   20
              UserID
              F_Action
              F_Adventure
              F_Animation
              F_Children
              F_Comedy
              F_Crime
              F_Documentary
              F_Drama
              F_Fantasy
              F_FilmNoir
              F_Horror
              F_Musical
              F_Mystery
              F_Romance
              F_SciFi
              F_Thriller
              F_War
              F_Western
              Tendencia_Voto
Test mode:user supplied test set:     1208instances

=== Classifier model (full training set) ===

Random forest of 500 trees, each constructed while considering 5 random features.
Out of bag error: 0.2432



Time taken to build model: 19.72 seconds

=== Evaluation on test set ===
=== Summary ===

Correctly Classified Instances         902               74.6689 %
Incorrectly Classified Instances       306               25.3311 %
Kappa statistic                          0.1638
Mean absolute error                      0.3525
Root mean squared error                  0.4227
Relative absolute error                 92.0017 %
Root relative squared error             96.0529 %
Total Number of Instances             1208

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.948     0.82       0.765     0.948     0.847      0.68     N
                 0.18      0.052      0.553     0.18      0.271      0.68     P
Weighted Avg.    0.747     0.619      0.709     0.747     0.696      0.68

=== Confusion Matrix ===

   a   b   <-- classified as
 845  46 |   a = N
 260  57 |   b = P
```

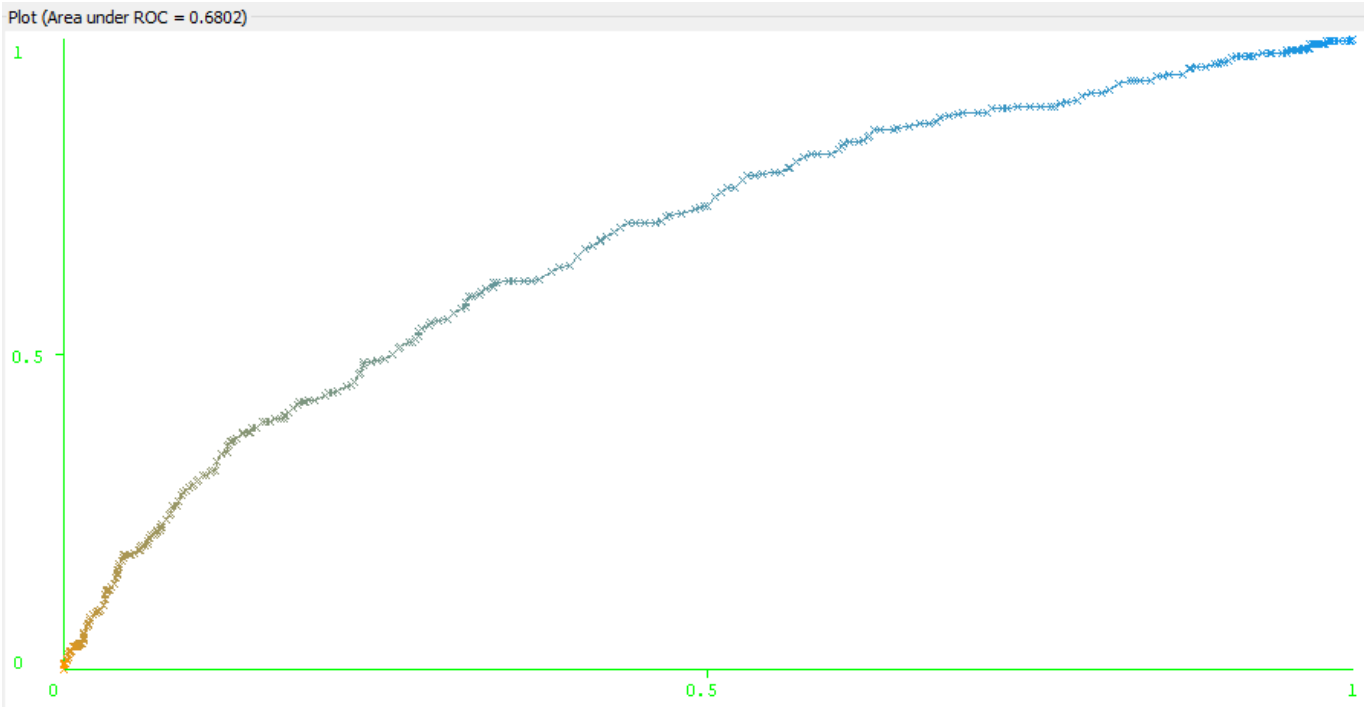
El porcentaje de clasificación correcta fue del 74,66%, un poco menor al caso de

validación cruzada.

Curva ROC para Negativos



Curva ROC para Positivos



El área bajo la curva es de 0.68 y resulto ser un poco menor a la obtenida en la validación cruzada.

5.2.6. Objetivo: Dividir a los usuarios en clusters según la cantidad de películas calificadas de cada genero y analizar cada uno de ellos

En el primer caso, en donde solo se consideran la cantidad de calificaciones por genero, se genero un archivo .csv el que para cada calificación solo tiene el ID del usuario y los géneros de la película. En el segundo caso se uso el archivo .csv con la totalidad de los datos.

Código en Anexo I: 6.9.1 Generar el primer fichero

Para pasar los datos de los archivos .csv a la base de datos, se uso la herramienta “Spoon”.

5.2.6.1. Transformación del primer caso

SQL para Crear la tabla destino:

Código en Anexo I: 6.9.2.1 Crear la tabla destino

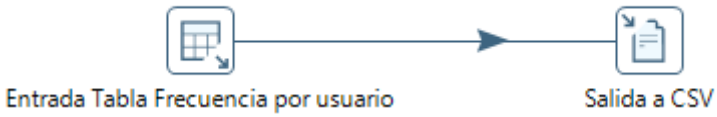
SQL para crear la tabla con la cantidad de películas por genero:

Código en Anexo I: 6.9.2.2 Crear la tabla con cantidad de películas por genero

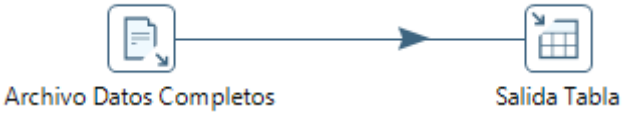
SQL para agrupar por cada usuario la cantidad de películas vistas por genero y guardar el resultado en la tabla anterior:

Código en Anexo I: 6.9.2.3 Agrupar por cada usuario la cantidad de películas vistas por genero

Transformación de la tabla resultante a .csv



5.2.6.2. Transformación del segundo caso



SQL para crear la tabla destino:

Código en Anexo I: 6.9.3.1 Crear la tabla destino

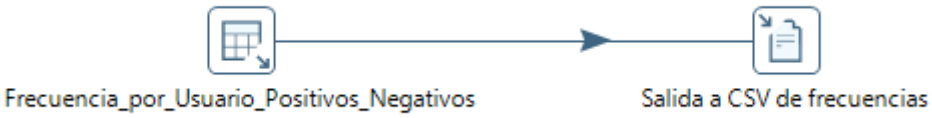
SQL para crear la tabla con la cantidad de películas, positivos y negativos por genero:

Código en Anexo I: 6.9.3.2 Crear la tabla con la cantidad de películas, positivos y negativos por genero

SQL para agrupar por cada usuario la cantidad de películas, positivos y negativos por genero y guardar el resultado en la tabla anterior:

Código en Anexo I: 6.9.3.3 Agrupar por cada usuario la cantidad de películas, positivos y negativos por genero

Transformación de la tabla resultante a .csv



Ahora teniendo los dos archivos .csv resultantes de realizar las agrupaciones usando MySQL, pasamos a la generar los clusters usando el algoritmos de K-Medias para distintas cantidades de grupos, de los cuales determinaremos la mejor agrupación usando el coeficiente de silueta.

5.2.6.3. División en clusters del primer caso

Leemos el archivo .csv y eliminamos el ID para que no afecte al agrupamiento, normalizamos los valores, calculamos el algoritmo de K-Medias para 2 a 8 clusters, calculamos la matriz de distancias de los valores del data frame y calculamos el coeficiente de silueta de cada agrupación realizada.

Código en Anexo I: 6.9.4 División en clusters del primer caso

Analizamos los resultados del coeficiente de silueta de cada agrupamiento:

```
> summary(coef.silueta.kmeans2)

Silhouette of 6040 units in 2 clusters from silhouette.default(x = kmeans2$cluster, dist =
distancias.usuarios) :
 Cluster sizes and average silhouette widths:
      5188      852
0.7122784 0.2402067
Individual silhouette widths:
    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-0.1732  0.5284  0.7689  0.6457  0.8082  0.8236

> summary(coef.silueta.kmeans3)

Silhouette of 6040 units in 3 clusters from silhouette.default(x = kmeans3$cluster, dist =
distancias.usuarios) :
 Cluster sizes and average silhouette widths:
      1268      354     4418
```

```
0.1724545 0.2170480 0.6538157
Individual silhouette widths:
    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-0.2400  0.3340  0.6459  0.5272  0.7566  0.7811

> summary(coef.silueta.kmeans4)

Silhouette of 6040 units in 4 clusters from silhouette.default(x = kmeans4$cluster, dist =
distancias.usuarios) :
 Cluster sizes and average silhouette widths:
      1433     3814      592      201
0.1016130 0.6204207 0.1702376 0.2026672
Individual silhouette widths:
    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-0.2923  0.2151  0.5082  0.4393  0.7133  0.7540

> summary(coef.silueta.kmeans5)

Silhouette of 6040 units in 5 clusters from silhouette.default(x = kmeans5$cluster, dist =
distancias.usuarios) :
 Cluster sizes and average silhouette widths:
      3722      628      181      530      979
0.6053092 0.1157456 0.1998486 0.1686453 0.1515372
Individual silhouette widths:
    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-0.2056  0.2122  0.4642  0.4304  0.6993  0.7448

> summary(coef.silueta.kmeans6)

Silhouette of 6040 units in 6 clusters from silhouette.default(x = kmeans6$cluster, dist =
distancias.usuarios) :
 Cluster sizes and average silhouette widths:
       345     3310      577      587     1101      120
0.1129643 0.5651534 0.1631989 0.1084396 0.1321009 0.1771424
Individual silhouette widths:
    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-0.2324  0.1677  0.3514  0.3699  0.6450  0.7117

> summary(coef.silueta.kmeans7)

Silhouette of 6040 units in 7 clusters from silhouette.default(x = kmeans7$cluster, dist =
distancias.usuarios) :
 Cluster sizes and average silhouette widths:
        187      1433        34       435       392      3012       547
0.18907581 0.06361318 0.22830845 0.11408890 0.12079718 0.57304743 0.15678640
Individual silhouette widths:
    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-0.3013  0.1205  0.3067  0.3383  0.6312  0.7134

> summary(coef.silueta.kmeans8)

Silhouette of 6040 units in 8 clusters from silhouette.default(x = kmeans8$cluster, dist =
distancias.usuarios) :
 Cluster sizes and average silhouette widths:
       999      249      664      200      546       34      168     3180
0.1403607 0.1150006 0.1205256 0.1796001 0.1318671 0.2182019 0.1788958 0.5687143
Individual silhouette widths:
    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-0.2192  0.1592  0.3405  0.3647  0.6407  0.7140
```

Vemos que el agrupamiento con la media mas alta para el coeficiente de silueta es el correspondiente a 2 grupos, por lo tanto elegimos ese para trabajar. Luego añadimos el atributo del cluster obtenido al data frame con las frecuencias por usuario y lo guardamos en un .csv para un posterior análisis:

Código en Anexo I: 6.9.5 Guardar clusters del primer caso

5.2.6.4. División en clusters del segundo caso

Realizamos los mismos pasos que en el caso anterior para obtener los coeficientes de silueta de las distintas agrupaciones:

Código en Anexo I: 6.9.6 División en clusters del segundo caso

```
> summary(coef.silueta.kmeans_P_N_2)

Silhouette of 6040 units in 2 clusters from silhouette.default(x = kmeans_P_N_2$cluster, dist =
distancias.usuarios_P_N) :
 Cluster sizes and average silhouette widths:
       849     5191
0.1803411 0.7087615
Individual silhouette widths:
    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-0.2262  0.5262  0.7619  0.6345  0.8034  0.8184

> summary(coef.silueta.kmeans_P_N_3)

Silhouette of 6040 units in 3 clusters from silhouette.default(x = kmeans_P_N_3$cluster, dist =
distancias.usuarios_P_N) :
 Cluster sizes and average silhouette widths:
       320     1273     4447
0.1574147 0.1324087 0.6522683
Individual silhouette widths:
    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-0.2707  0.3102  0.6438  0.5165  0.7524  0.7766

> summary(coef.silueta.kmeans_P_N_4)

Silhouette of 6040 units in 4 clusters from silhouette.default(x = kmeans_P_N_4$cluster, dist =
distancias.usuarios_P_N) :
 Cluster sizes and average silhouette widths:
       3788       593      1459       200
0.61292507 0.11500269 0.06885033 0.12177680
Individual silhouette widths:
    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-0.3127  0.1806  0.4976  0.4164  0.7026  0.7444

> summary(coef.silueta.kmeans_P_N_5)

Silhouette of 6040 units in 5 clusters from silhouette.default(x = kmeans_P_N_5$cluster, dist =
distancias.usuarios_P_N) :
 Cluster sizes and average silhouette widths:
       3714       188       638       963       537
0.60432528 0.10752363 0.07799175 0.11993386 0.11149570
Individual silhouette widths:
    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-0.2449  0.1828  0.4661  0.4122  0.6949  0.7397
```

```
> summary(coef.silueta.kmeans_P_N_6)

Silhouette of 6040 units in 6 clusters from silhouette.default(x = kmeans_P_N_6$cluster, dist = distancias.usuarios_P_N) :
 Cluster sizes and average silhouette widths:
        750      3429       445       983       133       300
0.12387422 0.55819124 0.05240789 0.07975923 0.07844099 0.12012277
Individual silhouette widths:
    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-0.2494  0.1405  0.3489  0.3568  0.6380  0.6992

> summary(coef.silueta.kmeans_P_N_7)

Silhouette of 6040 units in 7 clusters from silhouette.default(x = kmeans_P_N_7$cluster, dist = distancias.usuarios_P_N) :
 Cluster sizes and average silhouette widths:
        645      3020       209        47       402      1302       415
0.12090606 0.55150979 0.10124343 0.06956333 0.09020737 0.03015028 0.06027545
Individual silhouette widths:
     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
-0.31130  0.08025  0.26920  0.30940  0.60690  0.69420

> summary(coef.silueta.kmeans_P_N_8)

Silhouette of 6040 units in 8 clusters from silhouette.default(x = kmeans_P_N_8$cluster, dist = distancias.usuarios_P_N) :
 Cluster sizes and average silhouette widths:
        158      3024        34       227       264      1072       666       595
0.11513776 0.54552704 0.07707686 0.10315116 0.04934162 0.09614081 0.08735061 0.09580852
Individual silhouette widths:
    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-0.2559  0.1095  0.2847  0.3187  0.6029  0.6908
```

Al igual que en el caso anterior, la media mas alta resulto ser la de la agrupación en dos clusters, por lo tanto generamos el .csv correspondiente para su posterior análisis:

Código en Anexo I: 6.9.7 Guardar clusters del segundo caso

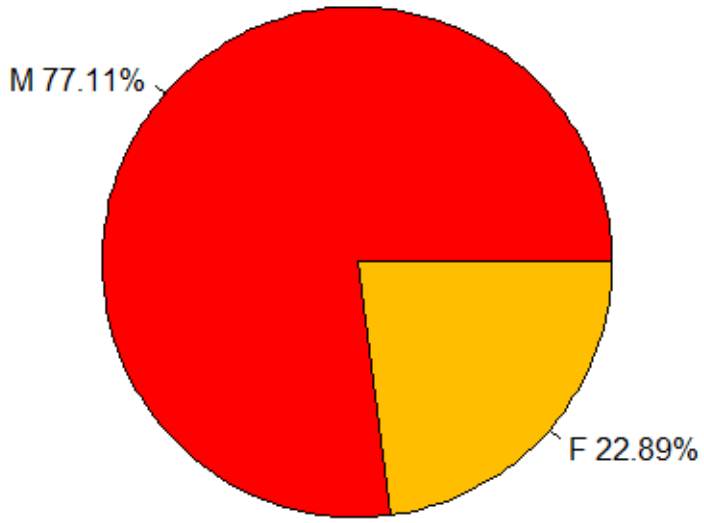
Una vez obtenidas las agrupaciones para ambos casos vistos, se realizo una comparación entre los dos datasets, mediante la cual se determino que ambos resultaron ser prácticamente idénticos, por lo tanto los análisis posteriores en este trabajo fueron realizados tomando el dataset del primer caso (En el cual se consideran solamente la cantidad de películas de cada genero que vio el usuario).

5.2.6.5. Análisis del primer cluster

5.2.6.5.1. Distribución según el genero de los usuarios

Código en Anexo I: 6.9.8.1 Distribución según el genero de los usuarios

Gráfico



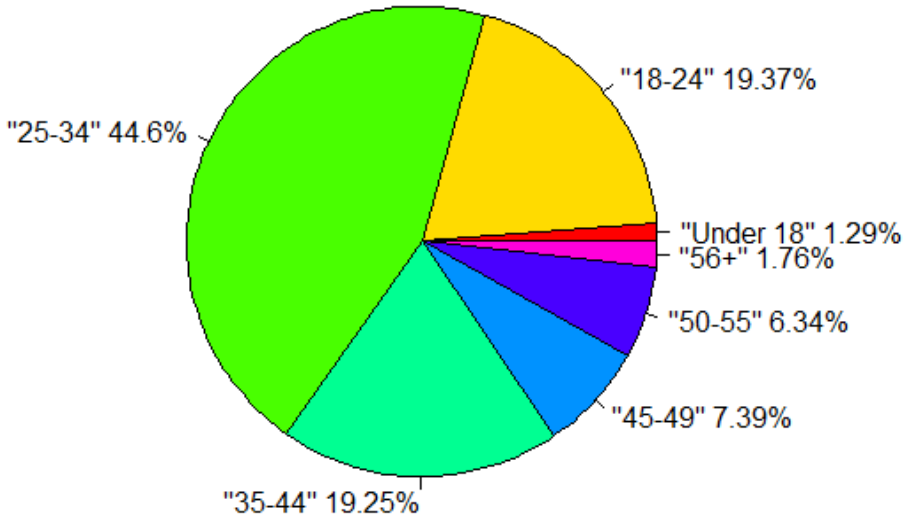
Evaluación e interpretación

En este cluster la proporción de varones resulta ser mayoritaria abarcando un poco mas que las tres cuartas partes del total de los usuarios.

5.2.6.5.2. Distribución según la edad de los usuarios

Código en Anexo I: 6.9.8.2 Distribución según la edad de los usuarios

Gráfico



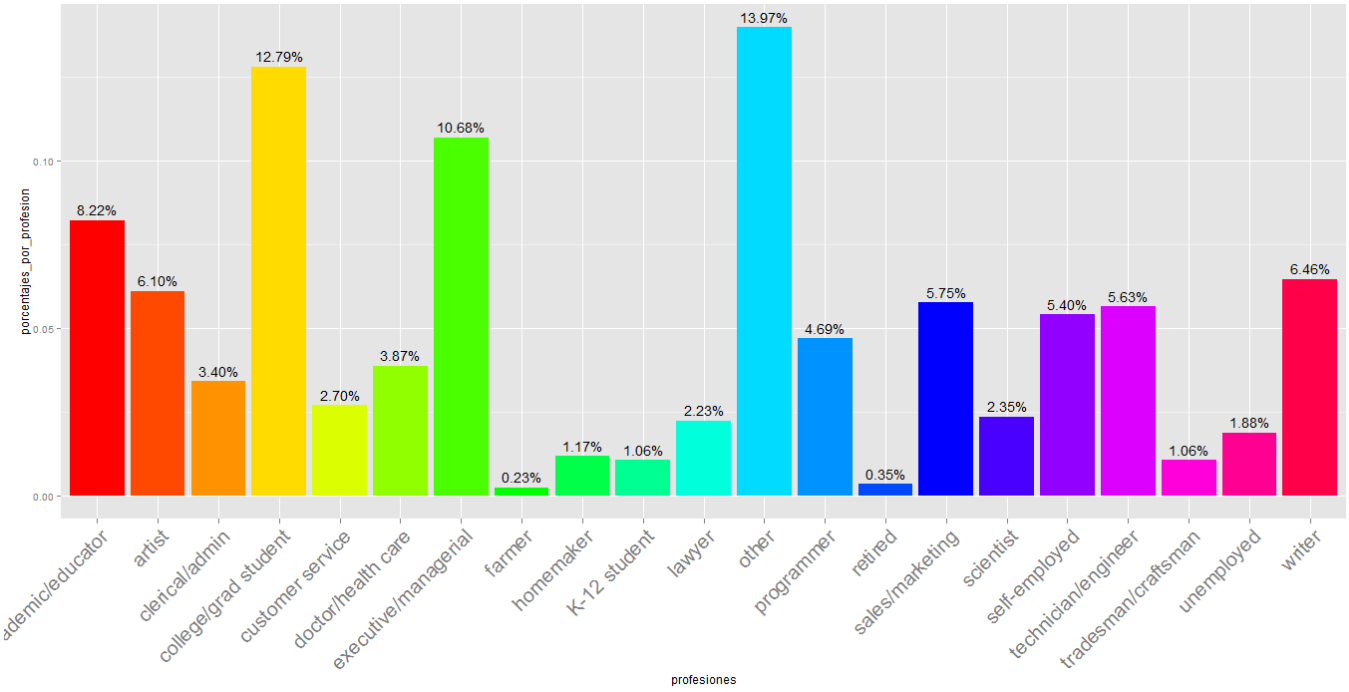
Evaluación e interpretación

Como podemos ver, casi la mitad de los usuarios pertenecientes a este cluster tienen entre 25 y 34 años. Si tomamos el rango que va desde los 18 hasta los 44 años, abarcamos mas del 80% de los usuarios del cluster.

5.2.6.5.3. Distribución según la profesión de los usuarios

Código en Anexo I: 6.9.8.3 Distribución según la profesión de los usuarios

Gráfico



Evaluación e interpretación

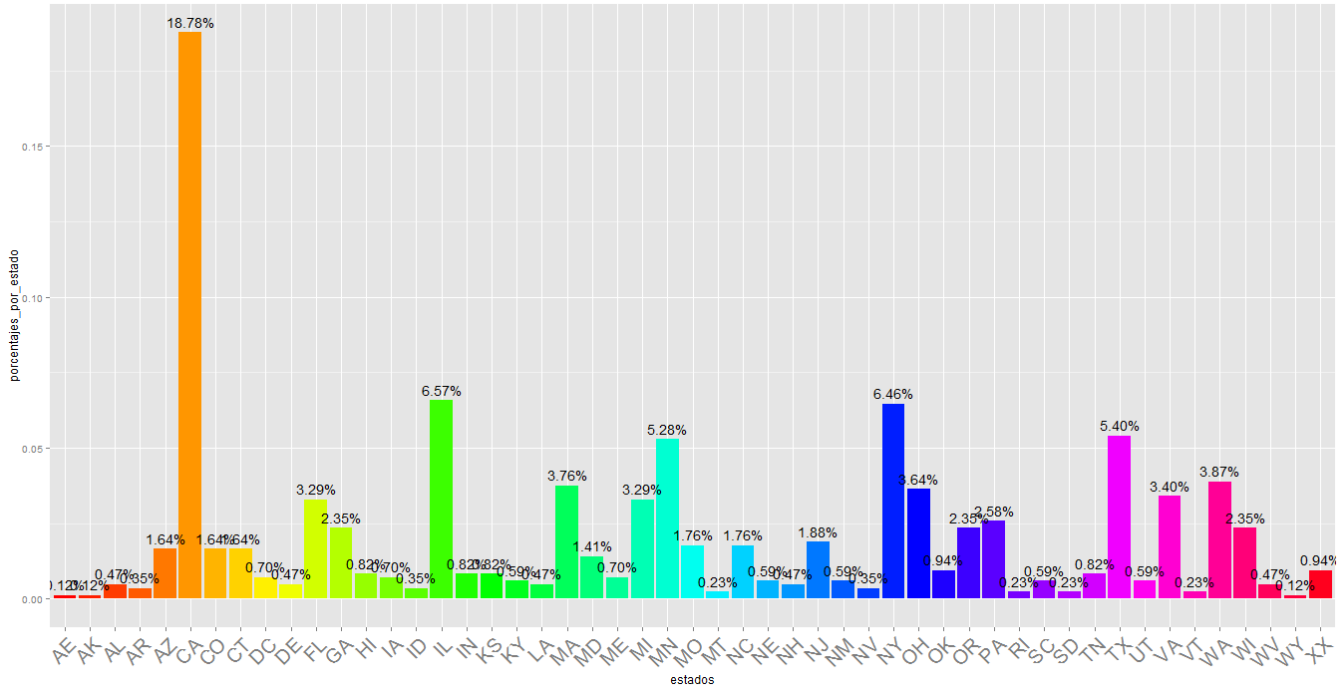
La mayoría de los usuarios del cluster no pertenece a ninguna de las profesiones consideradas en el dataset. Las profesiones con mayor frecuencia son en orden descendente: Estudiante, ejecutivo, académico, escritor, artista y vendedor.

Los estudiantes, ejecutivos, académicos y profesiones no consideradas abarcan mas del 40% de los casos.

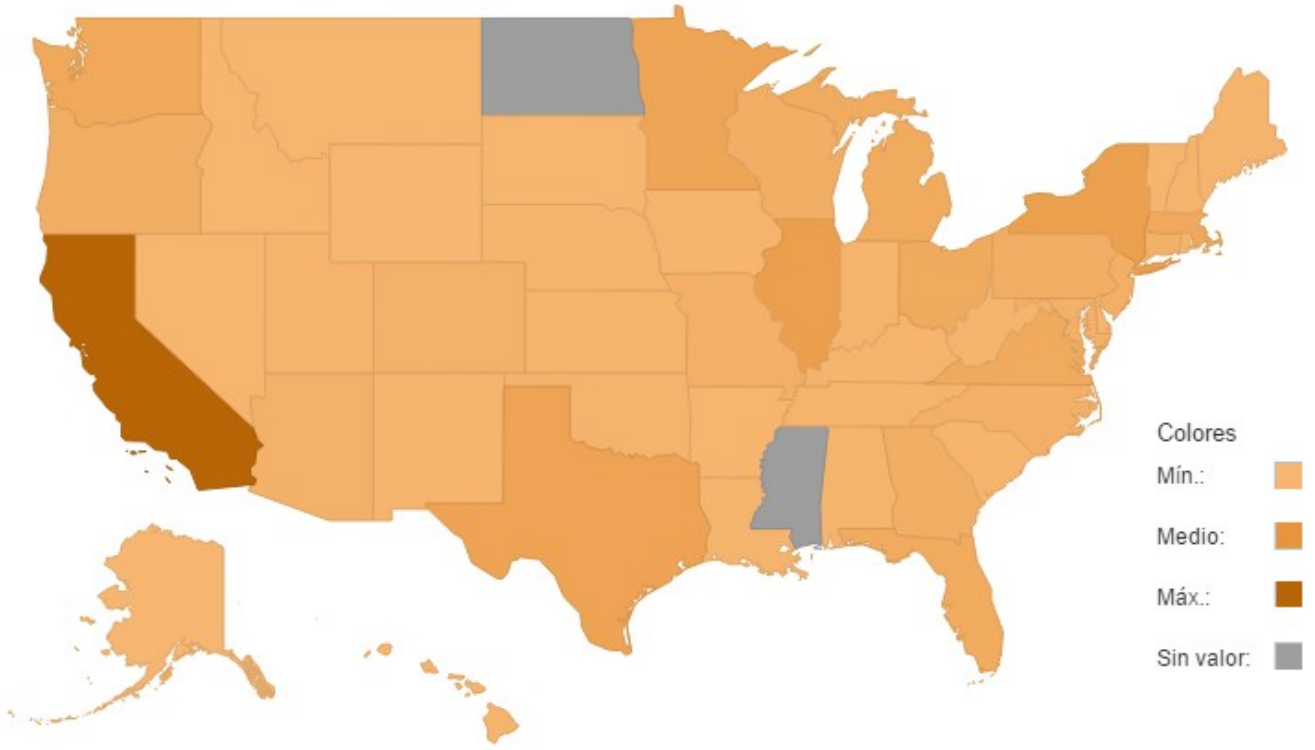
5.2.6.5.4. Distribución según el estado en donde viven los usuarios

Código en Anexo I: 6.9.8.4 Distribución según el estado en donde viven los usuarios

Gráfico



Mapa por estados



Evaluación e interpretación

Puede verse que el porcentaje de usuarios que viven en california es bastante mayor al resto de los estados y abarca casi el 20% de los casos. También se nota a simple vista la ausencia de usuarios de Mississippi y Dakota del Norte.

Ademas de california, el resto de estados en donde mas se concentran los usuarios son: Illinois, Nueva York, Texas,y Minnesota.

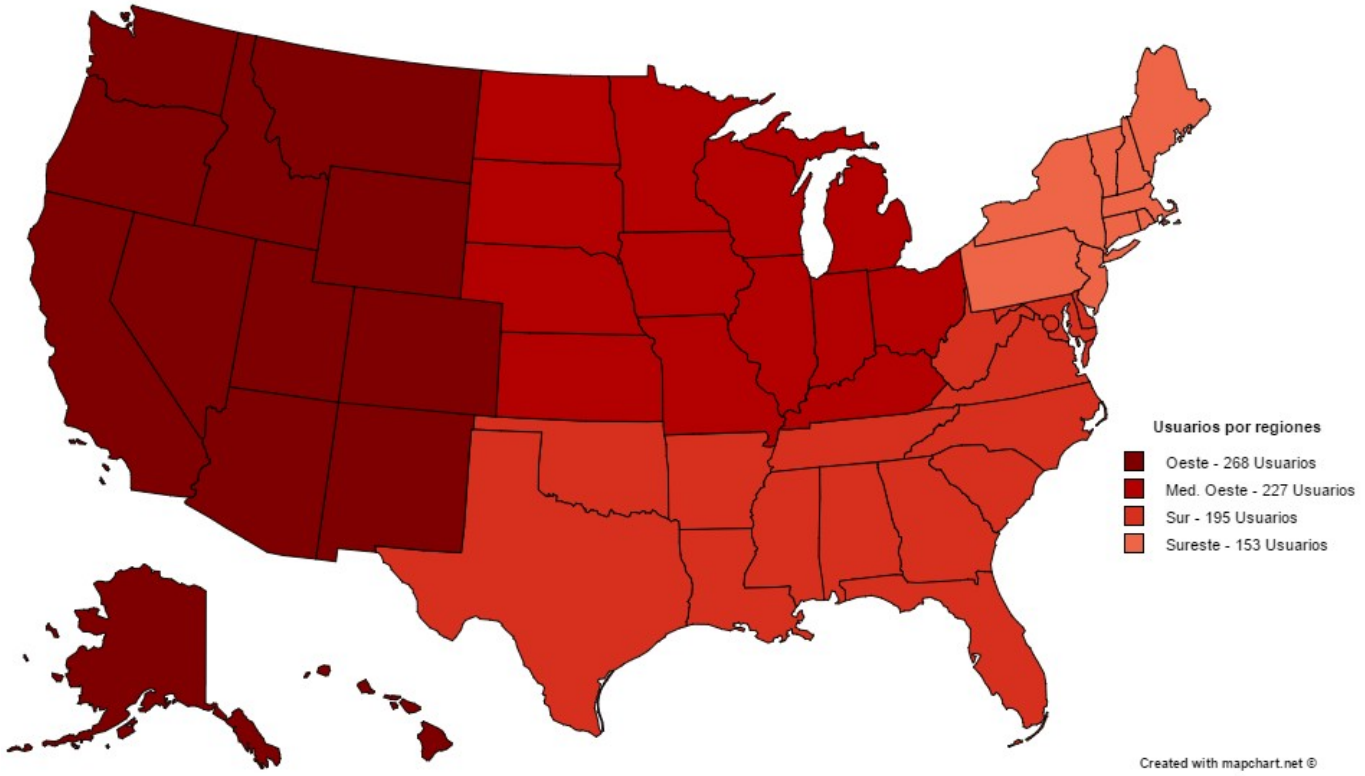
5.2.6.5.5. Agrupamiento por regiones

Código en Anexo I: 6.9.8.5 Agrupamiento por regiones

Salida

Oeste: 268
Medio Oeste: 227
Sur: 195
Sureste: 153

Mapa por regiones



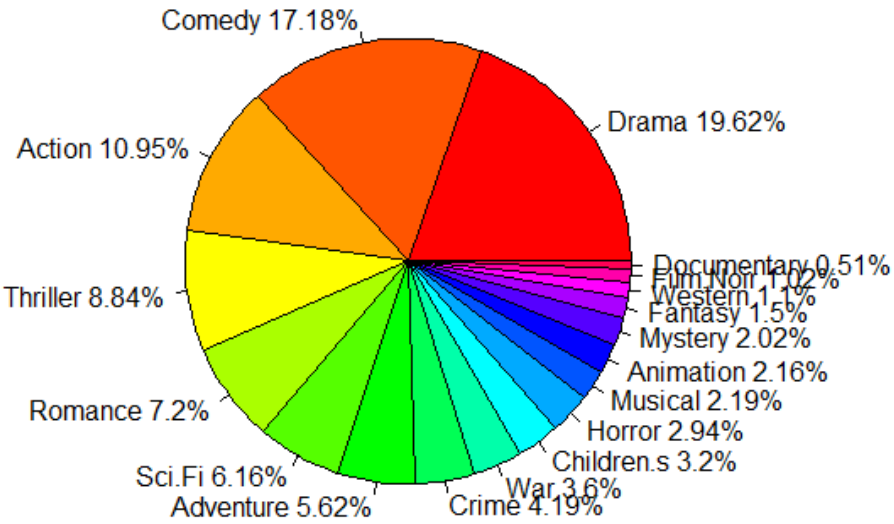
Evaluación e interpretación

El orden de concentración de usuarios en orden descendente es: Oeste, Medio Oeste, Sur y Sureste. La diferencia de usuarios entre una región y la siguiente no es muy elevada.

5.2.6.5.6. Popularidad de los géneros de películas

Código en Anexo I: 6.9.8.6 Popularidad de los géneros de películas

Gráfico



Evaluación e interpretación

De manera similar a los análisis hechos previamente para el total de los usuarios del dataset, los géneros mas populares siguen siendo “Drama”, “Comedia” y “Acción”, los cuales abarcan casi el 50% de las calificaciones positivas del cluster.

5.2.6.5.7. Popularidad de géneros de películas por estado

Código en Anexo I: 6.9.8.7 Popularidad de géneros de películas por estado

Salida

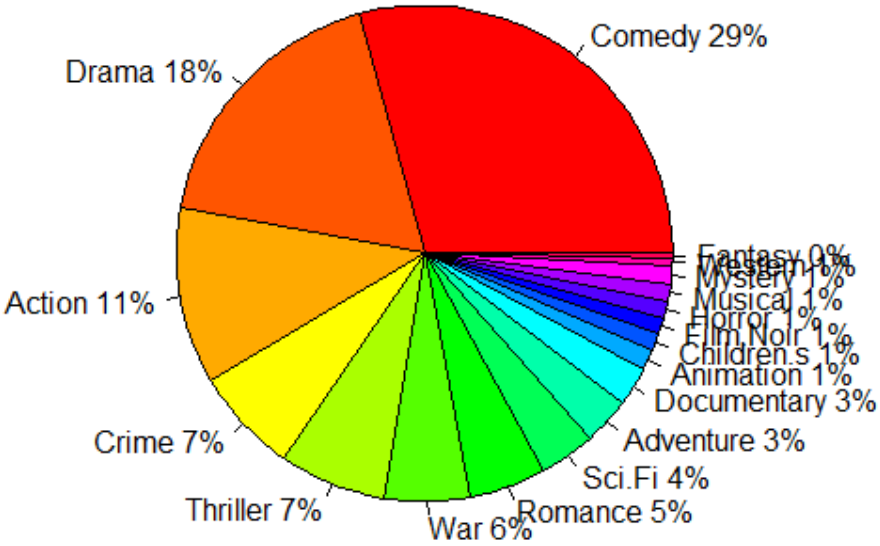
```
ESTADO: GENERO MAS POPULAR
AL: Drama
AK: Drama
AZ: Comedy
AR: Drama
AE: Comedy
CA: Drama
CO: Drama
CT: Drama
DE: Drama
DC: Drama
FL: Drama
GA: Drama
HI: Drama
ID: Drama
IL: Drama
IN: Comedy
IA: Comedy
KS: Drama
KY: Drama
LA: Drama
ME: Drama
MD: Drama
MA: Drama
MI: Drama
MN: Comedy
MO: Drama
```

MT: Drama
NE: Drama
NV: Comedy
NH: Drama
NJ: Drama
NM: Drama
NY: Drama
NC: Drama
OH: Drama
OK: Drama
OR: Drama
PA: Drama
RI: Drama
SC: Drama
SD: Drama
TN: Drama
TX: Drama
XX: Drama
UT: Drama
VT: Drama
VA: Drama
WA: Drama
WV: Drama
WI: Comedy
WY: Comedy

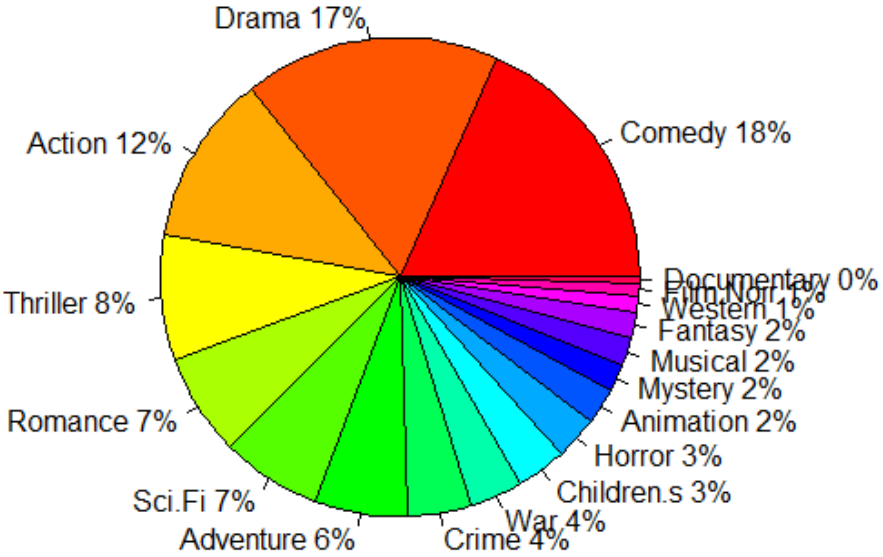
```
> print(generos_oeste)

      Comedy       Drama      Action       Crime    Thriller         War     Romance      Sci.Fi
         105          64          41          25          25          20          18          13
  Adventure Documentary   Animation  Children.s   Film.Noir      Horror     Musical     Mystery
          11           9           5           4           4           4           4           4
     Western     Fantasy
           2           1

> print(generos_medio_oeste)

      Comedy       Drama      Action    Thriller     Romance      Sci.Fi   Adventure       Crime
        2098        1971        1326         966         780         769         724         482
         War  Children.s      Horror   Animation     Mystery     Musical     Fantasy     Western
         416         383         340         263         220         219         193         134
  Film.Noir Documentary
          93          48

> print(generos_sur)

       Drama      Comedy      Action    Thriller      Sci.Fi     Romance         War   Adventure
         525         339         259         237         140         133         120         105
      Horror       Crime     Mystery     Musical  Children.s     Western     Fantasy   Animation
         103          95          72          57          53          49          32          29
  Film.Noir Documentary
          28          10

> print(generos_sureste)

       Drama      Comedy     Romance    Thriller      Action      Sci.Fi   Adventure       Crime
         353         276         108         103          95          63          49          47
 Children.s         War     Mystery   Animation     Musical   Film.Noir Documentary     Fantasy
          44          40          34          30          27          13          11          11
      Horror     Western
          11          11
```

Gráficos

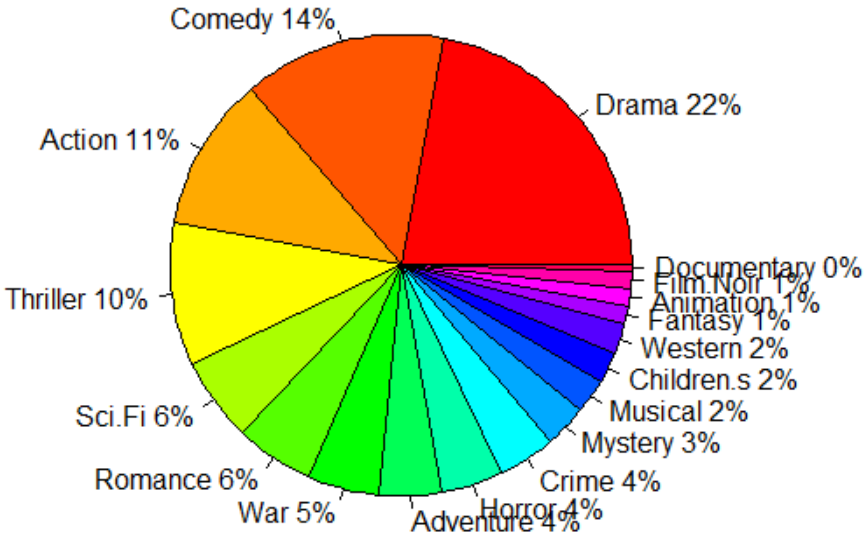
Oeste



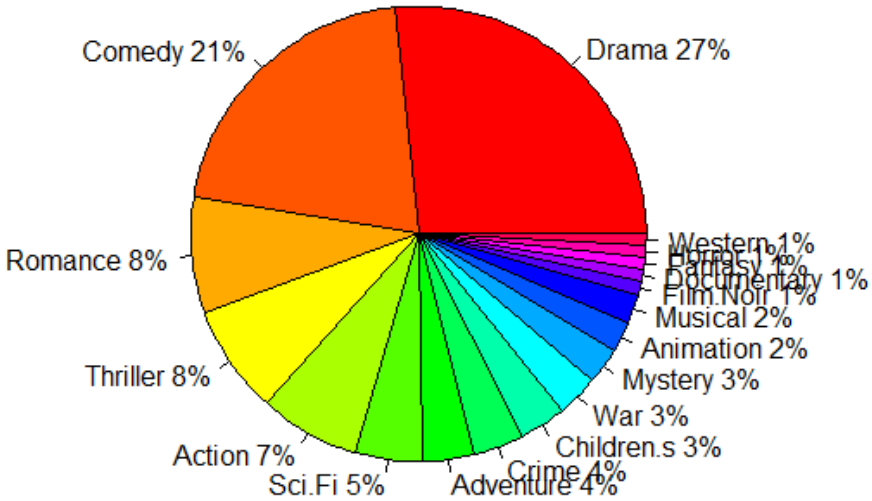
Medio Oeste



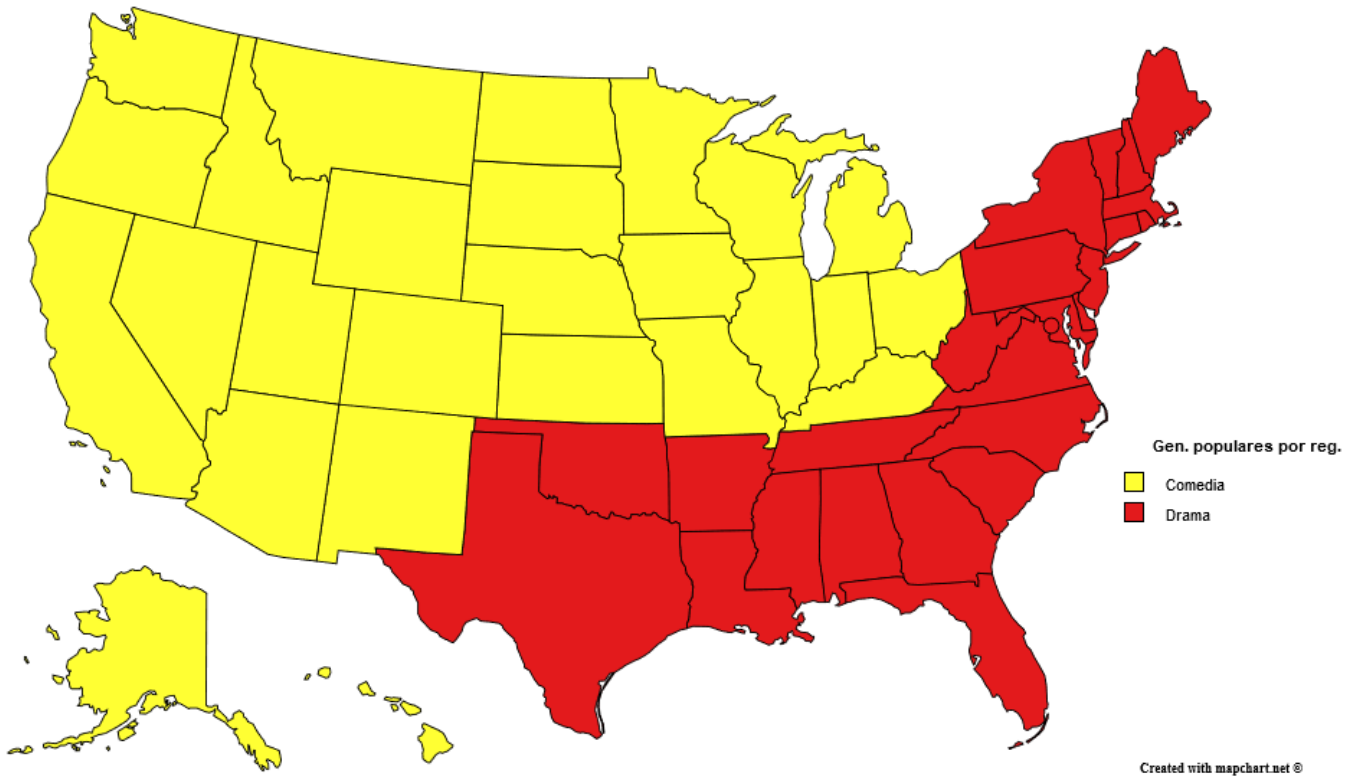
Sur



Sureste



Mapa de genero mas popular por región



Evaluación e interpretación

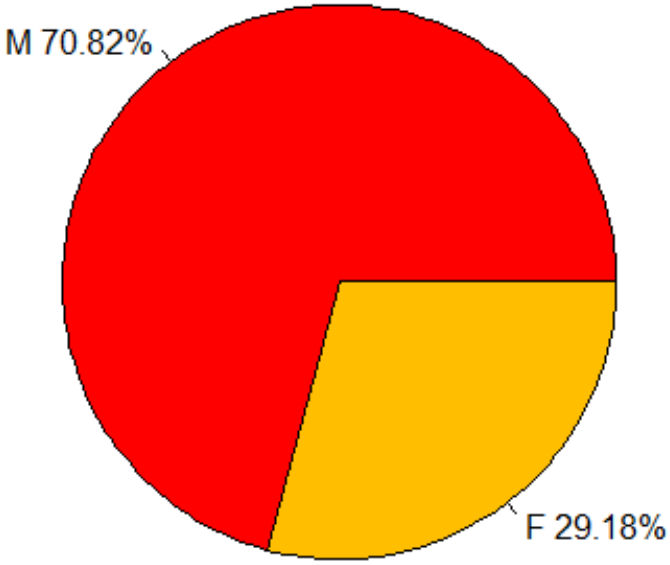
Al tomar los géneros mas populares por región es vez de por estado, podemos ver que existe una división en las preferencias de los usuarios, los que están en las regiones del oeste y medio oeste prefieren las películas de comedia, mientras que los que están en el sur o el sureste prefieren las películas dramáticas.

Otro aspecto a destacar es que en todas las regiones menos en el sureste, las temáticas mas populares son el drama, la comedia y la acción, mientras que en este ultimo la acción se ve desplazada por el romance, al igual que sucedía en las películas mas populares para las personas de genero femenino.

5.2.6.6. Análisis del segundo cluster

Se omitió escribir los algoritmos en R debido a que son esencialmente los mismos que los escritos en el apartado anterior.

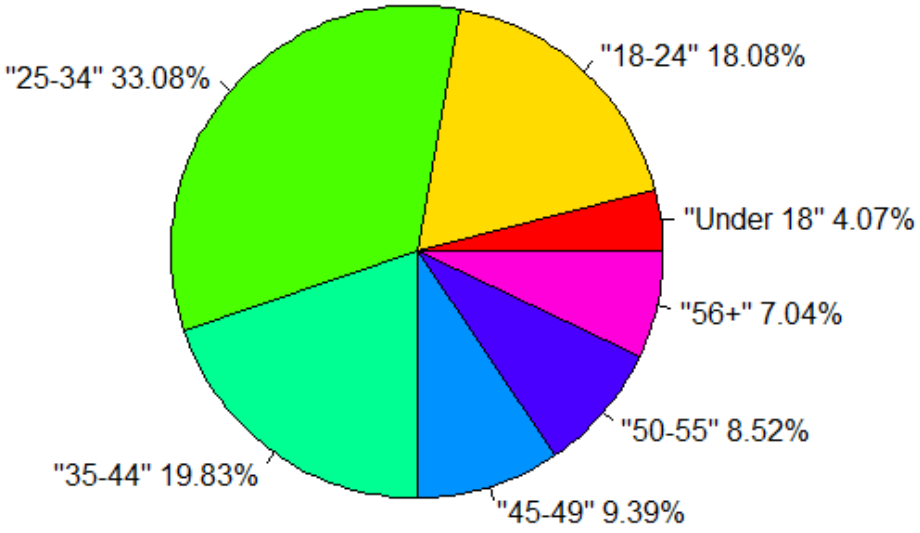
5.2.6.6.1. Distribución según el genero de los usuarios



Evaluación e interpretación

El porcentaje de usuarios de genero masculino resulta ser menor al del cluster analizado previamente, sin embargo sigue siendo mayoritario con mas del 70% de los casos.

5.2.6.6.2. Distribución según la edad de los usuarios



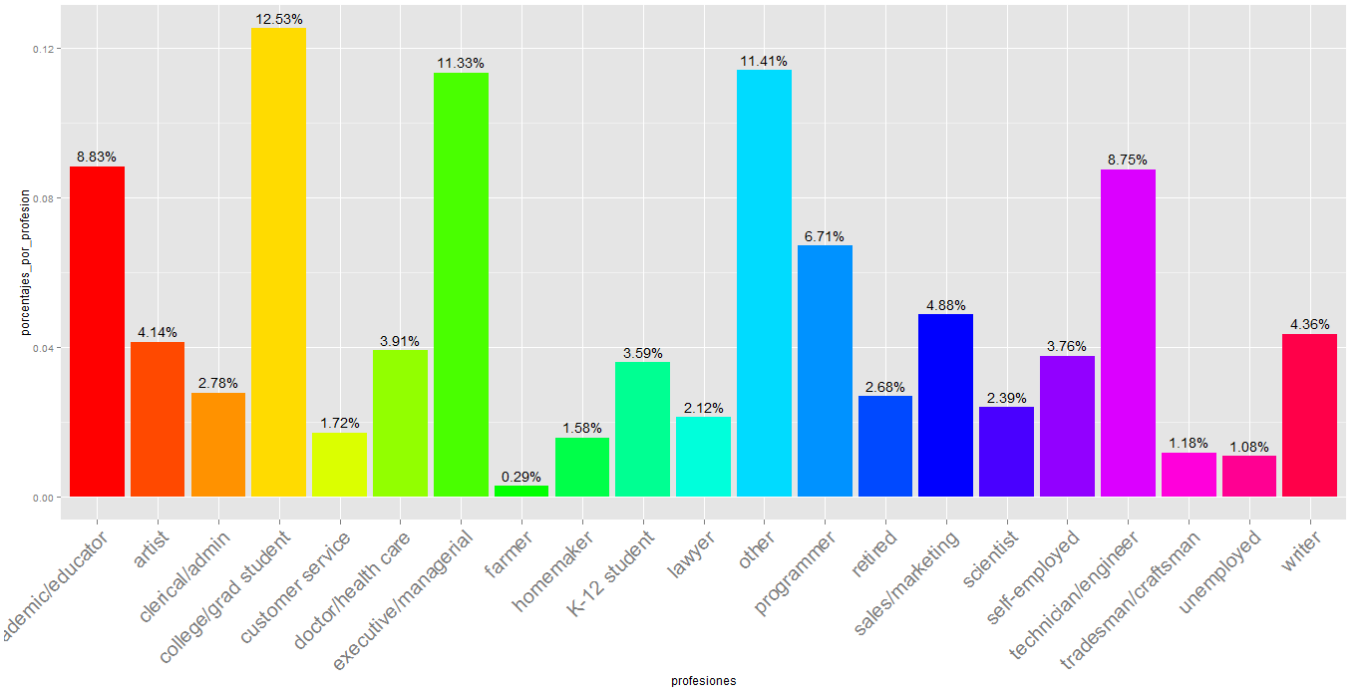
Evaluación e interpretación

Comparándolo con respecto al cluster anterior, puede verse una disminución del porcentaje correspondiente a las personas cuya edad esta entre los 25 y 34 años y un aumento en los rangos de edades que van de los 45 años en adelante y los que son menores de 18 años.

En este caso si tomamos el rango de usuarios cuya edad va desde los 18 hasta los 44

años, abarcamos alrededor del 70% de los casos.

5.2.6.6.3. Distribución según la profesión de los usuarios



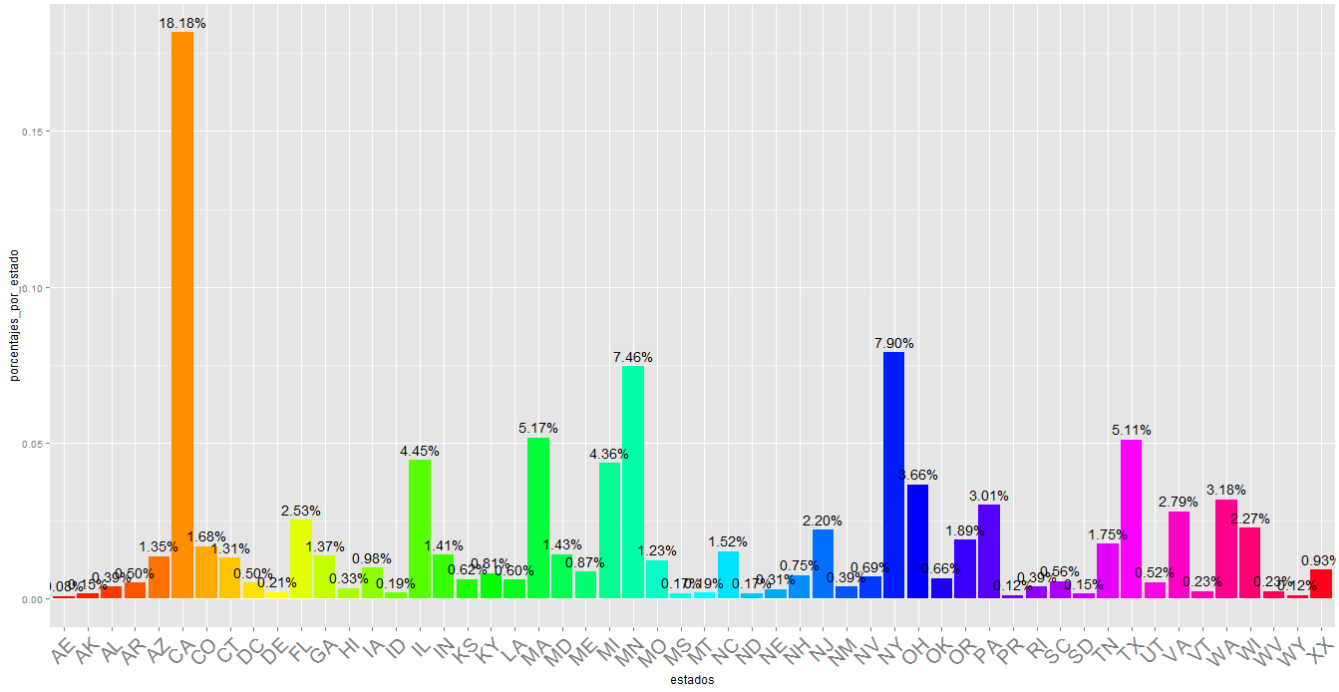
Evaluación e interpretación

En este cluster el porcentaje de las personas cuya profesión no fue considerada pasa a ocupar el segundo lugar, dejando el primer puesto a los usuarios que son estudiantes.

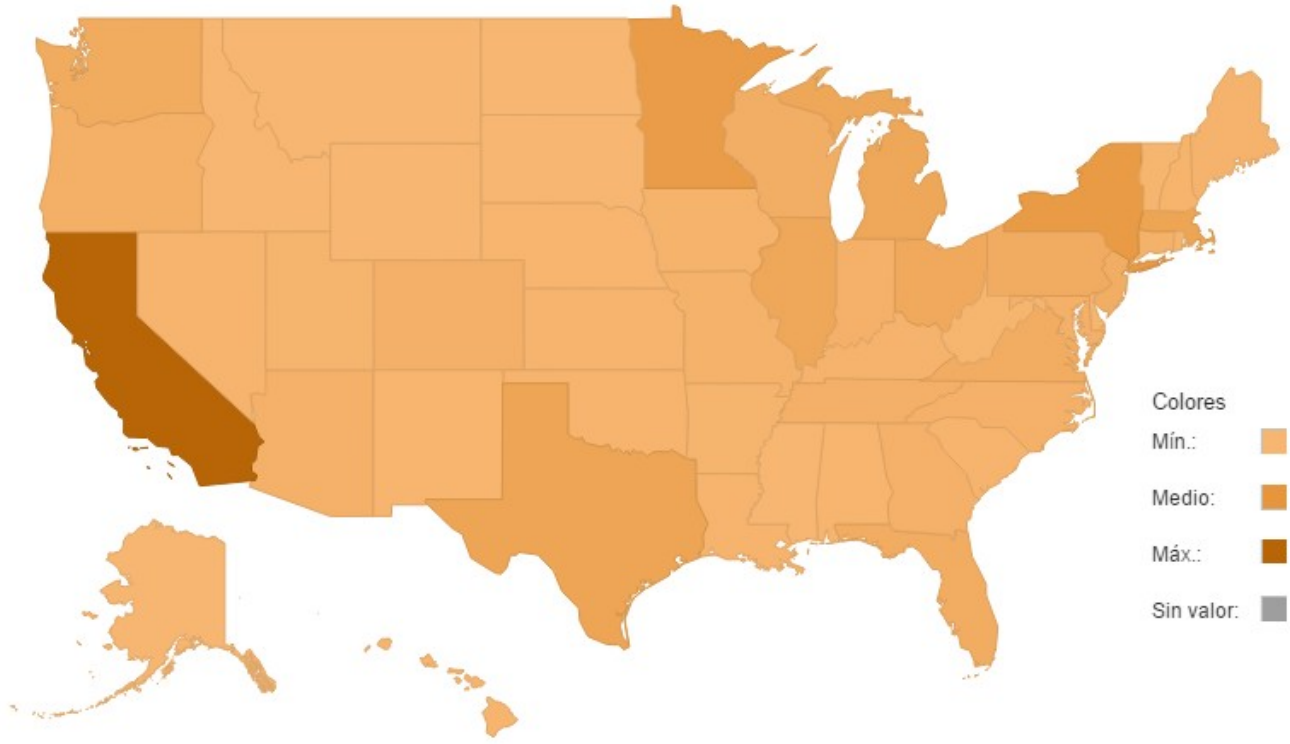
Una diferencia notable con respecto al cluster anterior es que los técnicos e ingenieros pasaron a ocupar el cuarto puesto, seguidos de los programadores en el quinto.

El orden resultante de profesiones mas populares (Sin considerar los casos no incluidos) es: Estudiante, ejecutivo, académico, técnico, programador, vendedor.

5.2.6.6.4. Distribución según el estado en donde viven los usuarios



Mapa por estados



Evaluación e interpretación

Con respecto a California sucede los mismo que en el cluster anterior, hay un nivel de usuarios que esta concentrado en dicho estado.

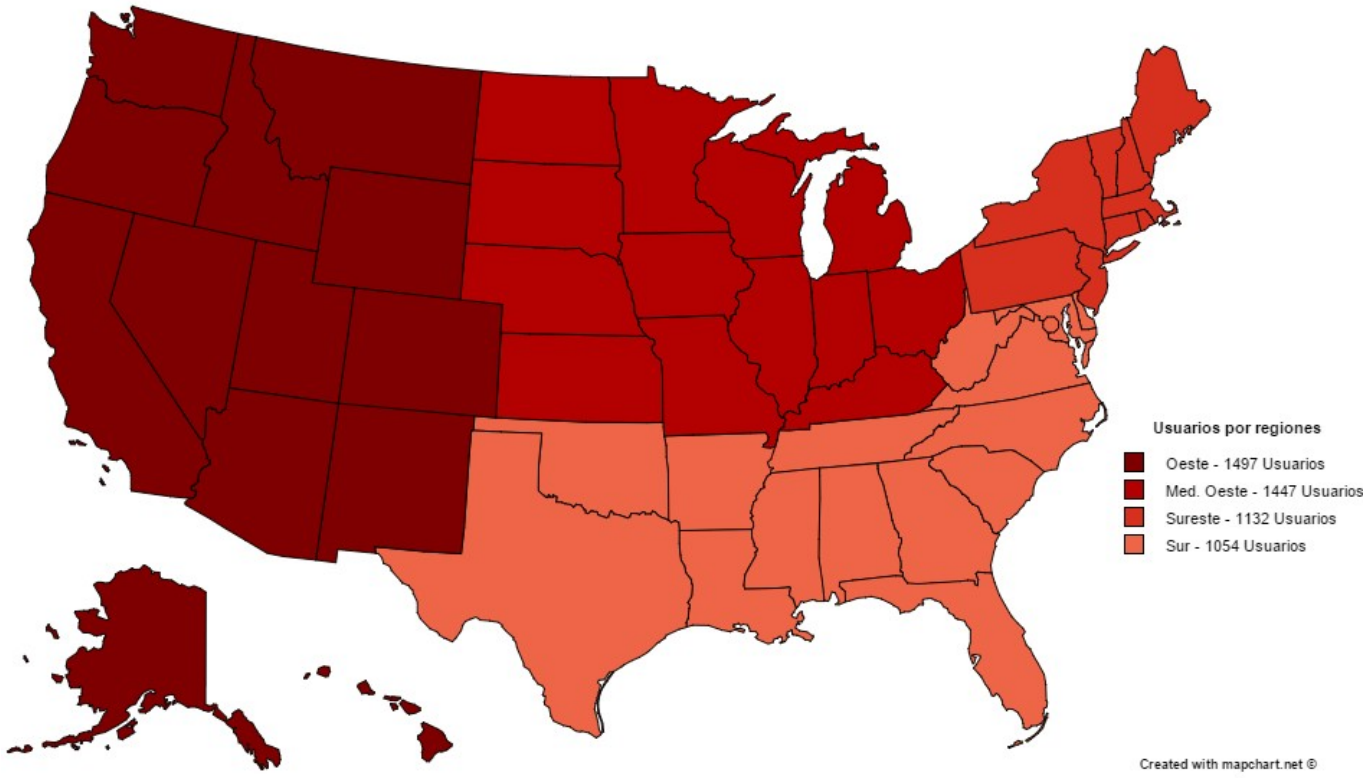
En el mapa puede apreciarse que en este caso no hay ausencia de usuarios en ninguno de los estados.

Ademas de California, el resto de estados en donde mas se concentran los usuarios son: Nueva York, Minnesota, Massachusetts y Texas.

5.2.6.6.5. Agrupamiento por regiones

Oeste: 1497
Medio Oeste: 1447
Sur: 1054
Sureste: 1132

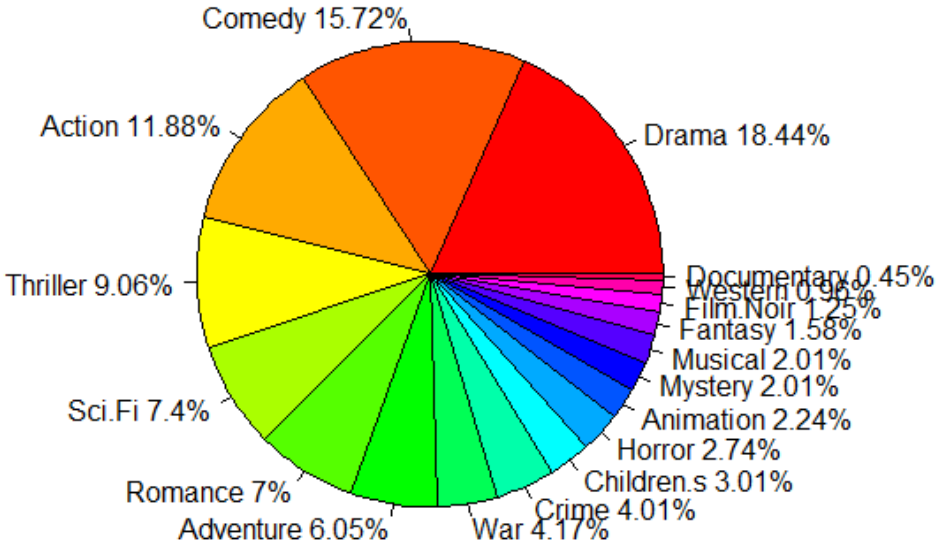
Mapa por regiones



Evaluación e interpretación

La diferencia con respecto al cluster anterior es que en este caso la cantidad de usuarios que hay en la región sureste es mayor a la cantidad que hay en la región del sur. (Ademas de que en este cluster la cantidad de usuarios es mucho mayor).

5.2.6.6.6. Popularidad de los géneros de películas



Evaluación e interpretación

El gráfico resulta ser bastante similar al del primer cluster, sin embargo dentro de los géneros mas populares hay una diferencia en el orden de “Sci Fi” y “Romance” los cuales estaban invertidos en el caso anterior.

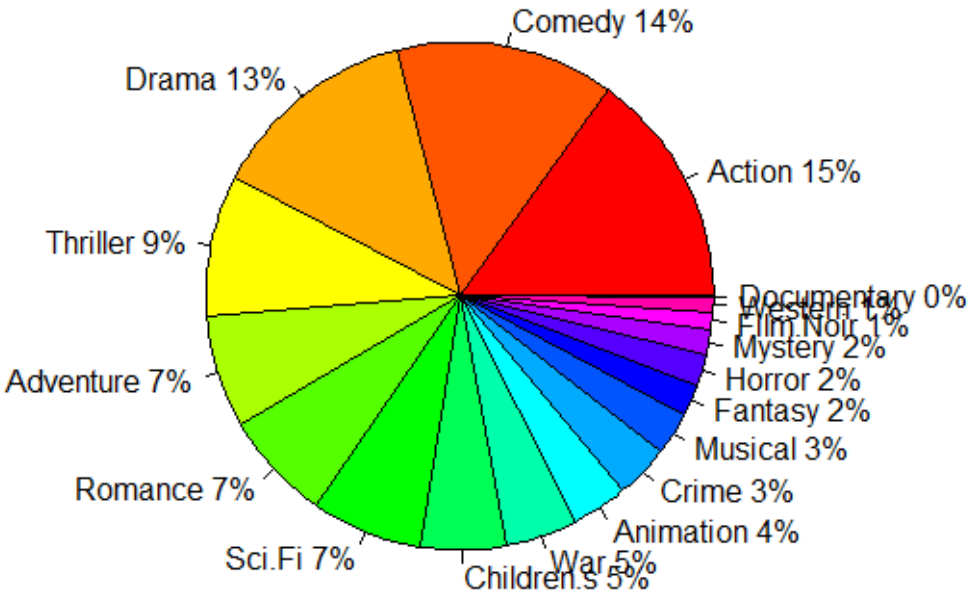
5.2.6.6.7. Popularidad de géneros de películas por estado

ESTADO: GENERO MAS POPULAR
AL: Action
AK: Drama
AZ: Drama
AR: Drama
AE: Action
CA: Drama
CO: Comedy
CT: Drama
DE: Comedy
DC: Drama
FL: Action
GA: Drama
HI: Comedy
ID: Drama
IL: Drama
IN: Comedy
IA: Drama
KS: Comedy
KY: Drama
LA: Drama
ME: Drama
MD: Drama
MA: Drama
MI: Drama
MN: Drama
MS: Drama
MO: Drama
MT: Comedy
NE: Action
NV: Drama

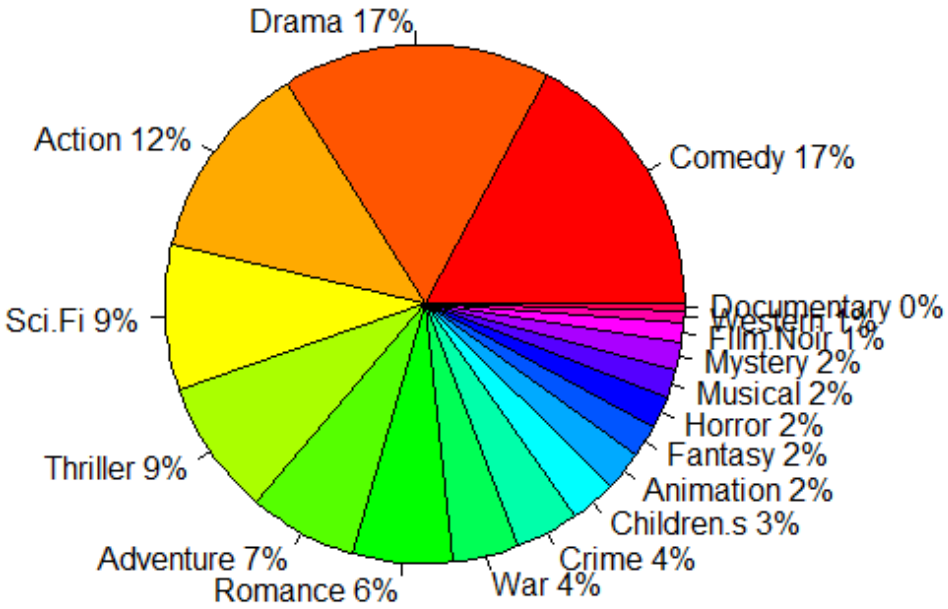
NH: Drama
NJ: Drama
NM: Drama
NY: Drama
NC: Drama
ND: Comedy
OH: Drama
OK: Drama
OR: Drama
PA: Drama
PR: Drama
RI: Comedy
SC: Drama
SD: Drama
TN: Drama
TX: Drama
XX: Drama
UT: Comedy
VT: Drama
VA: Drama
WA: Drama
WV: Drama
WI: Comedy
WY: Action

```
> print(generos_oeste)

      Action      Comedy       Drama    Thriller   Adventure     Romance      Sci.Fi  Children.s
         169         156         150         101          80          79          79          61
         War   Animation       Crime     Musical     Fantasy      Horror     Mystery   Film.Noir
          52          40          39          30          23          22          18          13
     Western Documentary
          10           1

> print(generos_medio_oeste)

      Comedy       Drama      Action      Sci.Fi    Thriller   Adventure     Romance         War
        2647        2579        1904        1394        1311        1031         963         634
       Crime  Children.s   Animation     Fantasy      Horror     Musical     Mystery   Film.Noir
         604         443         382         309         307         275         259         198
     Western Documentary
         102          69

> print(generos_sur)

        Drama      Action      Comedy    Thriller     Romance       Crime   Adventure      Sci.Fi
          319         269         252         196         164         113         103          80
     Musical         War  Children.s      Horror     Mystery   Animation     Fantasy   Film.Noir
           65          65          49          46          44          29          19          16
     Western Documentary
            9           3

> print(generos_sureste)

        Drama      Comedy      Action    Thriller      Sci.Fi     Romance         War   Adventure
          325         316         182         148         130         105          93          91
  Children.s       Crime   Animation     Musical      Horror     Mystery     Fantasy   Film.Noir
           72          71          65          49          42          31          28          19
     Western Documentary
           16           6
```

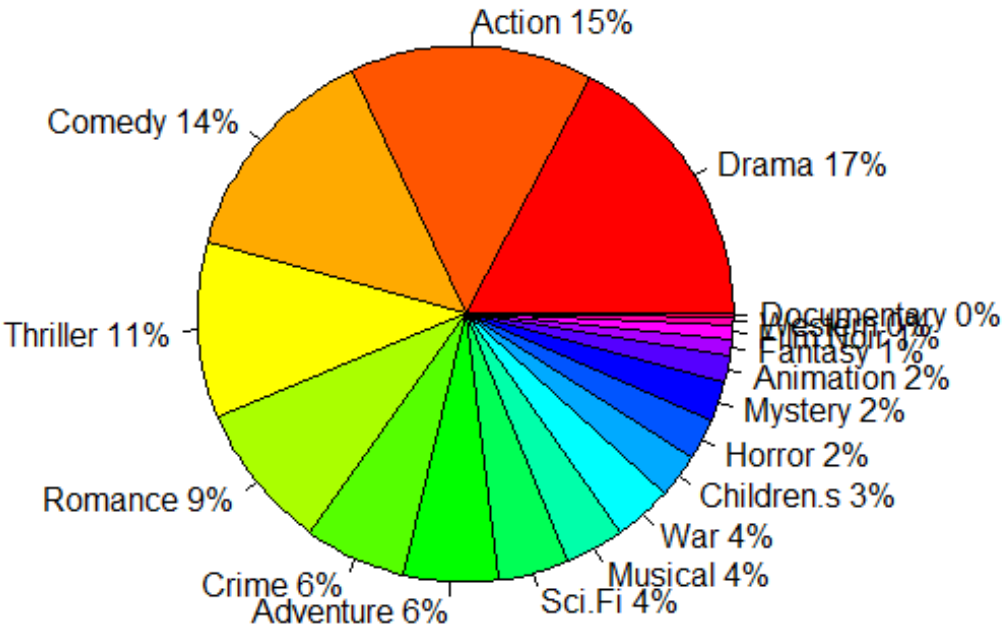
Oeste



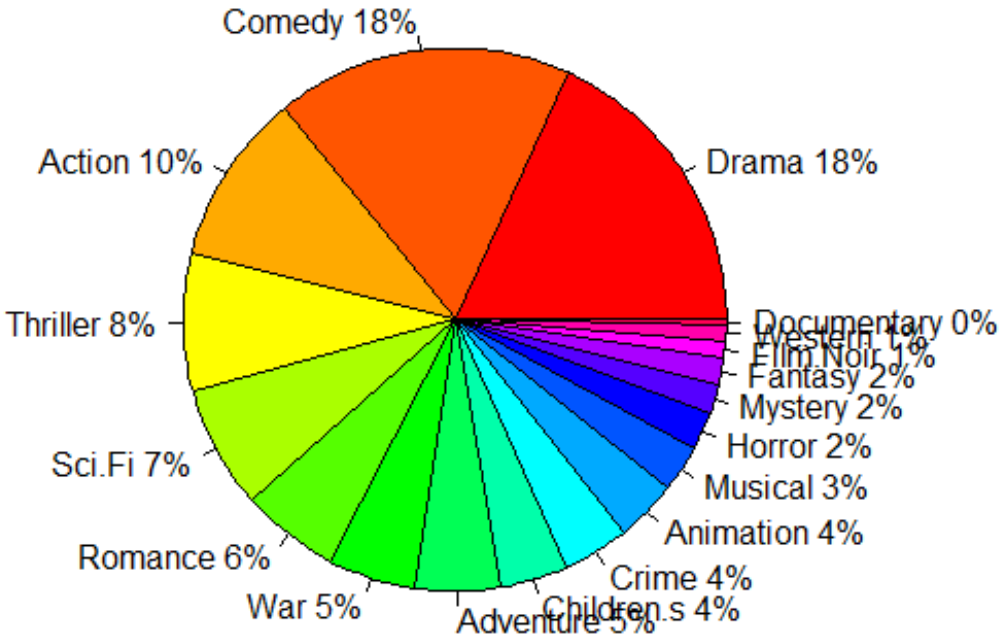
Medio Oeste



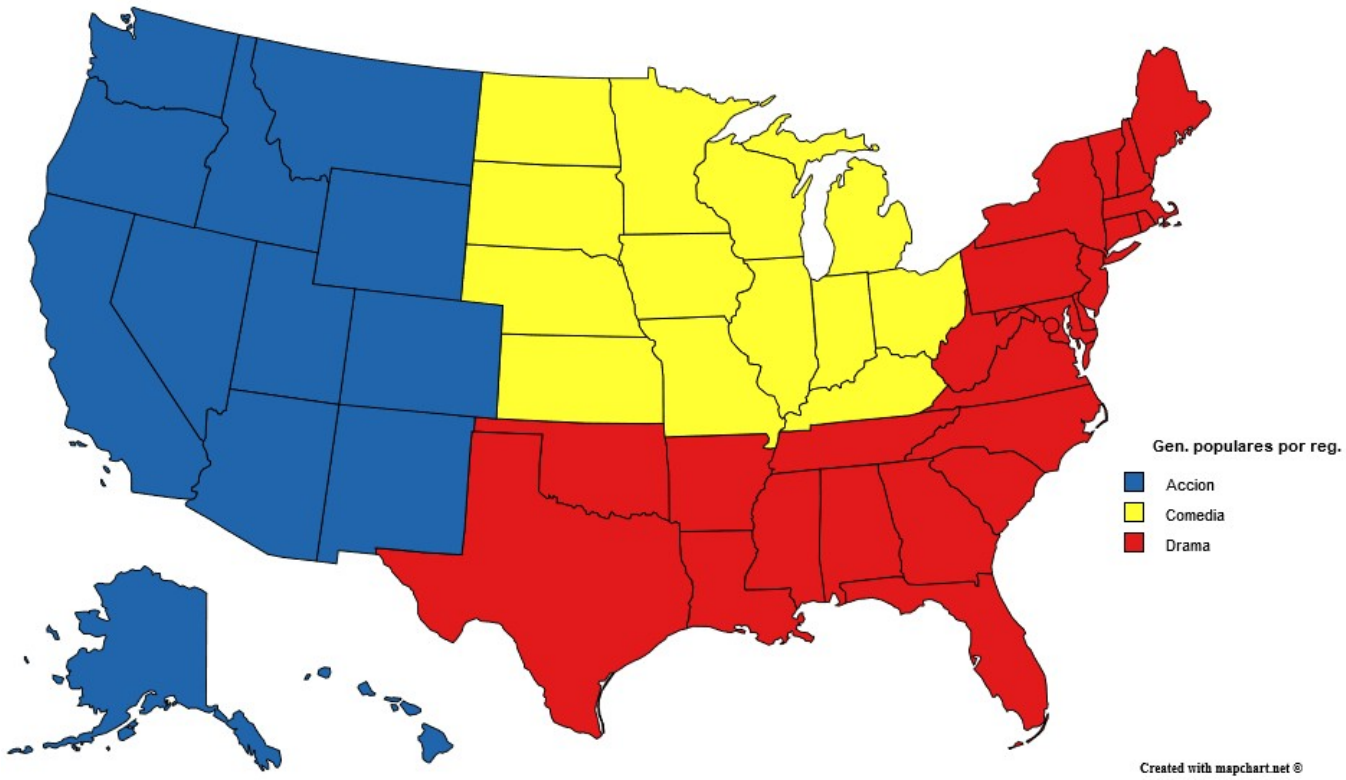
Sur



Sureste



Mapa de genero mas popular por región



Evaluación e interpretación

Un cambio notable con respecto al cluster anterior es que la acción paso a tener mayor popularidad tanto en la región oeste (En donde paso a ser el mas popular) como en el sur (Donde paso a ocupar el segundo puesto después del drama). Debido a esto nos quedaron tres divisiones en el mapa: El oeste que prefiere las películas de acción, el medio oeste que prefiere las comedias y el sur junto al sureste los cuales prefieren las películas dramáticas.

6. Anexo I: Código utilizado

6.1. Transformando el archivo de películas

Código en R

```
movies_transformadas <- read.csv("C:/movies_transformadas.csv")
generos_separados <- as.data.frame.matrix(xtabs(~Movie_ID+Movie_Genre,data =
movies_transformadas))
movies_transformadas <- movies_transformadas[-3]
movies_transformadas <- unique(movies_transformadas)
generos_separados <- cbind(Movie_ID = rownames(generos_separados), generos_separados)
movies <- merge(movies_transformadas,generos_separados,by = "Movie_ID")
write.csv(movies,"C:/movies_transformadas_binario.csv",row.names = FALSE)
```

6.2. Unificando los datos

Código en R

```
datos_completos_transformados$Valoration <-
```

```
ifelse(datos_completos_transformados$Rating>3,"P","N")
```

6.3. Calificaciones realizadas por personas de genero masculino

Código en R

```
generos <- vector(length = 18)
nombres_generos <- colnames(datos_completos_transformados)[14:31]
names(generos) <- nombres_generos
index <- 1
for (j in 14:31) {
  generos[index] <-
sum(datos_completos_transformados[j]==1&datos_completos_transformados$Valoration=="P"&datos_completos_tra
nsformados$Gender=="M")
  index <- index + 1
}
generos <- sort(generos,decreasing = TRUE)
print(as.data.frame(generos,nm = "Cantidad de valoraciones positivas"))
mejores <- generos
pct <- round(mejores/sum(mejores)*100,digits = 2)
lbls <- names(mejores)
lbls <- paste(lbls, pct)
lbls <- paste(lbls,"%",sep="")
pie(mejores,labels = lbls,main = "Popularidad de Generos para Varones",col=rainbow(length(mejores)))
```

6.4. Calificaciones realizadas por personas de genero femenino

Código en R

```
generos <- vector(length = 18)
nombres_generos <- colnames(datos_completos_transformados)[14:31]
names(generos) <- nombres_generos
index <- 1
for (j in 14:31) {
  generos[index] <-
sum(datos_completos_transformados[j]==1&datos_completos_transformados$Valoration=="P"&datos_completos_tra
nsformados$Gender=="F")
  index <- index + 1
}
generos <- sort(generos,decreasing = TRUE)
print(as.data.frame(generos,nm = "Cantidad de valoraciones positivas"))
mejores <- generos
pct <- round(mejores/sum(mejores)*100,digits = 2)
lbls <- names(mejores)
lbls <- paste(lbls, pct)
lbls <- paste(lbls,"%",sep="")
pie(mejores,labels = lbls,main = "Popularidad de Generos para Mujeres",col=rainbow(length(mejores)))
```

6.5. Determinar los géneros de películas mas populares por estado

Código en R

```
datos_por_estado <- split(datos_completos_transformados,datos_completos_transformados$State)
cat("ESTADO: GENERO MAS POPULAR",sep = "\n")
for (i in 1:54) {
  data_estado <- datos_por_estado[[i]]
  estado <- data_estado$State.Abbreviation[1]
  generos <- vector(length = 18)
  nombres_generos <- colnames(data_estado)[14:31]
  names(generos) <- nombres_generos
  index <- 1
  for (j in 14:31) {
    generos[index] <- sum(data_estado[j]==1&data_estado$Valoration=="P")
    index <- index + 1
  }
  generos <- sort(generos,decreasing = TRUE)
  estado_char <- as.character(estado)
```

```
  mejor_genero <- names(generos[1])
  estado_genero <- paste(estado_char,": ",mejor_genero,sep = "")
  estado_genero <- noquote(estado_genero)
  cat(estado_genero,sep = "\n")
}
```

6.6. Determinar los géneros de películas mas populares por profesión

Código en R

```
cat("Top 3 Generos mas populares por profesion",sep = "\n")
nombres_generos <- colnames(datos_completos_transformados)[14:31]
nombres_profesiones <- unique(datos_completos_transformados$Occupation)
for (i in 1:length(nombres_profesiones)) {
  generos <- vector(length = 18)
  names(generos) <- nombres_generos
  index <- 1
  for (j in 14:31) {
    generos[index] <-
sum(datos_completos_transformados[j]==1&datos_completos_transformados$Valoration=="P"&datos_completos_tra
nsformados$Occupation==nombres_profesiones[i])
    index <- index + 1
  }
  generos <- sort(generos,decreasing = TRUE)
  pct <- round(generos/sum(generos)*100,digits = 2)
  texto <- paste(nombres_profesiones[i],"\n\t1. ",names(generos[1])," - ",pct[1],"%",sep = "")
  texto <- paste(texto,"\n\t2. ",names(generos[2])," - ",pct[2],"%",sep = "")
  texto <- paste(texto,"\n\t3. ",names(generos[3])," - ",pct[3],"%",sep = "")
  cat(texto,sep = "\n")
}
```

6.7. Determinar los géneros de películas mas populares por edad

Código en R

```
cat("Top 3 Generos mas populares por edad",sep = "\n")
generos <- vector(length = 18)
nombres_generos <- colnames(datos_completos_transformados)[14:31]
edades <- unique(datos_completos_transformados$Age)
for (i in 1:length(edades)) {
  names(generos) <- nombres_generos
  index <- 1
  for (j in 14:31) {
    generos[index] <-
sum(datos_completos_transformados[j]==1&datos_completos_transformados$Valoration=="P"&datos_completos_tra
nsformados$Age==edades[i])
    index <- index + 1
  }
  generos <- sort(generos,decreasing = TRUE)
  #print(as.data.frame(generos,nm = edades[i]))
  pct <- round(generos/sum(generos)*100,digits = 2)
  texto <- paste(edades[i],"\n\t1. ",names(generos[1])," - ",pct[1],"%",sep = "")
  texto <- paste(texto,"\n\t2. ",names(generos[2])," - ",pct[2],"%",sep = "")
  texto <- paste(texto,"\n\t3. ",names(generos[3])," - ",pct[3],"%",sep = "")
  cat(texto,sep = "\n")
}
```

6.8. Predecir la tendencia a calificar de manera positiva o negativa de los usuarios

6.8.1. Crear la tabla con la cantidad de películas que vio cada usuario por genero y la tendencia de voto que tiene

```
CREATE TABLE `tbl_tendencia_votos_usuarios_extra` (
  `UserID` bigint(20) DEFAULT NULL,
  `F_Action` bigint(20) DEFAULT NULL,
```

```
  `F_Adventure` bigint(20) DEFAULT NULL,
  `F_Animation` bigint(20) DEFAULT NULL,
  `F_Children` bigint(20) DEFAULT NULL,
  `F_Comedy` bigint(20) DEFAULT NULL,
  `F_Crime` bigint(20) DEFAULT NULL,
  `F_Documentary` bigint(20) DEFAULT NULL,
  `F_Drama` bigint(20) DEFAULT NULL,
  `F_Fantasy` bigint(20) DEFAULT NULL,
  `F_FilmNoir` bigint(20) DEFAULT NULL,
  `F_Horror` bigint(20) DEFAULT NULL,
  `F_Musical` bigint(20) DEFAULT NULL,
  `F_Mystery` bigint(20) DEFAULT NULL,
  `F_Romance` bigint(20) DEFAULT NULL,
  `F_SciFi` bigint(20) DEFAULT NULL,
  `F_Thriller` bigint(20) DEFAULT NULL,
  `F_War` bigint(20) DEFAULT NULL,
  `F_Western` bigint(20) DEFAULT NULL,
  `Tendencia_Voto` char(1) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

6.8.2. Agrupar por cada usuario la cantidad de películas vistas por genero y añadirle la tendencia de voto

```
insert into `tbl_tendencia_votos_usuarios_extra`
select `UserID`, F_Action, F_Adventure, F_Animation, F_Children, F_Comedy, F_Crime,
F_Documentary,
F_Drama, F_Fantasy, F_FilmNoir, F_Horror, F_Musical, F_Mystery, F_Romance, F_SciFi,
F_Thriller, F_War,
F_Western, case when P_Rating >= 4 then 'P' else 'N' end as Tendencia_Voto from (
select `UserID`, sum(`Action`) as F_Action, sum(`Adventure`) as F_Adventure,
sum(`Animation`) as F_Animation, sum(`Children.s`) as F_Children, sum(`Comedy`) as F_Comedy,
sum(`Crime`) as F_Crime, sum(`Documentary`) as F_Documentary, sum(`Drama`) as F_Drama,
sum(`Fantasy`) as F_Fantasy, sum(`Film.Noir`) as F_FilmNoir, sum(`Horror`) as F_Horror,
sum(`Musical`) as F_Musical, sum(`Mystery`) as F_Mystery, sum(`Romance`) as F_Romance,
sum(`Sci.Fi`) as F_SciFi, sum(`Thriller`) as F_Thriller, sum(`War`) as F_War,
sum(`Western`) as F_Western, avg(`Rating`) as P_Rating
from `tbl_calificaciones`
group by `UserID` asc ) as Usuarios_Cantidad_Peliculas;
```

6.8.3. Separar un 20% de los datos para testeo del árbol

```
tendencia_votos_usuarios_extra <-
read.csv("C:/Users/Ale/Desktop/tendencia_votos_usuarios_extra.csv")
cant_columnas <- nrow(tendencia_votos_usuarios_extra)
porcentaje <- (cant_columnas / 100) * 20
testing_extra <- tendencia_votos_usuarios_extra[sample(cant_columnas,porcentaje),]
training_extra <- tendencia_votos_usuarios_extra[ !(tendencia_votos_usuarios_extra$UserID %in%
testing_extra$UserID), ]
write.csv(x = testing_extra,file = "C:/Users/Ale/Desktop/Tendencia/testing_extra.csv",
          quote = TRUE,row.names = FALSE)
write.csv(x = training_extra,file = "C:/Users/Ale/Desktop/Tendencia/training_extra.csv",
          quote = TRUE,row.names = FALSE)
```

6.9. Dividir a los usuarios en clusters según la cantidad de películas calificadas de cada genero y analizar cada uno de ellos

6.9.1. *Generar el primer fichero*

Código en R

```
dct_clase <- read.csv("C:/dct_clase.csv")
dct_clase <- dct_clase[-1:-2]
```

```
dct_clase <- dct_clase[-2:-11]
dct_clase <- dct_clase[-20]
dct_clase <- dct_clase[order(dct_clase$UserID),]
write.csv(dct_clase,file = "C:/Usuarios_Generos.csv",row.names = FALSE,quote = TRUE)
```

6.9.2. Transformación del primer caso

6.9.2.1. Crear la tabla destino

Código SQL

```
CREATE TABLE `tbl_usuarios_generos` (
  `UserID` bigint(20) DEFAULT NULL,
  `Action` bigint(20) DEFAULT NULL,
  `Adventure` bigint(20) DEFAULT NULL,
  `Animation` bigint(20) DEFAULT NULL,
  `Children.s` bigint(20) DEFAULT NULL,
  `Comedy` bigint(20) DEFAULT NULL,
  `Crime` bigint(20) DEFAULT NULL,
  `Documentary` bigint(20) DEFAULT NULL,
  `Drama` bigint(20) DEFAULT NULL,
  `Fantasy` bigint(20) DEFAULT NULL,
  `Film.Noir` bigint(20) DEFAULT NULL,
  `Horror` bigint(20) DEFAULT NULL,
  `Musical` bigint(20) DEFAULT NULL,
  `Mystery` bigint(20) DEFAULT NULL,
  `Romance` bigint(20) DEFAULT NULL,
  `Sci.Fi` bigint(20) DEFAULT NULL,
  `Thriller` bigint(20) DEFAULT NULL,
  `War` bigint(20) DEFAULT NULL,
  `Western` bigint(20) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

6.9.2.2. Crear la tabla con cantidad de películas por genero

Código SQL

```
CREATE TABLE `tbl_usuarios_frecuencia_generos` (
  `UserID` bigint(20) DEFAULT NULL,
  `F_Action` bigint(20) DEFAULT NULL,
  `F_Adventure` bigint(20) DEFAULT NULL,
  `F_Animation` bigint(20) DEFAULT NULL,
  `F_Children` bigint(20) DEFAULT NULL,
  `F_Comedy` bigint(20) DEFAULT NULL,
  `F_Crime` bigint(20) DEFAULT NULL,
  `F_Documentary` bigint(20) DEFAULT NULL,
  `F_Drama` bigint(20) DEFAULT NULL,
  `F_Fantasy` bigint(20) DEFAULT NULL,
  `F_FilmNoir` bigint(20) DEFAULT NULL,
  `F_Horror` bigint(20) DEFAULT NULL,
  `F_Musical` bigint(20) DEFAULT NULL,
  `F_Mystery` bigint(20) DEFAULT NULL,
  `F_Romance` bigint(20) DEFAULT NULL,
  `F_SciFi` bigint(20) DEFAULT NULL,
  `F_Thriller` bigint(20) DEFAULT NULL,
  `F_War` bigint(20) DEFAULT NULL,
  `F_Western` bigint(20) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

6.9.2.3. Agrupar por cada usuario la cantidad de películas vistas por genero

Código SQL

```
insert into `tbl_usuarios_frecuencia_generos`
select `UserID`, sum(`Action`) as F_Action, sum(`Adventure`) as F_Adventure,
sum(`Animation`) as F_Animation, sum(`Children.s`) as F_Children, sum(`Comedy`) as F_Comedy,
sum(`Crime`) as F_Crime, sum(`Documentary`) as F_Documentary, sum(`Drama`) as F_Drama,
sum(`Fantasy`) as F_Fantasy, sum(`Film.Noir`) as F_FilmNoir, sum(`Horror`) as F_Horror,
sum(`Musical`) as F_Musical, sum(`Mystery`) as F_Mystery, sum(`Romance`) as F_Romance,
sum(`Sci.Fi`) as F_SciFi, sum(`Thriller`) as F_Thriller, sum(`War`) as F_War,
sum(`Western`) as F_Western from `tbl_usuarios_generos`
group by `UserID` asc;
```

6.9.3. Transformación del segundo caso

6.9.3.1. Crear la tabla destino

Código SQL

```
CREATE TABLE `tbl_calificaciones` (
  `Rating` bigint(20) DEFAULT NULL,
  `DateRating` datetime DEFAULT NULL,
  `UserID` bigint(20) DEFAULT NULL,
  `Gender` char(1) DEFAULT NULL,
  `Age` varchar(8) DEFAULT NULL,
  `Occupation` varchar(20) DEFAULT NULL,
  `Zip.Code` bigint(20) DEFAULT NULL,
  `Place.Name` varchar(22) DEFAULT NULL,
  `State` varchar(20) DEFAULT NULL,
  `State.Abbreviation` varchar(2) DEFAULT NULL,
  `MovieID` bigint(20) DEFAULT NULL,
  `Movie_Title` varchar(82) DEFAULT NULL,
  `Movie_Year` bigint(20) DEFAULT NULL,
  `Action` bigint(20) DEFAULT NULL,
  `Adventure` bigint(20) DEFAULT NULL,
  `Animation` bigint(20) DEFAULT NULL,
  `Children.s` bigint(20) DEFAULT NULL,
  `Comedy` bigint(20) DEFAULT NULL,
  `Crime` bigint(20) DEFAULT NULL,
  `Documentary` bigint(20) DEFAULT NULL,
  `Drama` bigint(20) DEFAULT NULL,
  `Fantasy` bigint(20) DEFAULT NULL,
  `Film.Noir` bigint(20) DEFAULT NULL,
  `Horror` bigint(20) DEFAULT NULL,
  `Musical` bigint(20) DEFAULT NULL,
  `Mystery` bigint(20) DEFAULT NULL,
  `Romance` bigint(20) DEFAULT NULL,
  `Sci.Fi` bigint(20) DEFAULT NULL,
  `Thriller` bigint(20) DEFAULT NULL,
  `War` bigint(20) DEFAULT NULL,
  `Western` bigint(20) DEFAULT NULL,
  `Valoration` char(1) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

6.9.3.2. Crear la tabla con la cantidad de películas, positivos y negativos por genero

Código SQL

```
CREATE TABLE `tbl_usuarios_frecuencia_generos_y_calificaciones` (
  `UserID` bigint(20) DEFAULT NULL,
  `F_Action_P` bigint(20) DEFAULT NULL,
  `F_Action_N` bigint(20) DEFAULT NULL,
  `F_Action` bigint(20) DEFAULT NULL,
  `F_Adventure_P` bigint(20) DEFAULT NULL,
  `F_Adventure_N` bigint(20) DEFAULT NULL,
  `F_Adventure` bigint(20) DEFAULT NULL,
  `F_Animation_P` bigint(20) DEFAULT NULL,
  `F_Animation_N` bigint(20) DEFAULT NULL,
  `F_Animation` bigint(20) DEFAULT NULL,
  `F_Children_P` bigint(20) DEFAULT NULL,
  `F_Children_N` bigint(20) DEFAULT NULL,
  `F_Children` bigint(20) DEFAULT NULL,
  `F_Comedy_P` bigint(20) DEFAULT NULL,
  `F_Comedy_N` bigint(20) DEFAULT NULL,
  `F_Comedy` bigint(20) DEFAULT NULL,
  `F_Crime_P` bigint(20) DEFAULT NULL,
  `F_Crime_N` bigint(20) DEFAULT NULL,
  `F_Crime` bigint(20) DEFAULT NULL,
  `F_Documentary_P` bigint(20) DEFAULT NULL,
  `F_Documentary_N` bigint(20) DEFAULT NULL,
  `F_Documentary` bigint(20) DEFAULT NULL,
  `F_Drama_P` bigint(20) DEFAULT NULL,
  `F_Drama_N` bigint(20) DEFAULT NULL,
  `F_Drama` bigint(20) DEFAULT NULL,
  `F_Fantasy_P` bigint(20) DEFAULT NULL,
  `F_Fantasy_N` bigint(20) DEFAULT NULL,
  `F_Fantasy` bigint(20) DEFAULT NULL,
  `F_FilmNoir_P` bigint(20) DEFAULT NULL,
  `F_FilmNoir_N` bigint(20) DEFAULT NULL,
  `F_FilmNoir` bigint(20) DEFAULT NULL,
  `F_Horror_P` bigint(20) DEFAULT NULL,
  `F_Horror_N` bigint(20) DEFAULT NULL,
  `F_Horror` bigint(20) DEFAULT NULL,
  `F_Musical_P` bigint(20) DEFAULT NULL,
  `F_Musical_N` bigint(20) DEFAULT NULL,
  `F_Musical` bigint(20) DEFAULT NULL,
  `F_Mystery_P` bigint(20) DEFAULT NULL,
  `F_Mystery_N` bigint(20) DEFAULT NULL,
  `F_Mystery` bigint(20) DEFAULT NULL,
  `F_Romance_P` bigint(20) DEFAULT NULL,
  `F_Romance_N` bigint(20) DEFAULT NULL,
  `F_Romance` bigint(20) DEFAULT NULL,
  `F_SciFi_P` bigint(20) DEFAULT NULL,
  `F_SciFi_N` bigint(20) DEFAULT NULL,
  `F_SciFi` bigint(20) DEFAULT NULL,
  `F_Thriller_P` bigint(20) DEFAULT NULL,
  `F_Thriller_N` bigint(20) DEFAULT NULL,
  `F_Thriller` bigint(20) DEFAULT NULL,
  `F_War_P` bigint(20) DEFAULT NULL,
  `F_War_N` bigint(20) DEFAULT NULL,
  `F_War` bigint(20) DEFAULT NULL,
  `F_Western_P` bigint(20) DEFAULT NULL,
  `F_Western_N` bigint(20) DEFAULT NULL,
  `F_Western` bigint(20) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

6.9.3.3. Agrupar por cada usuario la cantidad de películas, positivos y negativos por genero

Código SQL

```
insert into `tbl_usuarios_frecuencia_generos_y_calificaciones`
```

```
select `UserID`,
count(case when `Action` = 1 and `Valoration` = 'P' then 1 else null end) as F_Action_P,
count(case when `Action` = 1 and `Valoration` = 'N' then 1 else null end) as F_Action_N,
sum(`Action`) as F_Action,
count(case when `Adventure` = 1 and `Valoration` = 'P' then 1 else null end) as F_Adventure_P,
count(case when `Adventure` = 1 and `Valoration` = 'N' then 1 else null end) as F_Adventure_N,
sum(`Adventure`) as F_Adventure,
count(case when `Animation` = 1 and `Valoration` = 'P' then 1 else null end) as F_Animation_P,
count(case when `Animation` = 1 and `Valoration` = 'N' then 1 else null end) as F_Animation_N,
sum(`Animation`) as F_Animation,
count(case when `Children.s` = 1 and `Valoration` = 'P' then 1 else null end) as F_Children_P,
count(case when `Children.s` = 1 and `Valoration` = 'N' then 1 else null end) as F_Children_N,
sum(`Children.s`) as F_Children,
count(case when `Comedy` = 1 and `Valoration` = 'P' then 1 else null end) as F_Comedy_P,
count(case when `Comedy` = 1 and `Valoration` = 'N' then 1 else null end) as F_Comedy_N,
sum(`Comedy`) as F_Comedy,
count(case when `Crime` = 1 and `Valoration` = 'P' then 1 else null end) as F_Crime_P,
count(case when `Crime` = 1 and `Valoration` = 'N' then 1 else null end) as F_Crime_N,
sum(`Crime`) as F_Crime,
count(case when `Documentary` = 1 and `Valoration` = 'P' then 1 else null end) as
F_Documentary_P,
count(case when `Documentary` = 1 and `Valoration` = 'N' then 1 else null end) as
F_Documentary_N,
sum(`Documentary`) as F_Documentary,
count(case when `Drama` = 1 and `Valoration` = 'P' then 1 else null end) as F_Drama_P,
count(case when `Drama` = 1 and `Valoration` = 'N' then 1 else null end) as F_Drama_N,
sum(`Drama`) as F_Drama,
count(case when `Fantasy` = 1 and `Valoration` = 'P' then 1 else null end) as F_Fantasy_P,
count(case when `Fantasy` = 1 and `Valoration` = 'N' then 1 else null end) as F_Fantasy_N,
sum(`Fantasy`) as F_Fantasy,
count(case when `Film.Noir` = 1 and `Valoration` = 'P' then 1 else null end) as F_FilmNoir_P,
count(case when `Film.Noir` = 1 and `Valoration` = 'N' then 1 else null end) as F_FilmNoir_N,
sum(`Film.Noir`) as F_FilmNoir,
count(case when `Horror` = 1 and `Valoration` = 'P' then 1 else null end) as F_Horror_P,
count(case when `Horror` = 1 and `Valoration` = 'N' then 1 else null end) as F_Horror_N,
sum(`Horror`) as F_Horror,
count(case when `Musical` = 1 and `Valoration` = 'P' then 1 else null end) as F_Musical_P,
count(case when `Musical` = 1 and `Valoration` = 'N' then 1 else null end) as F_Musical_N,
sum(`Musical`) as F_Musical,
count(case when `Mystery` = 1 and `Valoration` = 'P' then 1 else null end) as F_Mystery_P,
count(case when `Mystery` = 1 and `Valoration` = 'N' then 1 else null end) as F_Mystery_N,
sum(`Mystery`) as F_Mystery,
count(case when `Romance` = 1 and `Valoration` = 'P' then 1 else null end) as F_Romance_P,
count(case when `Romance` = 1 and `Valoration` = 'N' then 1 else null end) as F_Romance_N,
sum(`Romance`) as F_Romance,
count(case when `Sci.Fi` = 1 and `Valoration` = 'P' then 1 else null end) as F_SciFi_P,
count(case when `Sci.Fi` = 1 and `Valoration` = 'N' then 1 else null end) as F_SciFi_N,
sum(`Sci.Fi`) as F_SciFi,
count(case when `Thriller` = 1 and `Valoration` = 'P' then 1 else null end) as F_Thriller_P,
count(case when `Thriller` = 1 and `Valoration` = 'N' then 1 else null end) as F_Thriller_N,
sum(`Thriller`) as F_Thriller,
count(case when `War` = 1 and `Valoration` = 'P' then 1 else null end) as F_War_P,
count(case when `War` = 1 and `Valoration` = 'N' then 1 else null end) as F_War_N,
sum(`War`) as F_War,
count(case when `Western` = 1 and `Valoration` = 'P' then 1 else null end) as F_Western_P,
count(case when `Western` = 1 and `Valoration` = 'N' then 1 else null end) as F_Western_N,
sum(`Western`) as F_Western
from `tbl_calificaciones`
group by `UserID` asc;
```

6.9.4. División en clusters del primer caso

Código en R

```
frecuencia_por_usuario <- read.csv("C:/frecuencia_por_usuario.csv")
frecuencia_por_usuario <- frecuencia_por_usuario[-1]
```

```
frecuencia_por_usuario <- scale(frecuencia_por_usuario)

kmeans2 <- kmeans(frecuencia_por_usuario,2)
kmeans3 <- kmeans(frecuencia_por_usuario,3)
kmeans4 <- kmeans(frecuencia_por_usuario,4)
kmeans5 <- kmeans(frecuencia_por_usuario,5)
kmeans6 <- kmeans(frecuencia_por_usuario,6)
kmeans7 <- kmeans(frecuencia_por_usuario,7)
kmeans8 <- kmeans(frecuencia_por_usuario,8)

distancias.usuarios <- dist(frecuencia_por_usuario,method = "euclidean")

coef.silueta.kmeans2 <- silhouette(kmeans2$cluster,distancias.usuarios)
coef.silueta.kmeans3 <- silhouette(kmeans3$cluster,distancias.usuarios)
coef.silueta.kmeans4 <- silhouette(kmeans4$cluster,distancias.usuarios)
coef.silueta.kmeans5 <- silhouette(kmeans5$cluster,distancias.usuarios)
coef.silueta.kmeans6 <- silhouette(kmeans6$cluster,distancias.usuarios)
coef.silueta.kmeans7 <- silhouette(kmeans7$cluster,distancias.usuarios)
coef.silueta.kmeans8 <- silhouette(kmeans8$cluster,distancias.usuarios)
```

6.9.5. Guardar clusters del primer caso

Código en R

```
frecuencia_por_usuario$cluster <- kmeans2$cluster

write.csv(x = frecuencia_por_usuario,
          file = "C:/frecuencia_por_usuario_CLUSTERS.csv",
          row.names = FALSE, quote = TRUE)
```

6.9.6. División en clusters del segundo caso

Código en R

```
frecuencia_por_usuario_P_N <- read.csv("C:/frecuencia_por_usuario_positivos_negativos.csv")
frecuencia_por_usuario_P_N <- frecuencia_por_usuario_P_N[-1]

frecuencia_por_usuario_P_N <- scale(frecuencia_por_usuario_P_N)

kmeans_P_N_2 <- kmeans(frecuencia_por_usuario_P_N,2)
kmeans_P_N_3 <- kmeans(frecuencia_por_usuario_P_N,3)
kmeans_P_N_4 <- kmeans(frecuencia_por_usuario_P_N,4)
kmeans_P_N_5 <- kmeans(frecuencia_por_usuario_P_N,5)
kmeans_P_N_6 <- kmeans(frecuencia_por_usuario_P_N,6)
kmeans_P_N_7 <- kmeans(frecuencia_por_usuario_P_N,7)
kmeans_P_N_8 <- kmeans(frecuencia_por_usuario_P_N,8)

distancias.usuarios_P_N <- dist(frecuencia_por_usuario_P_N,method = "euclidean")

coef.silueta.kmeans_P_N_2 <- silhouette(kmeans_P_N_2$cluster,distancias.usuarios_P_N)
coef.silueta.kmeans_P_N_3 <- silhouette(kmeans_P_N_3$cluster,distancias.usuarios_P_N)
coef.silueta.kmeans_P_N_4 <- silhouette(kmeans_P_N_4$cluster,distancias.usuarios_P_N)
coef.silueta.kmeans_P_N_5 <- silhouette(kmeans_P_N_5$cluster,distancias.usuarios_P_N)
coef.silueta.kmeans_P_N_6 <- silhouette(kmeans_P_N_6$cluster,distancias.usuarios_P_N)
coef.silueta.kmeans_P_N_7 <- silhouette(kmeans_P_N_7$cluster,distancias.usuarios_P_N)
coef.silueta.kmeans_P_N_8 <- silhouette(kmeans_P_N_8$cluster,distancias.usuarios_P_N)
```

6.9.7. Guardar clusters del segundo caso

Código en R

```
frecuencia_por_usuario_P_N$cluster <- kmeans_P_N_2$cluster

write.csv(x = frecuencia_por_usuario_P_N,
          file = "C:/frecuencia_por_usuario_P_N_CLUSTERS.csv",
          row.names = FALSE, quote = TRUE)
```

6.9.8. Análisis de los clusters

6.9.8.1. Distribución según el genero de los usuarios

Código en R

```
dct_clase_c1 <- read.csv("C:/dct_clase_c1.csv")

datos_usuarios_c1 <- unique(dct_clase_c1[3:10])

cant_mujeres <- sum(datos_usuarios_c1$Gender=="F")
cant_varones <- sum(datos_usuarios_c1$Gender=="M")

cantidades_por_genero <- c(cant_varones,cant_mujeres)

names(cantidades_por_genero) <- c("M","F")
pct <- round(cantidades_por_genero/sum(cantidades_por_genero)*100,digits = 2)
lbls <- names(cantidades_por_genero)
lbls <- paste(lbls, pct)
lbls <- paste(lbls,"%",sep="")
pie(cantidades_por_genero,labels = lbls,main = "Distribucion de generos de usuarios del
cluster 1",col=rainbow(8))
```

6.9.8.2. Distribución según la edad de los usuarios

Código en R

```
cantidades_por_edad <- vector(length = length(unique(datos_usuarios_c1$Age)))

edades <- c("Under 18","18-24","25-34","35-44","45-49","50-55","56+")

names(cantidades_por_edad) <- edades

for (i in 1:length(edades)) {
  cantidades_por_edad[i] <- sum(datos_usuarios_c1$Age==edades[i])
}

pct <- round(cantidades_por_edad/sum(cantidades_por_edad)*100,digits = 2)
lbls <- names(cantidades_por_edad)
lbls <- paste("\"",lbls,"\" ",pct,sep = "")
lbls <- paste(lbls,"%",sep="")
pie(cantidades_por_edad,labels = lbls,main = "Distribucion de edades de usuarios del cluster
1",col=rainbow(length(cantidades_por_edad)))
```

6.9.8.3. Distribución según la profesión de los usuarios

Código en R

```
cantidades_por_profesion <- vector(length = length(unique(datos_usuarios_c1$Occupation)))

profesiones <- unique(datos_usuarios_c1$Occupation)
```

```
names(cantidades_por_profesion) <- profesiones

for (i in 1:length(profesiones)) {
  cantidades_por_profesion[i] <- sum(datos_usuarios_c1$Occupation==profesiones[i])
}

library(ggplot2)

pct <- round(cantidades_por_profesion/sum(cantidades_por_profesion)*100,digits = 2)
lbls <- names(cantidades_por_profesion)
lbls <- paste("\"",lbls,"\" ",pct,sep = "")
lbls <- paste(lbls,"%",sep="")
porcentajes_por_profesion <- cantidades_por_profesion/sum(cantidades_por_profesion)
grafico <- qplot(x=profesiones, y=porcentajes_por_profesion, geom="bar",
                 stat="identity",position="dodge",
                 fill=I(rainbow(length(porcentajes_por_profesion))))
grafico + theme(axis.text.x=element_text(angle=45, vjust=1, hjust=1, size=20)) +
  geom_text(aes(label = sprintf("%.2f%%", porcentajes_por_profesion * 100)),vjust = -.5)
```

6.9.8.4. Distribución según el estado en donde viven los usuarios

Código en R

```
cantidades_por_estado <- vector(length = length(unique(datos_usuarios_c1$State.Abbreviation)))

estados <- unique(datos_usuarios_c1$State.Abbreviation)

names(cantidades_por_estado) <- estados

for (i in 1:length(estados)) {
  cantidades_por_estado[i] <- sum(datos_usuarios_c1$State.Abbreviation==estados[i])
}

library(ggplot2)

pct <- round(cantidades_por_estado/sum(cantidades_por_estado)*100,digits = 2)
lbls <- names(cantidades_por_estado)
lbls <- paste("\"",lbls,"\" ",pct,sep = "")
lbls <- paste(lbls,"%",sep="")
porcentajes_por_estado <- cantidades_por_estado/sum(cantidades_por_estado)
grafico <- qplot(x=estados, y=porcentajes_por_estado, geom="bar",
                 stat="identity",position="dodge",
                 fill=I(rainbow(length(porcentajes_por_estado))))
grafico + theme(axis.text.x=element_text(angle=45, vjust=1, hjust=1, size= 20)) +
  geom_text(aes(label = sprintf("%.2f%%", porcentajes_por_estado * 100)),vjust = -.5)
```

6.9.8.5. Agrupamiento por regiones

Código en R

```
oeste <- c("WA","OR","ID","CA","NV","MT","WY","UT","CO","AZ","NM","AK","HI")
medio_oeste <- c("ND","SD","NE","KS","MN","IA","MO","MI","WI","IL","IN","KY","OH")
sur <- c("TX","OK","AR","LA","TN","MS","AL","FL","GA","SC","NC","VA","WV","MD","DC","DE")
sureste <- c("PA","NJ","NY","CT","RI","MA","VT","NH","ME")

indices_oeste <- match(oeste,names(cantidades_por_estado))
indices_medio_oeste <- match(medio_oeste,names(cantidades_por_estado))
indices_sur <- match(sur,names(cantidades_por_estado))
indices_sureste <- match(sureste,names(cantidades_por_estado))

indices_oeste <- indices_oeste[!is.na(indices_oeste)]
indices_medio_oeste <- indices_medio_oeste[!is.na(indices_medio_oeste)]
```

```
indices_sur <- indices_sur[!is.na(indices_sur)]
indices_sureste <- indices_sureste[!is.na(indices_sureste)]

cantidad_oeste <- sum(cantidades_por_estado[indices_oeste])
cantidad_medio_oeste <- sum(cantidades_por_estado[indices_medio_oeste])
cantidad_sur <- sum(cantidades_por_estado[indices_sur])
cantidad_sureste <- sum(cantidades_por_estado[indices_sureste])

cat(paste("Oeste: ",cantidad_oeste,"\n","Medio Oeste: ",cantidad_medio_oeste,"\n",
            "Sur: ",cantidad_sur,"\n","Sureste: ",cantidad_sureste,sep = ""))
```

6.9.8.6. Popularidad de los géneros de películas

Código en R

```
generos <- vector(length = 18)
nombres_generos <- colnames(dct_clase_c1)[14:31]
names(generos) <- nombres_generos
for (i in 1:length(generos)) {
  index = i + 13
  generos[i] <- sum(dct_clase_c1[index]==1&dct_clase_c1$Valoration=="P")
}
generos <- sort(generos,decreasing = TRUE)
generos
pct <- round(generos/sum(generos)*100,digits = 2)
lbls <- names(generos)
lbls <- paste(lbls, pct)
lbls <- paste(lbls,"%",sep="")
pie(generos,labels = lbls,main = "Popularidad de Generos para cluster
1",col=rainbow(length(generos)))
```

6.9.8.7. Popularidad de géneros de películas por estado

Código en R

```
datos_por_estado <- split(dct_clase_c1,dct_clase_c1$State)

oeste <- c("WA","OR","ID","CA","NV","MT","WY","UT","CO","AZ","NM","AK","HI")
medio_oeste <- c("ND","SD","NE","KS","MN","IA","MO","MI","WI","IL","IN","KY","OH")
sur <- c("TX","OK","AR","LA","TN","MS","AL","FL","GA","SC","NC","VA","WV","MD","DC","DE")
sureste <- c("PA","NJ","NY","CT","RI","MA","VT","NH","ME")

generos_oeste <- vector(length = 18)
generos_medio_oeste <- vector(length = 18)
generos_sur <- vector(length = 18)
generos_sureste <- vector(length = 18)

nombres_generos <- colnames(dct_clase_c1)[14:31]

names(generos_oeste) <- nombres_generos
names(generos_medio_oeste) <- nombres_generos
names(generos_sur) <- nombres_generos
names(generos_sureste) <- nombres_generos

cat("ESTADO: GENERO MAS POPULAR",sep = "\n")

for (i in 1:length(datos_por_estado)) {
  data_estado <- datos_por_estado[[i]]
  estado <- data_estado$State.Abbreviation[1]
  generos <- vector(length = 18)
  nombres_generos <- colnames(data_estado)[14:31]
  names(generos) <- nombres_generos
  index <- 1
```

```
  for (j in 14:31) {
    generos[index] <- sum(data_estado[j]==1&data_estado$Valoration=="P")
    if (estado %in% oeste) {
      generos_oeste[index] <- generos[index]
    } else if (estado %in% medio_oeste) {
      generos_medio_oeste[index] <- generos[index]
    } else if (estado %in% sur) {
      generos_sur[index] <- generos[index]
    } else if (estado %in% sureste) {
      generos_sureste[index] <- generos[index]
    }
    index <- index + 1
  }
  generos <- sort(generos,decreasing = TRUE)
  estado_char <- as.character(estado)
  mejor_genero <- names(generos[1])
  estado_genero <- paste(estado_char,": ",mejor_genero,sep = "")
  estado_genero <- noquote(estado_genero)
  cat(estado_genero,sep = "\n")
}

generos_oeste <- sort(generos_oeste,decreasing = TRUE)
generos_medio_oeste <- sort(generos_medio_oeste,decreasing = TRUE)
generos_sur <- sort(generos_sur,decreasing = TRUE)
generos_sureste <- sort(generos_sureste,decreasing = TRUE)

print(generos_oeste)
print(generos_medio_oeste)
print(generos_sur)
print(generos_sureste)

valores <- generos_oeste

pct <- round(valores/sum(valores)*100)
lbls <- names(valores)
lbls <- paste(lbls, pct)
lbls <- paste(lbls,"%",sep="")
pie(valores,labels = lbls,main = "Oeste",col=rainbow(length(valores)))

valores <- generos_medio_oeste

pct <- round(valores/sum(valores)*100)
lbls <- names(valores)
lbls <- paste(lbls, pct)
lbls <- paste(lbls,"%",sep="")
pie(valores,labels = lbls,main = "Medio Oeste",col=rainbow(length(valores)))

valores <- generos_sur

pct <- round(valores/sum(valores)*100)
lbls <- names(valores)
lbls <- paste(lbls, pct)
lbls <- paste(lbls,"%",sep="")
pie(valores,labels = lbls,main = "Sur",col=rainbow(length(valores)))

valores <- generos_sureste

pct <- round(valores/sum(valores)*100)
lbls <- names(valores)
lbls <- paste(lbls, pct)
lbls <- paste(lbls,"%",sep="")
pie(valores,labels = lbls,main = "Sureste",col=rainbow(length(valores)))
```