Sistemas de Recomendación: aplicación a MovieLens

por

Mª Ángeles Meca Espinosa

Tesis presentada en conformidad con los requisitos del
Máster en economía, finanzas y computación.

Universidad de Huelva
Universidad Internacional de Andalucía
Septiembre de 2018

Septiembre 2018

Sistemas de Recomendación: aplicación a MovieLens

Mª Ángeles Meca Espinosa
Máster en Economía, Finanzas y Computación
José Manuel Bravo Caro
Universidad de Huelva y Universidad Internacional de Andalucía
2018

Abstact

The aim of the current study is the realization of a collaborative filtering (CF) recommender system applied to the films' world. In order to achieve this goal we have to forecast the ratings that users give to films that they have not seen, so we could recommend them the movies in which we have predicted ratings above a set threshold. All of this will be implemented in Python language, although R and Stata will also be used, and we will use a several machine learning techniques, both supervised and unsupervised, with the target of examining the ratings of the users who are present in the data because this work can help us to predict the ratings of other users who are similar to them. The final purpose is helping users to find movies that match their and thus improve the user experience.

Key words: Recommender System, Collaborative Filtering, KNN, SVD, Slope One, Principal Component Analysis, Hierarchical Clustering, Dendogram, Ordered multinomial models, Memory-based algorithms, Model-based algorithms, Regression

Resumen

El objetivo del presente trabajo radica en la realización de un Sistema de Recomendación, de filtrado colaborativo, aplicado al mundo de las películas. Dicho sistema de recomendación consiste en predecir el rating que darían usuarios a películas que no han visto, para recomendarles aquellas en las que se predicen ratings superiores a cierto umbral. Para ello, haciendo uso principalmente del lenguaje de programación Python, pero también con la ayuda de R y Stata, se utilizarán diversas técnicas de aprendizaje, tanto supervisado como no supervisado, con el objetivo de examinar de manera minuciosa los ratings de los usuarios que conforman los datos y que pueden ayudar a predecir ratings de otros usuarios "similares" a ellos. Con todo ello se pretende ayudar a los usuarios a encontrar películas afines a ellos y mejorar así la experiencia de usuario.

Palabras clave: Sistema de Recomendación, Filtrado Colaborativo, KNN, SVD, Slope One, Análisis de Componentes principales, Clustering jerárquico, Dendograma, Modelo multinomial ordenado, Algoritmos basados en memoria, Algoritmos basados en modelos, Regresión

Agradecimientos

En este apartado me gustaría expresar a Dr. José Manuel Bravo Caro mi agradecimiento por su predisposición y ayuda en la realización del presente estudio. Así mismo, a mis padres y hermanos por el apoyo mostrado desde la distancia. A mis abuelos que también me han ayudado en todo lo posible y, por supuesto, al más pequeño de la casa y a sus padres, por el apoyo recibido y por la alegría que me ha dado verlo crecer en sus llamadas diarias.

Índice

1. Sistemas de recomendación

1.1. Introducción.....8

1.2. Objetivos derivados de su uso.....9

1.3. Técnicas de recomendación.....10

1.3.1. CF basado en usuarios.....10

1.3.2. CF basado en items.....10

1.3.3. Reducción de dimensionalidad.....11

1.4. Algortimos.....11

1.4.1. K-vecinos más cercanos basado en usuarios.....12

1.4.2. K-vecinos más cercanos basado en items.....13

1.4.3. Concepto de similitud.....14

1.4.4. Slope One.....15

1.4.5. Descomposición en valores singulares 15

1.5. Evaluación.....17

2. Aplicación práctica

2.1. Formulación del problema.....18

2.2. Datos.....20

2.3. Análisis de los datos.....21

2.3.1. Análisis descriptivo.....21

2.3.2. Modelo multinomial ordenado.....26

2.3.3. Análisis de componentes principales.....31

2.3.4. Clustering.....35

2.4. Predicción.....40

2.4.1. Librería….....41

2.4.2. Escenario base….....42

2.4.3. Propuesta de mejora.....46

2.5. Resultados......48

3. Nueva propuesta.....51

4. Conclusiones.....54

5. Referencias bibliográficas.....55

Lista de tablas

Tabla 1. Modelo multinomial ordenado. Primera aproximación. Fuente: elaboración propia **(p.27)**

Tabla 2. Modelo multinomial ordenado. Efectos marginales media global del resto de valoraciones. Fuente: elaboración propia **(p.28)**

Tabla 3. Modelo multinomial ordenado. Efectos marginales: edad. Fuente: elaboración propia **(p.28)**

Tabla 4. Modelo multinomial ordenado. Efectos marginales: valoraciones del usuario. Fuente: elaboración propia **(p.29)**

Tabla 5. Modelo multinomial ordenado. Efectos marginales: grado de diferencia. Fuente: elaboración propia **(p.29)**

Tabla 6. Modelo multinomial ordenado. Efectos marginales: género. Fuente: elaboración propia **(p.30)**

Tabla 7. Modelo multinomial ordenado. Desequilibrio de clases. Fuente: elaboración propia. **(p.30)**

Tabla 8. Modelo multinomial ordenado. Tabla de errores y aciertos. Fuente: elaboración propia **(p.30)**

Tabla 9. Media de cada atributo por Cluster. Fuente: Elaboración propia **(p.39)**

Tabla 10. Estructura datos de entrada algoritmo. Fuente: Elaboración propia **(p.42)**

Tabla 11. Resultados Escenario base para KNN-item y KNN-user. Fuente: Elaboración propia **(p.43)**

Tabla 12. Resultados Escenario base para SlopeOne. Fuente: Elaboración propia **(p.44)**

Tabla 13. Resultados Escenario base SVD. Fuente: Elaboración propia **(p.44)**

Tabla 14. Resultados Cluster 1 para KNN-item y KNN-user. Fuente: Elaboración propia **(p.46)**

Tabla 15. Resultados Cluster 2 para KNN-item y KNN-user. Fuente: Elaboración propia **(p.46)**

Tabla 16. Resultados Propuesta de mejora para SlopeOne. Fuente: Elaboración propia **(p.47)**

Tabla 17. Resultados Propuesta de mejora para SVD. Fuente: Elaboración propia **(p.47)**

Tabla 18. Resultados finales Propuesta de mejora. Fuente: Elaboración propia **(p.47)**

Tabla 19. Comparación Escenario base VS Propuesta de mejora. Fuente: Elaboración propia **(p.48)**

Tabla 20. Estructura predicciones. Fuente: Elaboración propia **(p.51)**

Tabla 21. Propuesta de mejora 2: coeficientes. Fuente: Elaboración propia **(p.52)**

Tabla 22. Propuesta de mejora 3: coeficientes. Fuente: Elaboración propia **(p.53)**

Tabla 23. Comparación escenarios brutos. Fuente: Elaboración propia **(p.53)**

Tabla 24. Comparación escenarios redondeados. Fuente: Elaboración propia **(p.53)**

Lista de gráficos

Gráfico 1. Evolución del uso de TIC por las personas de 16 a 74 años. Fuente: INE. Notas de prensa **(p. 8)**

Gráfico 2. Top 5 Digital Signage Trends for 2018. Fuente: Samsung Display **(p.19)**

Gráfico 3. Porcentaje de películas por rating. Fuente: Elaboración propia **(p. 22)**

Gráfico 4. Valoración media en función del número de películas vistas. Fuente: elaboración propia**(p. 22)**

Gráfico 5. Valoración media en función del número de películas valoradas por grupos de edad. Fuente: elaboración propia **(p. 23)**

Gráfico 6. Valoración media en función del número de películas valoradas por grupos de edad y género. Fuente: elaboración propia **(p. 23)**

Gráfico 7. Valoración media en función del número de valoraciones por película. Fuente: elaboración propia **(p. 24)**

Gráfico 8. Gráfico de sedimentación. Fuente: Elaboración propia **(p. 33)**

Gráfico 9. 3D Biplot PCA. Fuente: Elaboración propia **(p. 33)**

Gráfico 10. 3D Biplot PCA. Fuente: Elaboración propia **(p. 34)**

Gráfico 11. 3D Biplot PCA. Fuente: Elaboración propia **(p. 35)**

Gráfico 12. Hierarchical clustering: Dendograma. Fuente: Elaboración propia **(p. 38)**

Gráfico 13. Hierarchical clustering: gráfico en función de los 3 componentes principales. Fuente: Elaboración propia **(p. 39)**

Gráfico 14. Procedimiento Escenario base. Fuente: Elaboración propia **(P.40)**

Gráfico 15. Procedimiento Propuesta de mejora .Fuente: Elaboración propia **(p. 41)**

Gráfico 16. MAE para distintos valores de k en KNN-item y KNN-user. Fuente: Elaboración propia. **(p. 42)**

Gráfico 17. MAE en función del número de factores considerados .Fuente: Elaboración propia. **(p. 44)**

Gráfico 18. Item-item vs. User-user at Selected Neighborhood Sizes (at x=0.8).Fuente: Collaborative Filtering Recommendation Algorithms **(p.45)**

Gráfico 19. Propuesta de mejora VS Escenario Base .Fuente: Elaboración propia **(p. 49)**

Gráfico 20. Propuesta de mejora con redondeo VS Escenario base con redondeo. Fuente: Elaboración propia **(p. 49)**

Gráfico 21. Distribución de los errores en estrellas Propuesta de mejora VS Escenario base. Fuente: Elaboración propia **(p. 50)**

Gráfico 22. Comparación Escenario base VS Propuestas de mejora **(p. 54)**

1. Sistemas de recomendación

1.1. Introducción

El crecimiento explosivo, tanto de la información digital disponible en internet como de los usuarios que acceden a la red ha generado un desafío derivado de una sobrecarga de información en la que cada vez se requiere más tiempo para poder distinguir aquella información que realmente se adapta a las preferencias de cada uno de los usuarios de aquella que no es de interés. En 2017, El Instituto Nacional de Estadística establece que el 80% de los usuarios de Internet, con respecto al total nacional, entre 16 y 74 años son usuarios frecuentes de las TIC, además un 85 % de los mismos ha utilizado internet durante los últimos 3 meses y un 40% ha realizado compras por internet durante el último trimestre. Todo ello se puede observar en el siguiente gráfico, donde cabe destacar que todos los patrones mostrados presentan una tendencia alcista.

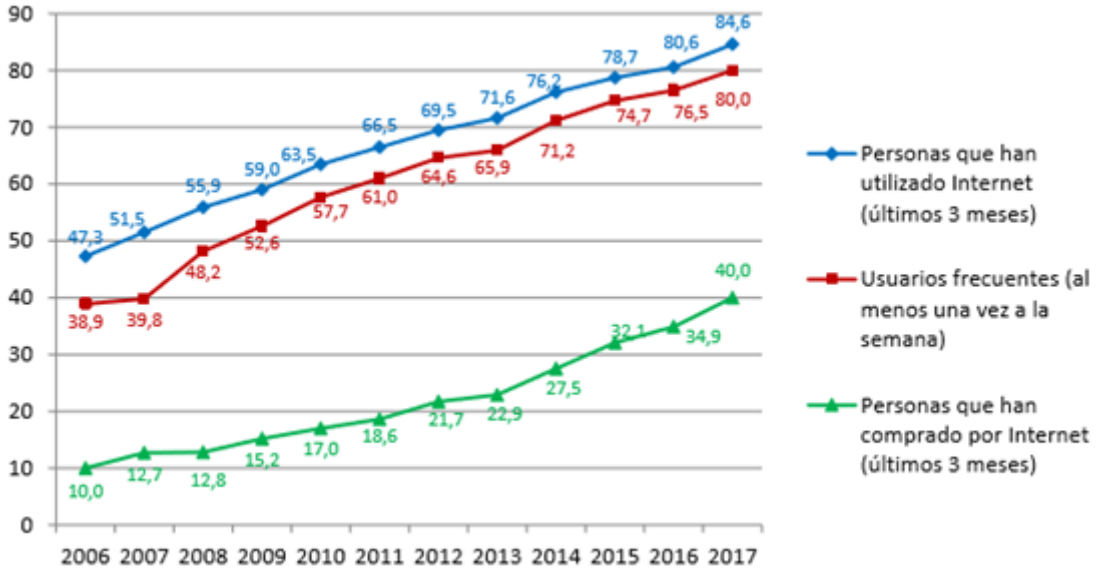

Gráfico 1. Evolución del uso de TIC por las personas de 16 a 74 años. Fuente: INE. Notas de prensa

Con lo cual parece necesaria una solución que permita ahorrar tiempo y adaptar a cada usuario la información que realmente interesa, en especial si se trata de compras, películas, música, noticias… para poder combatir dicha sobrecarga de información. Esta solución viene aportada por los sistemas de recomendación que Ricci, Rokach, Shapira & Kantor (2011) definen como herramientas y técnicas de software que adaptan el contenido disponible a los intereses de cada uno de los usuarios, es decir, filtran el total de información disponible prediciendo las preferencias de cada perfil de usuario.

1.2. Objetivos derivados de su uso

Ricci, Rokach, Shapira & Kantor (2011) afirman que algunas de las funciones que persigue la implementación de un sistema de recomendación son las siguientes:

• Aumento de conversiones: en principio, un buen sistema de recomendación permite materializar la conversión de un conjunto adicional de items que no se habría materializado en caso de no haber implementado dicho sistema. Esto se debe a la capacidad de adaptar las recomendaciones a los deseos o necesidades de los usuarios.

• Aumento de la diversidad de los items[1] seleccionados: permiten a los usuarios seleccionar items difíciles de encontrar sin una recomendación precisa. Por tanto, pueden extender el interés del catálogo disponible al conjunto de items disponibles en lugar de centrarse solo en los más populares.

• Incremento de la satisfacción de usuario: un sistema de recomendación bien diseñado permitirá a los usuarios encontrar las recomendaciones interesantes, relevantes y precisas.

• Aumento de la fidelidad de los usuarios: los sistemas de recomendación deben reconocer a los usuarios-clientes y tratarlos como a visitantes valiosos. Esa valiosa característica surge como consecuencia del cálculo de recomendaciones en base a interacciones previas, como por ejemplo, calificaciones de items por los que han mostrado interés anteriormente. Cuanto más interactúa el usuario con el sitio, mayor es la información de la que se dispone mejorando la representación de las preferencias de usuario.

• Detección del patrón de intereses de usuarios: permiten el conocimiento de las preferencias de los usuarios, ya sea recopiladas explícitamente o predichas por el sistema. El proveedor del servicio podrá utilizar esa información para objetivos diversos, como por ejemplo, recomendar una película o un viaje específico a nuevos sectores de clientes, emitir un mensaje promocional derivado del análisis de datos recopilados por el sistema de recomendación…

[1] Se hará referencia al término “Item” para designar aquello que el sistema referencia a los usuarios.

1.3. Técnicas de recomendación

Este estudio centra su atención en un sistema de recomendación de filtrado colaborativo (CF). Ekstrand, Riedl & Konstan (2011) afirman que: "El CF es un algoritmo de recomendación popular que basa sus predicciones y recomendaciones en las evaluaciones o el comportamiento de otros usuarios del sistema" (p.88). La suposición que radica en dicha tipología de algoritmos, es que la predicción de un usuario concreto puede ser construida en base a la agregación de preferencias del resto de usuarios, es decir, asumen que si un grupo de usuarios A ha mostrado preferencias similares a un usuario B, entonces es probable que el usuario B tenga interés en aquellos items que gustan al grupo A y que desconoce. Hay varios métodos distintos de CF, los dos más "tradicionales" son el filtrado colaborativo basado en usuarios y el filtrado colaborativo basado en elementos. Posteriormente se han adaptado otros más novedosos como la reducción de dimensionalidad aplicada al CF.

1.3.1. FC basado en usuarios

El filtrado colaborativo basado en usuarios opera de la siguiente forma: dado un usuario concreto, si se encuentran otros usuarios cuyas calificaciones o comportamientos anteriores son similares[2] a dicho usuario, entonces podrán ser tenidas en cuenta las calificaciones de dicho grupo en items que el usuario inicial no ha visto para predecir sus recomendaciones. Todo ello ha de ser ejecutado teniendo en cuenta el grado de similitud o acuerdo entre cada usuario del grupo y el usuario al que se están prediciendo gustos.

1.3.2. FC basado en items

El filtrado colaborativo basado en items se basa en la similitud entre el rating o calificación asignada a los items, en lugar de usar el comportamiento de ratings entre usuarios como en el caso anterior. Por tanto, según lo anterior, si dos items tienen el mismo patrón de ratings, entonces serán similares y bajo la hipótesis de que los usuarios tengan preferencias similares por items similares, se recomendarán aquellos items similares a los que el usuario haya puntuado con calificaciones altas.

[2] El concepto de similitud será estudiado posteriormente.

Ambos métodos necesitan una definición precisa de "similitud" antes de proceder a su aplicación. Dicha definición se realizará en el apartado 1.4.3.

1.3.3. Reducción de dimensionalidad

Si se centra la atención en los dos métodos presentados anteriormente, FC basado en usuarios y FC basado en items, se podría pensar en un item como un vector u-dimensional de calificaciones de los usuarios para dicho item, con algunos valores nulos (usuarios que no han calificado dicho item). De la misma forma se podría ver un usuario como un vector i-dimensional de calificaciones de dicho usuario a los distintos ratings, pudiendo incluir también valores nulos (item que no ha valorado). Bajo la idea de que, en general, podrán distinguirse grupos de preferencias similares tanto de usuarios como de items, la pregunta que cabe realizar, tal y como plantean Ekstrand, Riedl & Konstan (2011), es si se podría encontrar un menor número de dimensiones k, que permita representar elementos y usuarios por vectores k-dimensionales.

Siguiendo con el planteamiento anterior, los modelos de factorización matricial buscan un espacio latente que explique las características tanto de items como de usuarios, y que se deduce del propio comportamiento, es decir de los ratings, de los usuarios. Esta es la idea que reside detrás del algoritmo SVD que se estudiará posteriormente.

1.4. Algoritmos

A grandes rasgos, siguiendo el planteamiento del punto anterior, se pueden agrupar los métodos de filtrado colaborativo en dos bloques bien diferenciados. En un primer bloque se encontrarían los métodos basados en memoria, definiéndose como aquellos en los que las calificaciones o ratings disponibles en el sistema se utilizan directamente para realizar las predicciones de ratings desconocidos. En un segundo bloque se hallarían los métodos basados en modelos que, a diferencia de los primeros, utilizan los ratings disponibles para aprender un modelo predictivo. La idea general de éstos últimos es modelar las interacciones usuario-item con los factores que representan características latentes de usuarios y elementos del sistema para posteriormente entrenar al modelo con datos disponibles y predecir así ratings de usuarios para nuevos items[3]. Según señalan Ricci, Rokach, Shapira & Kantor (2011) como ventaja de los métodos

[3] Con nuevos items se hace referencia a items que son previamente desconocidos o no valorados por el usuario.

basados en memoria con respecto a los métodos basados en modelos, se haya la mayor facilidad para extraer conclusiones, es decir, para poder estudiar el "por qué" se le recomienda un item concreto a un usuario específico, cosa que no tiene por qué ocurrir con los métodos basados en modelos ya que se caracterizan por un proceso de recomendación algo más "abstracto" o difícil de razonar. Los métodos basados en memoria también suelen ser intuitivos y fáciles de implementar y además, no requieren fases de entrenamiento costosas, como si ocurre con los métodos basados en modelos. Así mismo, tienen atribuida una mayor estabilidad, es decir, se ven poco afectados por la constante edición de usuarios, elementos y ratings pudiendo realizar recomendaciones a nuevos usuarios sin la necesidad de volver a entrenar el sistema. No obstante, es importante resaltar que lo anterior no implica que los primeros métodos sean mejores que los segundos ni al contrario.

Este estudio centrará la atención en cuatro algoritmos, concretamente tres de ellos pertenecen al bloque de métodos basados en memoria que son los siguientes: K-vecinos más cercanos basado en usuarios (KNN-users), K-vecinos más cercanos basado en items (KNN- items) y Slope One. Por el contrario, el último algoritmo se engloba dentro de los métodos basados en modelos: Descomposición en valores singulares (SVD). A continuación se procede a un estudio más detallado de dichos algoritmos.

1.4.1. K-vecinos más cercanos basado en usuarios

En general, los sistemas de recomendación basados en k-vecinos más cercanos materializan el principio del "Word-of-mouth", es decir, aquellas situaciones en las que se suele confiar en la opinión de personas con ideales o preferencias cercanas a las propias para evaluar el valor de un items. El método de recomendación k-vecinos más cercanos basados en usuarios, predice el rating que un usuario concreto daría a un item (rui), usando las calificaciones dadas a dicho item (i) por usuarios similares al usuario en cuestión (u). Los vecinos más cercanos denotados por $N_{i(u)}^{k}$, por tanto serán los k usuarios con la similitud más alta al usuario. No obstante hay que tener en cuenta que sólo podrán utilizarse en la predicción de rui, los usuarios que hayan valorado el item i. La forma más básica en la que se podría materializar todo lo anteriormente descrito en un algoritmo sería mediante la siguiente fórmula:

$$\widehat{r_{ui}} = \frac{1}{\left|N_{i(u)}^{k}\right|} \sum_{v \epsilon N_{i(u)}^{k}} r_{vi}$$

No obstante, dicha fórmula, que presenta simplemente una media simple de los ratings de los k usuarios definidos como más cercanos, tiene como desventaja el no considerar que no todos los vecinos son igual de similares, es decir, a la hora de realizar predicciones, elegido el valor del parámetro k óptimo, no todos los vecinos han de aportar un mismo peso en la predicción, sino que aquellos con los que se tenga un patrón de comportamiento más similar deben tenerse más en cuenta que otros que, también lo comparten pero en menor grado. Por tanto, se podría transformar la fórmula anterior en una media ponderada por el grado de similitud entre usuarios (sim(u,v)):

$$\widehat{r_{ui}} = \frac{\sum_{v \epsilon N_{i(u)}^{k}} sim(u,v) * r_{vi}}{\sum_{v \epsilon N_{i(u)}^{k}} sim(u,v)}$$

Partiendo de la anterior base, se pueden realizar pequeñas modificaciones, como la que se presenta a continuación y que se utilizará en el análisis del sistema de recomendación:

$$\widehat{r_{ui}} = \mu_u + \frac{\sum_{v \epsilon N_{i(u)}^{k}} sim(u,v) * (r_{vi} - \mu_v)}{\sum_{v \epsilon N_{i(u)}^{k}} sim(u,v)}$$

La idea prevalece, pero ahora se parte del rating medio del usuario utilizando la desviación del rating asignado a dicha película con respecto al rating medio para cada uno de los vecinos que hayan sido definidos, con objeto de dinamizar así la media en función del grado de similitud de los usuarios y de dichas desviaciones.

1.4.2. K-vecinos más cercanos basado en item

Este método discrepa del anterior en que si bien KNN-user se basa en "usuarios similares" para predecir una calificación, ahora se va a realizar un giro en dirección a la idea de "elementos similares". Supongamos que se quiere determinar si una película C es afín a mis preferencias, si observo los usuarios que la han valorado y detecto que dichos usuarios le han otorgado calificaciones similares que a las películas A y B, las cuales tienen una alta valoración por mi parte, concluiría que, en principio, la película C es afín a mis gustos. Por tanto, formalizando la idea, ahora los vecinos más cercanos se

denotan por $N_{u(i)}^{k}$, y serán los k items valorados por el usuario u con la similitud más alta al item a predecir, i.

$$\widehat{r_{ui}} = \frac{\sum_{j \epsilon N_{u(i)}^{k}} sim(i,j) * r_{uj}}{\sum_{j \epsilon N_{u(i)}^{k}} sim(i,j)}$$

Del mismo modo en el KNN-user, partiendo de la idea anterior se obtiene la siguiente fórmula que será usada para el estudio real de los datos:

$$\widehat{r_{ui}} = \mu_i + \frac{\sum_{j \epsilon N_{u(i)}^{k}} sim(i,j) * (r_{uj} - \mu_j)}{\sum_{j \epsilon N_{u(i)}^{k}} sim(i,j)}$$

La base se establece en el rating medio del item a valorar y se corregirá en función de la similitud de los items vecinos y la desviación del rating del usuario u para cada vecino con respecto al rating medio del item en cuestión.

1.4.3. Concepto de similitud.

La similitud entre usuarios para el punto anterior va a ser definida de la siguiente manera:

$$\hat{\rho}_{uv} = \frac{\sum_{i \epsilon I_{uv}} (r_{ui} - b_{ui}) * (r_{vi} - b_{vi})}{\sqrt{\sum_{i \epsilon I_{uv}} (r_{ui} - b_{ui})^2} * \sqrt{\sum_{i \epsilon I_{uv}} (r_{vi} - b_{vi})^2}}$$

Donde I_{uv} hace referencia al conjunto de items que han sido valorados por dos usuarios, u y v. Por tanto, se elige la correlación de Pearson para cada par de usuarios, pero en vez de centrar en base a la media se hará entorno a la desviación con respecto a la misma, es decir, en torno a b.

Del mismo modo se procede con la similitud entre items:

$$\hat{\rho}_{ij} = \frac{\sum_{u \epsilon U_{ij}} (r_{ui} - b_{ui}) * (r_{uj} - b_{uj})}{\sqrt{\sum_{u \epsilon U_{ij}} (r_{ui} - b_{ui})^2} * \sqrt{\sum_{u \epsilon U_{ij}} (r_{uj} - b_{uj})^2}}$$

Donde Uij hace referencia al conjunto de usuarios que valoran ambos items, i y j.

1.4.4. Slope One

Slope One se enmarca dentro de las técnicas de CF basados en items, y son muchos los que refieren a él como el predictor más simple dentro de esta tipología. Sus introductores Lemire & Maclachlan (2005), afirman que el nombre "Pendiente uno" se debe a que sus predictores son de la forma f(x) = b + x, donde b es una constante y x una variable que representa los valores de calificación. Con la idea presente de que la media es uno de los predictores más básicos y efectivos, se comienza partiendo de dicho predictor e intentando mejorarlo. Para ello se profundiza en la siguiente idea, ¿Cuánto más es valorada una película con respecto a otra? Para comenzar a dar respuesta a dicha pregunta, como indican sus introductores Lemire & Maclachlan (2005), se considerará la desviación media del item i con respecto al item j de la siguiente manera:

$$dev(i,j) = \frac{1}{|U_{ij}|} \sum_{u \epsilon U_{ij}} (r_{ui} - r_{uj})$$

Donde i, j son items con ratings r_{ui} y r_{uj} respectivamente para las valoraciones de ciertos usuarios ($u \epsilon U_{ij}$). Por tanto, solo se considerarán usuarios que hayan valorado ambos items: j e i.

Para completar dicha idea, supongamos que existen varias opciones de pares de items con calificaciones que pueden usarse en la predicción, es decir los usuarios que han votado el item i, también han votado el z y el y. Por tanto para predecir $\widehat{r_{ui}}$ se utilizarían tanto los pares (i,z) como los pares (i,y). Entonces:

$$\widehat{r_{ui}} = \mu_u + \frac{1}{|R_{i(u)}|} \sum_{j \epsilon R_{i(u)}} dev(i,j)$$

Donde Ri(u) hace referencia al conjunto de items relevantes, es decir, el conjunto de items j valorados por u que tienen al menos un usuario común con el item i. Finalmente, se puede observar como la predicción no es más, que la media de valoraciones del item corregida por "Cuánto más es valorada dicha película que el resto" .

1.4.5. Descomposición en valores singulares (SVD)

SVD trata de encontrar factores que se deducen directamente de los ratings realizados y que caracterizan tanto usuarios como items. Por tanto, como indican Ricci, Rokach,

Shapira & Kantor (2011), estos modelos conocidos también como modelos de factorización matricial, "asignan usuarios y elementos a un espacio factorial latente, de dimensión f, de forma que las interacciones usuario- item se modelan como productos internos en ese espacio" (p.151). Cabe destacar que, como también explican Ricci, Rokach, Shapira & Kantor (2011), los factores, por ejemplo, asociados a las películas en sus extremos pueden ser totalmente interpretables y deducibles, tales como género, o completamente no interpretables. De esta forma, a cada usuario u se asocia a un vector $p_u \in \mathbb{R}^f$ que representa los factores de usuario y cada item i está asociado a un vector $q_i \in \mathbb{R}^f$ que representa los factores de items. Por tanto, para un usuario dado, p_u muestra el interés de dicho usuario en cada uno de los factores. El producto escalar, $q_i^T p_u$ contempla la interacción entre el usuario u y el item i. Sabiendo lo anterior, bajo este método la predicción, de $\widehat{r_{ui}}$ se realiza de la siguiente manera:

$$\widehat{r_{ui}} = \mu + b_i + b_u + {q_i}^T p_u$$

Donde bi hace referencia al sesgo o desviación del item i con respeto a la media, y b_u al sesgo o desviación del usuario u con respecto a su media. Por tanto, como indican, de nuevo, Ricci, Rokach, Shapira & Kantor (2011) el problema a resolver consiste en la minimización del error cuadrático regularizado:

$$\min_{b,q,p} \sum_{r_{ui} \epsilon Rtrain} (r_{ui} - \widehat{r_{ui}}) + \lambda \left(b_i^2 + b_u^2 + \|q_i\|^2 + \|p_u\|^2\right)$$

Donde λ introduce cierto grado de "penalización" a los coeficientes desconocidos. Por tanto se da un trade-off entre dos objetivos, en principio contrapuestos: bondad de ajuste y complejidad del modelo. La complejidad de dicha minimización radica en la no garantía de su convexidad[4]. El problema se resuelve mediante un método iterativo, concretamente mediante el método de descenso del gradiente estocástico[5]. Dicho algoritmo recorre todos los ratings en el conjunto de entrenamiento, realiza una predicción para cada uno de ellos y asocia un determinado error. Para cada caso de entrenamiento se modificarán los parámetros en la dirección contraria al gradiente, obteniendo:

[4] Esto implica que no se garantiza que el óptimo encontrado sea global, como si ocurre cuando el problema es convexo.

[5] Recibe la connotación de estocástico dado que las muestras se selecciona aleatoriamente en lugar de como un único grupo, con objeto de combatir el costo de computación del método del gradiente descendente en muestras muy extensas.

$$b_u \leftarrow b_u + \gamma(e_{ui} - \lambda b_u)$$

$$b_i \leftarrow b_i + \gamma(e_{ui} - \lambda b_i)$$

$$p_u \leftarrow p_u + \gamma(e_{ui} \cdot q_i - \lambda p_u)$$

$$q_i \leftarrow q_i + \gamma(e_{ui} \cdot p_u - \lambda q_i)$$

donde γ hace referencia al tamaño del paso.

Con el tamaño de paso adecuado, SGD converge, casi seguramente, al mínimo global si la función es convexa o pseudoconvexa, con la posibilidad que converger a un óptimo local si la función no es convexa.

1.5. Evaluación

Se define previamente la siguiente notación: U hará referencia al conjunto de usuarios, I al conjunto de items, R denota al conjunto de ratings recogidos y S al rango de ratings posible. Por tanto, el objetivo al que se enfrenta el presente trabajo es predecir, obviamente con el menor margen de error, el rating que un usuario concreto daría a un item concreto y con el que no ha tenido contacto, con objeto de poder recomendar a dicho usuario aquellos items donde se han predicho los ratings más altos[6]. En aquellos escenarios, similares al actual, basados en ratings que se encuentran disponibles, como indican Ricci, Rokach, Shapira & Kantor (2011) "el objetivo consiste en desarrollar una tarea de regresión o una clasificación con el fin último de aprender una función f : U x I → S que prediga los ratings f(u,i) de un usuario u a un item i"(p. 108). El término adoptado del inglés "Accuracy" se usa comúnmente para evaluar el rendimiento de un sistema de recomendación. Cualquier método de machine learning, tiene como primer paso la división del set de datos en dos partes $R_{training}$ y $R_{test,}$ usando posteriormente $R_{training}$ para aprender de los datos, es decir, para aprender la función f y R_{test} para evaluar la calidad de dicho aprendizaje. Esas, serán por tanto las bases del posterior desarrollo práctico. Se suelen usar dos medidas de "Accuracy" típicas que reflejan la calidad de las predicciones. La primera de ellas es Mean Absolute Error (MAE) que no

[6] Fijando, por ejemplo, un umbral en las predicciones, y recomendando aquellos items que lo sobrepasen

es más que la media de los errores cometidos en el conjunto R_{test}, como se indica a continuación.

$$MAE(f) = \frac{1}{R_{test}} \sum_{r_{ui} \epsilon R_{test}} |r_{ui} - f(u,i)|$$

La segunda, Root Mean Square Error (RMSE), hace referencia a la raíz cuadrada de la primera. Se presenta RMSE de la siguiente manera:

$$RMSE(f) = \sqrt{\frac{1}{R_{test}} \sum_{r_{ui} \epsilon R_{test}} \left(r_{ui} - f(u,i)\right)^2}$$

Por tanto, dichas medidas de evaluación permiten discriminar entre varios modelos, para poder elegir aquel que obtenga un menor error. Se considerará suficiente con presentar MAE, puesto que las dos medidas de manera conjunta presentan información redundante.

Koren & Sill (2013), argumentan dos problemas básicos del uso de RMSE: primero que no puede representar escalas de calificación que varían entre distintos usuarios, y segundo que no puede ser aplicado en casos donde las calificaciones son ordinales. Proponen, por tanto, FCP (Fraction of Concordant Pairs):

$$FCP = \frac{n_c}{n_c + n_d}$$

Donde $n_c = \sum_u \left|\{(i,j) \mid \hat{r}_{ui} > \hat{r}_{uj} \; and \; r_{ui} > r_{uj}\}\right|$ y n_d hará referencia a los pares discordantes de manera similar. Así mismo, cuando los dos problemas atribuidos anteriormente a RMSE sean obvios, otras posibles medidas de "Accuracy" son: Precision y Recall. No obstante, si bien se pretende remarcar la idea del condicionamiento de la técnica de evaluación a la naturaleza del set de datos, para el caso que ocupa actualmente será suficiente con MAE, puesto que escapa de los problemas narrados.

2. Aplicación práctica

2.1. Formulación del problema

Dada la sobrecarga de información digital disponible y el crecimiento de la misma que se sigue esperando, tal y como indica Samsung Display (2017), buscar una película para el tiempo de ocio puede ser una labor tediosa.

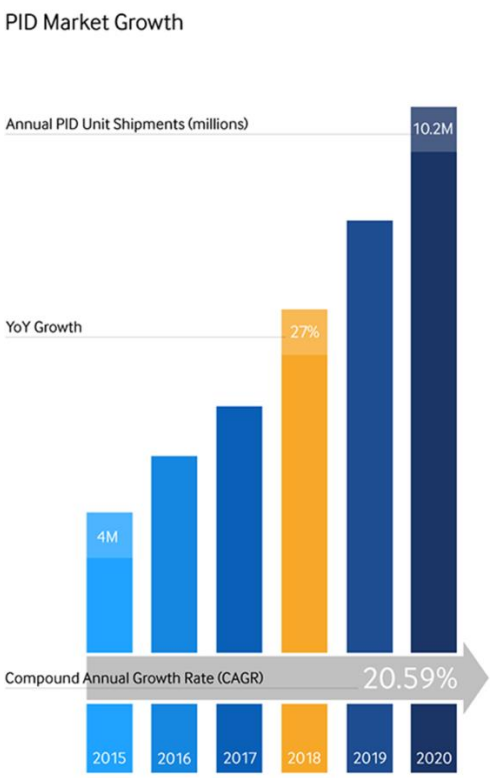

Gráfico 2. Top 5 Digital Signage Trends for 2018. Fuente: Samsung Display

Además, podemos resultar "engañados" por la aparente portada o sinopsis y concluir que la película no se adapta a nuestras preferencias. Conocido dicho problema, las páginas web que se enmarcan en este entorno, pueden actuar recomendando las películas que más se adaptan a cada usuario, reduciendo tiempo de búsqueda y además reduciendo la posibilidad de "ser engañados".

El presente estudio, busca minimizar esos dos problemas. Para ello, parte de un set de datos que contiene ratings, con un rango del 1 al 5, asignados por un conjunto de usuarios hacia una serie de películas concretas, a las que se hace referencia como items. El objetivo final radica en, a partir de los ratings recogidos por el sistema, predecir calificaciones que darían usuarios a películas que no han visto. Una vez realizada la predicción, sería fácil realizar la recomendación: sólo habría que marcan un umbral respecto al rating que indica un fuerte interés por la película, por ejemplo el 4, y recomendar a un usuario aquellas películas en las que se predice un rating igual o

superior a dicho umbral. Para poder realizar las predicciones, se divide el set de datos inicial, de manera aleatoria, en un set de entrenamiento o training set y un set de validación o test. Una vez realizada dicha división, se planteará un escenario base en el que se aplica cada uno de los algoritmos explicados en el punto 1.4 al set de entrenamiento y se realizan predicciones en test, procediendo a la elección del mejor algoritmo, para este set de datos en concreto, según indiquen los resultados arrojados por MAE.

Una vez concebido dicho escenario base se intentará mejorar como se indica a continuación. Mediante los resultados arrojados por el análisis de los datos que se realizará en el punto 2.3, se concluyen una serie de atributos, que nos permitirán caracterizar distintos perfiles de usuarios, en base a las cuales definir una clusterización de usuarios totalmente extremos que, en opinión propia, no parece lógico analizar en un mismo training set. De esta forma se obtendrán clusters de usuarios, que definirán conjuntos de datos distintos. Cada uno de estos i[7] subsets de datos, se dividirá en un conjunto de entrenamiento y un conjunto de test, aplicando para cada uno de ellos los algoritmos del punto 1.4 y escogiendo para hacer las predicciones aquel que mejores resultados posea, en términos de MAE para cada subset. Por tanto, esto nos permitirá segmentar usuarios dentro los datos originales, pudiendo aplicar el mejor algoritmo en términos de MAE dentro de cada grupo[8], para fusionarlos posteriormente a la realización de las predicciones en un único set que intentará mejorar los resultados del escenario base, obteniendo un menor error.

2.2. Datos.

La aplicación práctica se desarrolla en base a MovieLens 100K Dataset[9]. Dicho dataset recoge datos de calificación o ratings del sitio web MovieLens y es recopilado y proporcionado por GroupLens. Así mismo, es un referente, usado en múltiples estudios, a la hora de comenzar a entender y estudiar los sistemas de recomendación. El GroupLense es un laboratorio de investigación en el Departamento de Ciencias de la Computación e Ingeniería de la Universidad de Minnesota, Twin Cities que se

[7] Se distinguirán tantos set de datos como número de clusters.
[8] Por tanto, no se obliga a la elección de un único algoritmo para el set de datos global, como si ocurre en el escenario base, sino que distintos clusters podrán adaptarse a distintos algoritmos.
[9] Los datos pueden descargarse directamente del siguiente enlace https://grouplens.org/datasets/movielens/

especializa en sistemas de recomendación, comunidades en línea, tecnologías móviles y ubicuas, bibliotecas digitales y sistemas de información geográfica local.

MovieLens 100K presenta varios set de datos. El Dataset principal[10] consiste en 100.000 ratings asignados por 943 usuarios a 1682 películas. Dichos ratings oscilan numéricamente del 1 al 5, siendo 1 la calificación mínima a asignar a una película y 5 la máxima. Se introduce que los ratings indican realmente estrellas de valoración. Los datos fueron recolectados durante 7 meses desde el 19 de septiembre de 1197 hasta el 22 de abril de 1998. Destaca que el conjunto de datos está filtrado, de manera que sólo incluye usuarios que han valorado como mínimo 20 películas distintas, puesto que cada película solo puede ser valorada 1 vez. Por tanto, la información presente en dicho dataset es el userID (identificador de usuario), movieID (identificador de película), rating (valoración asignada por el userID a movieID) y timestamp (momento en el que se realiza la valoración). Además también se hará uso del dataset que recoge las características de los usuarios que realizan las valoraciones [11] y que contiene el género, la edad, la ocupación y el código postal.

Toda la información relativa a las películas como, por ejemplo, el título, la fecha de lanzamiento o el género, también es proporcionada por GroupLens[12]

2.3. Análisis de los datos

Con objeto de comprender, entender los datos y así poder definir una serie de atributos que nos ayuden en los puntos posteriores, antes de realizar las predicciones que darán forma al sistema de recomendación, se procede a hacer un análisis de los mismos.

2.3.1. Análisis descriptivo

El primer paso que se realiza en este apartado, consiste en observar simplemente un histograma de ratings con el objetivo de conocer en un primer vistazo el interés general por el catálogo global, obteniendo los siguientes resultados.

[10] u.data
[11] Dicho dataset se presenta como u.user
[12] Dicho dataset se presenta como u.item

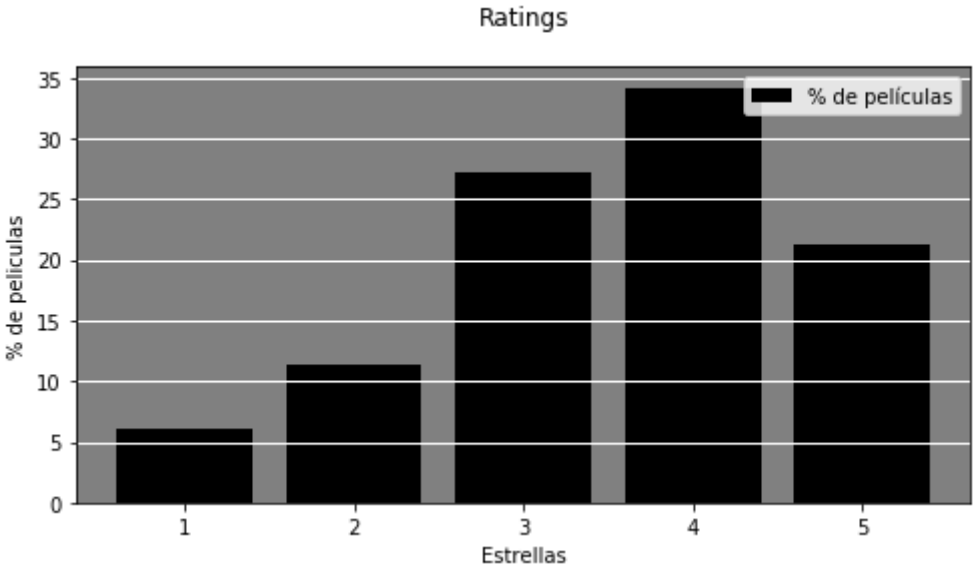

Gráfico 3. Porcentaje de películas por rating. Fuente: Elaboración propia.

En principio parece que la moda es 4 estrellas, y la media se centra en 3,53, con lo cual parece que a primera vista los usuarios valoran bien el catálogo disponible. La siguiente cuestión que se aborda es si habría relación entre el número de películas que un usuario valora y el rating medio de valoración, es decir, si puede ocurrir que a medida que un usuario valora y, por tanto, ve más películas se vuelve más exigente. El siguiente gráfico muestra la relación entre la valoración media y el número de valoraciones realizadas por cada usuario.

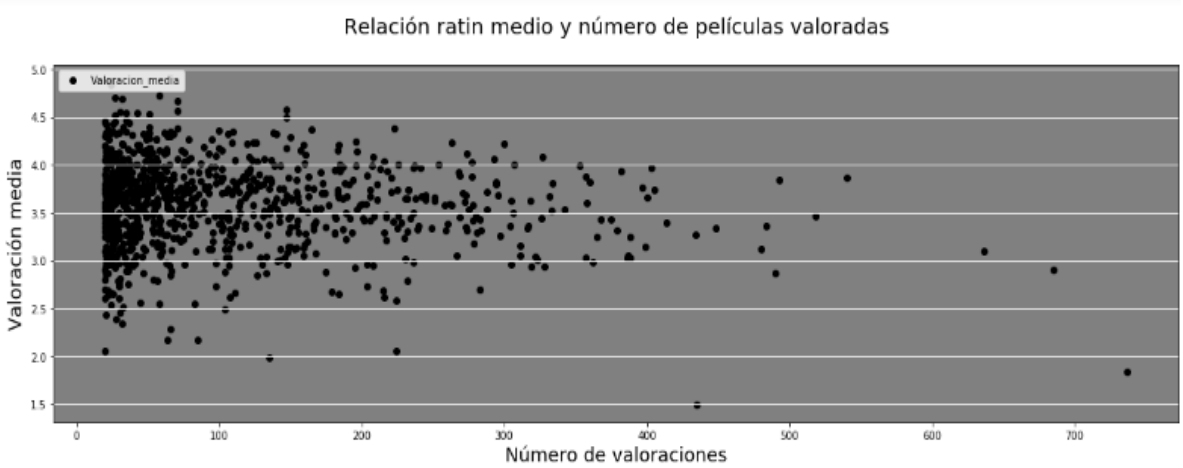

Gráfico 4. Valoración media en función del número de películas valoradas. Fuente: elaboración propia.

Como se muestra en el gráfico no existe una relación clara, pero sí parece apreciarse una tendencia bajista en la valoración media a medida que aumenta el número de valoraciones realizadas por el usuario. La correlación existente entre dichas dos variables es de -0.1378, por tanto, si bien es negativa, es mínima. Si consideramos la

edad, la relación parece más clara. A mayor edad, en general excepto los segmentos de usuarios de 0 a 9 años y de 10 a 19, se cumple que se realiza un menor número de valoraciones y además la valoración media es mayor. Es decir, los jóvenes valoran más películas, son más proactivos, excepto el extremo de edad inferior a los 18 años, y son más exigentes, tanto es así que incluso los grupos entre 10 y 19 años y entre 20 y 29 tienen una valoración media inferior a la media global, que se refleja por la línea marrón horizontal en el gráfico que se muestra a continuación.

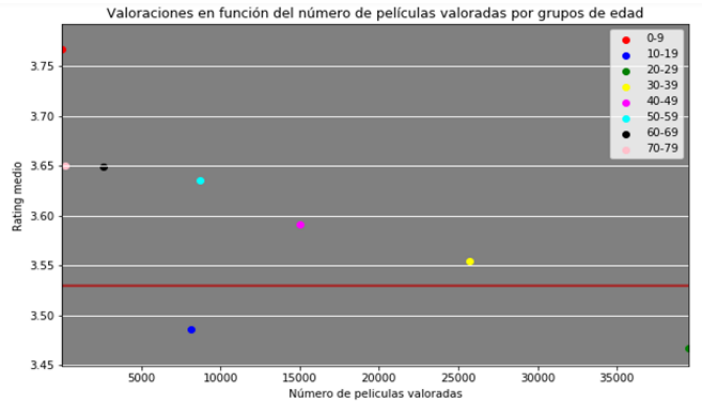

Gráfico 5. Valoración media en función del número de películas valoradas por grupos de edad. Fuente: elaboración propia.

Si añadimos la distinción por género y, por tanto, se observa la relación entre el número de películas vistas y la valoración media por grupo de edad y género, se obtienen los siguientes gráficos:

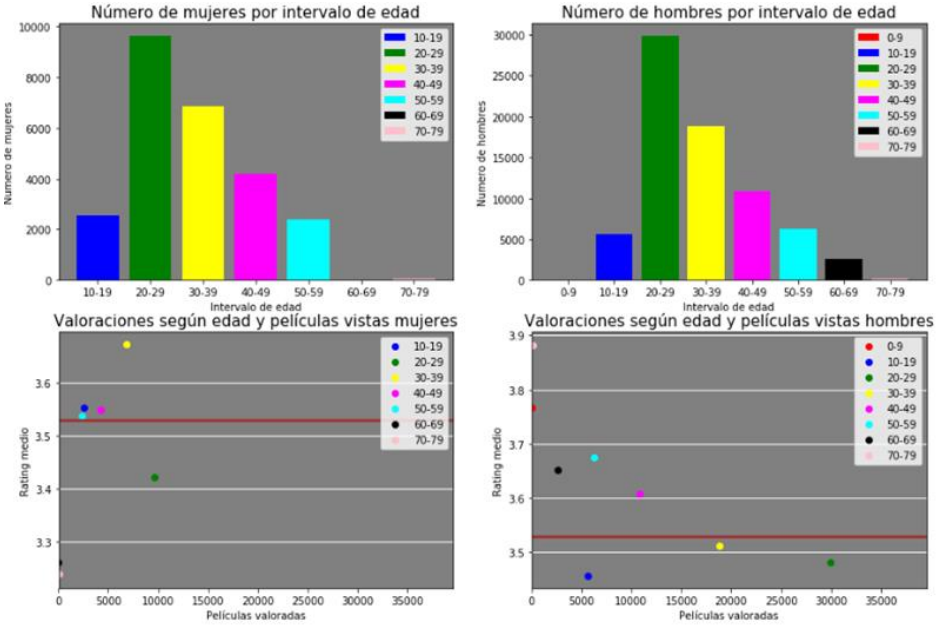

Gráfico 6. Valoración media en función del número de películas valoradas por grupos de edad y género. Fuente: elaboración propia.

Si bien es cierto, que el género masculino, cumple los patrones generales, es decir, a mayor edad excepto los dos grupos de edad inferiores a la mayoría de edad, menos valoraciones realizadas y rating medio superior a la media, en el caso de las mujeres, las valoraciones realizadas por los grupos de edad más avanzados 60-69 y 70-79 son mínimas y además los rating son inferiores al rating medio. Así mismo, el comportamiento está menos claro puesto que los ratings medios más altos vienen dados por el grupo de 30-39 años. Con lo cual el género es un factor a tener en cuenta para el estudio del rating.

Aunque se ha estudiado la relación entre número de películas valoradas y rating medio a nivel de usuario, también sería conveniente cambiar la visión en ese estudio y centrarnos en el número de veces que una película es valorada, por distintos usuarios, y su rating medio. Es decir, ¿puede haber una relación entre la calidad de la película y el número de valoraciones? Se cumpliría por tanto en base a esta idea, que las películas de mayor calidad, no solo tienen mayor rating medio sino que, además, también son valoradas más veces. La correlación entre los dos factores a analizar arroja un resultado de 0.43, y es por tanto, positiva y moderada.

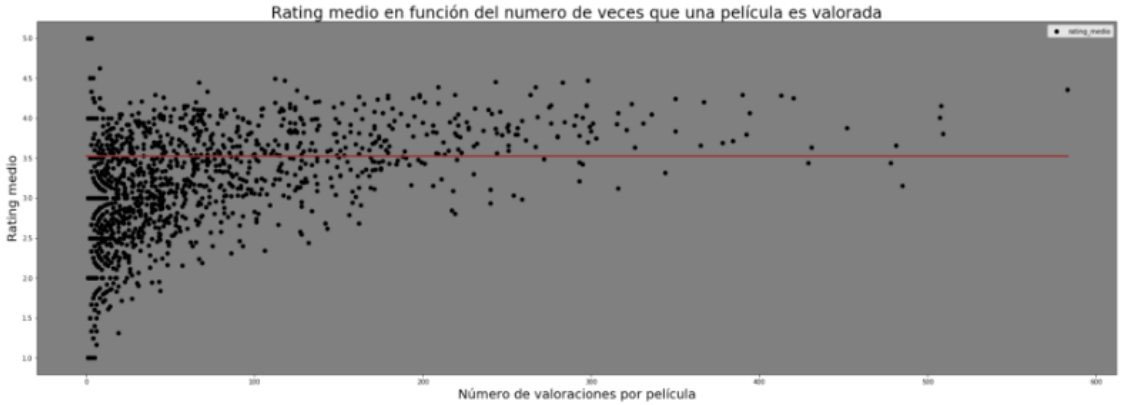

Gráfico 7. Valoración media en función del número de valoraciones por película. Fuente: elaboración propia.

No obstante, como se puede observar en el gráfico las películas con pocas valoraciones introducen mucha distorsión, puesto que dichas películas han sido valoradas 1 o 2 veces en general, tienen valoraciones muy extremas, algo que podría justificarse en que corresponden a gustos muy específicos y poco compartidos. Pero aun así no deben eliminarse de los datos, puesto que aunque la película haya sido valorada una sola vez,

el usuario ha hecho al menos 20 valoraciones, y puede arrojar información sobre el patrón de gustos del usuario y sobre nuevas posibles recomendaciones.

Por tanto y concluyendo, se han detectado varios factores que influyen en la calificación de un usuario a una película:

1.-El número de valoraciones del usuario

2.- El rating medio del usuario

3.- Grupo de edad

4.- El género

5.- El número de valoraciones que tiene una película

6.- El rating medio de la película

Se proponen las siguientes variables como atributos necesarios para identificar grupos de usuarios, con comportamientos "más similares" a la hora de calificar:

1.- Valoraciones del usuario. Suma de todas las valoraciones por usuario, con un mínimo de 20 y un máximo de 737. El número medio de valoraciones es de 106.

2.- Rating medio del usuario. Media de todos los rating por usuario. Los rating oscilan entre 1 y 5, no obstante agrupando por usuario el mínimo es 1,5 y el máximo 4,87.

3.- Edad. Referencia la edad concreta de cada usuario expresada en años.

4.- Género. La variable género se transforma en una variable categórica que tomará el valor 1 para referenciar al sexo masculino y -1 para el femenino.

5.- Media del número de valoraciones de todas las películas vistas. Esta variable selecciona las películas valoradas por el usuario y calcula la media del número de valoraciones dadas por el resto de usuarios a las mismas. El objetivo es tratar de identificar si el usuario valora películas "muy valoradas" o "nada valoradas" algo que podría entenderse, como una aproximación de si ve películas que ven muchos usuarios, y por tanto es "usuario común" o si por el contrario tiene "gustos extraños", califica películas que nadie califica y es un "usuario raro".

6.- Grado de diferencia. Esta variable calcula la diferencia entre dos valores: el rating medio del usuario en cuestión y la media del rating medio de todos los usuarios que han valorado las películas que ha visto el usuario. Indica la desviación con respecto a los gustos "comunes". Si la variable es positiva dichas películas han sido valoradas por el usuario al alza con respecto a la media del resto de usuarios que las han valorado. Lo contrario ocurre si es negativa.

2.3.2. Modelo multinomial ordenado

Dado el análisis anterior, el objetivo que se persigue es identificar una serie de características en los usuarios que ayuden a explicar el rating, para posteriormente tratar de realizar un clustering en base a ellas, y plantear una propuesta de mejora basada en clusters de usuarios para las predicciones frente a un escenario base, siendo este último el que englobe en su conjunto los datos originales. Para tener una visión más técnica de que los atributos, que se han escogido en base al análisis tienen razón de ser, se procede a realizar un modelo multinomial ordenado. Esta tipología de modelos econométricos, se engloba dentro de los de elección discreta, que son aquellos en los que la variable dependiente es discreta o toma valores discretos. Los modelos multinomiales ordenados tienen la particularidad de que la variable dependiente discreta toma más de dos valores y éstos están ordenados. Es decir, como indica Donate & Hernández (2007), la variable dependiente expresa preferencias u opiniones de los individuos sobre una determina cuestión, y las alternativas expresan un orden de utilidad y por ello, tienen carácter ordinal. Por tanto, el modelo indicará si los atributos, que serán las variables explicativas del mismo, realmente explican a la variable dependiente, es decir el rating. Se centra la atención en ver el efecto y comprobar si todas las variables explicativas son conjuntamente significativas.

En base a los atributos definidos en el punto anterior, se propone como variable dependiente el rating medio, por supuesto, redondeado puesto que la variable ha de ser discreta. Como variables explicativas se introducirán el resto de atributos enunciados, que son los siguientes: edad, valoraciones del usuario, género, media del número de valoraciones de todas las películas vistas y grado de diferencia. Para derivar formalmente el modelo multinomial ordenado, se tiene para el individuo i lo siguiente:

$$Y_i^* = x_i'\beta + u_i$$

Donde Y_i^*, que es la utilidad real del individuo, no es una variable de elección observada, y se relaciona con la variable observada Y_i de la siguiente manera:

$$Y_i = 1 \text{ si } Y_i^* \leq 0$$
$$Y_i = 2 \text{ si } 0 < Y_i^* \leq \alpha_1$$
$$Y_i = 3 \text{ si } \alpha_1 < Y_i^* \leq \alpha_2$$
$$Y_i = 4 \text{ si } \alpha_2 < Y_i^* \leq \alpha_3$$
$$Y_i = 5 \text{ si } \alpha_3 < Y_i^* \leq \alpha_4$$

Los umbrales o alfas, pueden ser conocidos o no, y se estimarán conjuntamente con los β. La estimación del modelo se llevará a cabo por el método de máxima verosimilitud. Por tanto, los parámetros β y α serán aquellos que maximicen la función de verosimilitud. Con la ayuda de Stata se tiene el siguiente resultado:

```
Ordered logistic regression                     Number of obs     =        943
                                                LR chi2(5)        =     806.07
                                                Prob > chi2       =     0.0000
Log likelihood = -349.56437                     Pseudo R2         =     0.5355
```

rating_medio_redondeado	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
media_global_resto_valoraciones	.0188103	.0032627	5.77	0.000	.0124156	.025205
valoraciones_del_usuario	.0017267	.0012271	1.41	0.159	-.0006783	.0041317
age	.0348981	.0082923	4.21	0.000	.0186455	.0511507
_Igender_2	.3770172	.217975	1.73	0.084	-.050206	.8042404
grado_de_diferencia	8.898714	.5534606	16.08	0.000	7.813951	9.983477
/cut1	-9.490219	1.440789			-12.31411	-6.666325
/cut2	-4.318949	.8644964			-6.013331	-2.624567
/cut3	4.612527	.8172466			3.010753	6.214301
/cut4	14.78821	1.123469			12.58625	16.99017

Tabla 1. Modelo multinomial ordenado. Primera aproximación. Fuente: elaboración propia

En principio, se deduce que las variables son conjuntamente significativas, tal y como indica el p-valor de significatividad conjunta que es 0.000. Además para las 5 clases o 5 posibles elecciones existentes, se predicen 4 umbrales, lo cual está en consonancia con los datos. En cuanto al Pseudo R2 cuenta con detractores sobre el uso o interpretación del mismo, puesto que es una medida de mejora relativa en la log-verosimilitud del modelo que incluye variables explicativas respecto al modelo que sólo incluye constante. Si bien se encuentra entre 0 y 1 no representa proporción de varianza explicada por el modelo. Por tanto, para poder determinar el alcance del modelo se recurrirá, posteriormente a una tabla de errores y aciertos. Por otro lado, se destaca que

en esta tipología de modelos los coeficientes β no coinciden con los efectos marginales, por tanto los primeros resultados carecen de capacidad interpretativa y se procede a analizar los resultados de las semielasticidades.

Si se analizan dichos resultados para la media del número de valoraciones de todas las películas vistas por usuario, a continuación se observa como a medida que aumenta el número de valoraciones dadas a una película por el resto de usuarios, es decir, a medida que una película es más valorada, disminuye la probabilidad de tener un rating bajo. El comportamiento no discrepa de lo que se desprendía del análisis de los datos.

	Delta-method				
	dy/dx	Std. Err.	z	P>|z|	[95% Conf. Interval]
media_global_resto_valoraciones					
_predict					
1	-.0000152	.0000105	-1.45	0.147	-.0000357 5.34e-06
2	-.0000908	.0000258	-3.52	0.000	-.0001413 -.0000403
3	-.0017965	.0002945	-6.10	0.000	-.0023737 -.0012193
4	.0017585	.0002884	6.10	0.000	.0011932 .0023238
5	.000144	.0000363	3.96	0.000	.0000727 .0002152

Tabla 2. Modelo multinomial ordenado. Efectos marginales: media global del resto de valoraciones. Fuente: elaboración propia.

Respecto a la edad, tal y como se intuía en el análisis de los datos, a medida que aumenta, entonces la probabilidad de asignar ratings bajos disminuye en pro de la probabilidad de asignar ratings altos (4 o 5 estrellas).

	Delta-method				
	dy/dx	Std. Err.	z	P>|z|	[95% Conf. Interval]
age					
_predict					
1	-.0000282	.00002	-1.41	0.158	-.0000673 .000011
2	-.0001685	.0000553	-3.05	0.002	-.0002768 -.0000602
3	-.003333	.0007715	-4.32	0.000	-.0048452 -.0018208
4	.0032626	.0007555	4.32	0.000	.0017818 .0047433
5	.0002671	.0000804	3.32	0.001	.0001096 .0004247

Tabla 3. Modelo multinomial ordenado. Efectos marginales: edad. Fuente: elaboración propia

En cuanto a la variable valoraciones del usuario, el p-valor indica que no es significativa, aunque se recuerda que conjuntamente si lo son todas ellas. Se observa que al aumentar el número de valoraciones de un usuario aumenta la probabilidad de asignar un mayor número de estrellas, o lo que es el mismo pertenecer a grupos de usuario más satisfechos. Esto choca con la intuición del análisis inicial, puesto que aparentemente ocurría que la correlación entre el número de valoraciones y el rating medio era muy baja pero negativa, y además si se considera la edad, entonces los

usuarios jóvenes solían valorar más y eran más estrictos. A continuación se muestran los resultados:

	Delta-method dy/dx	Std. Err.	z	P>|z|	[95% Conf.	Interval]
valoraciones_del_usuario _predict						
1	-1.39e-06	1.37e-06	-1.02	0.308	-4.07e-06	1.28e-06
2	-8.34e-06	6.13e-06	-1.36	0.174	-.0000203	3.67e-06
3	-.0001649	.0001169	-1.41	0.158	-.0003941	.0000642
4	.0001614	.0001143	1.41	0.158	-.0000627	.0003855
5	.0000132	9.73e-06	1.36	0.174	-5.85e-06	.0000323

Tabla 4. Modelo multinomial ordenado. Efectos marginales: valoraciones del usuario. Fuente: elaboración propia

Centrando la atención en la variable grado de diferencia, se recuerda que indicaba la desviación del usuario con respecto a los gustos comunes, y que el signo indica si la desviación es en alza y por tanto, se tiende a valorar más en media que el resto de usuarios o menos en caso contrario. Los resultados muestran que la variable es significativa y que como se esperaba a medida que aumenta disminuye la probabilidad de dar menos de 3 estrellas, en pro de dar 4 o 5 estrellas. Dichos resultados son los siguientes:

	Delta-method dy/dx	Std. Err.	z	P>|z|	[95% Conf.	Interval]
grado_de_diferencia _predict						
1	-.0071874	.0048266	-1.49	0.136	-.0166473	.0022725
2	-.0429597	.0101593	-4.23	0.000	-.0628715	-.0230478
3	-.8498873	.0203206	-41.82	0.000	-.889715	-.8100596
4	.8319222	.0217554	38.24	0.000	.7892825	.874562
5	.0681121	.0132141	5.15	0.000	.042213	.0940112

Tabla 5. Modelo multinomial ordenado. Efectos marginales: grado de diferencia. Fuente: elaboración propia

Por último, mostrando para todos los niveles una no significatividad, se encuentra la variable género en la que se observa que el hecho de que la valoración sea masculina aumenta la probabilidad de asignar ratings altos. Se muestran los resultados.

.

	Delta-method dy/dx	Std. Err.	z	P>|z|	[95% Conf.	Interval]
_Igender_2 _predict						
1	-.0003045	.0002719	-1.12	0.263	-.0008373	.0002283
2	-.0018201	.0011208	-1.62	0.104	-.0040168	.0003766
3	-.0360077	.0207081	-1.74	0.082	-.0765949	.0045795
4	.0352465	.020245	1.74	0.082	-.0044328	.0749259
5	.0028857	.001771	1.63	0.103	-.0005853	.0063568

Tabla 6. Modelo multinomial ordenado. Efectos marginales: género. Fuente: elaboración propia

Se procede a realizar la tabla de errores y aciertos para ver la capacidad predictiva del modelo. No obstante es muy importante comprobar anteriormente el desequilibrio de clases, que se refleja a continuación:

.

rating_medio_redondeado	Freq.	Percent	Cum.
1	1	0.11	0.11
2	12	1.27	1.38
3	355	37.65	39.02
4	562	59.60	98.62
5	13	1.38	100.00
Total	943	100.00	

Tabla 7. Modelo multinomial ordenado. Desequilibrio de clases. Fuente: elaboración propia

Como se puede observar las clases no están equilibradas. Sólo una persona tiene un rating medio de 1 estrella, 12 personas tienen de media 2 estrellas y 13 personas han valorado en media con 5 estrellas, estando repartido el grueso entre el 3 y 4. Esto se intuía, puesto que no se están considerando ratings reales, sino que se ha centrado el análisis en usuarios y no en películas o valoraciones, y por tanto, como cada persona al menos ha valorado 20 veces, la media de los ratings por usuario suele estar en torno al 3 o 4 escapando de valores extremos. Siendo conscientes de los problemas que puede provocar el desequilibrio de clases se procede a observar la tabla de errores y aciertos:

rating_medio_redondeado	index_rep_pred 2	3	4	5	Total
1	1	0	0	0	1
2	7	5	0	0	12
3	2	276	77	0	355
4	0	60	499	3	562
5	0	0	8	5	13
Total	10	341	584	8	943

Tabla 8. Modelo multinomial ordenado. Tabla de errores y aciertos. Fuente: elaboración propia.

Rating_medio_redondeado refleja los valores de la variable dependiente mientras que index_rep_pred refleja las predicciones que arroja el modelo multinomial ordenado. Por tanto podemos observar lo siguiente:

- Sólo 1 usuario tiene 1 estrella de media, y el modelo predice que tiene 2. En esta categoría acierta un 0% de las veces.
- 12 usuarios tienen una valoración media de 2 estrellas, el modelo predice que 7 de ellos tienen 2 estrellas y 5 tienen 3 estrellas. Por tanto, ha predicho bien esta categoría un 58.3% de las veces.
- 355 usuarios valoran de media con 3 estrellas y el modelo predice, que de ellos tienen 3 estrellas de media 276, 77 tienen 4 estrellas de media y para 2 de ellos predice 2 estrellas. En esta categoría el modelo predice bien el 78% de los casos.
- 562 usuarios tienen de media 4 estrellas, el modelo predice que esa puntuación correspondería a 499 usuarios, a otros 60 de ellos les asigna 3 estrellas y a 3 asigna 5 estrellas. En esta categoría el modelo ha predicho bien el 89% de los casos.
- Finalmente 13 usuarios tienen 5 estrellas y el modelo concluye que sólo serían 5, prediciendo para los 8 restantes 4 estrellas. El modelo actúa bien el 38% de las veces.

El grado de aciertos total se sitúa en un 83,46%, lo que es a priori buen valor. No obstante, combinado con el desequilibrio de clases los resultados no son tan buenos. Como se observa el hecho de que solo haya 1 observación con rating medio 1 está provocando la no significatividad de los resultados comentados anteriormente para el caso en el que la variable dependiente toma valor 1. Es decir, los valores extremos tanto 1 estrella como 5 estrellas son difíciles de detectar, lo cual encaja ya que se ha incluido como variable dependiente un rating medio existiendo pocos casos extremos. Como conclusión, teniendo siempre en mente el problema del desequilibrio de clases, el modelo confirma, a grandes rasgos, los patrones detectados en el análisis de los datos y se concluye que todas las variables son conjuntamente significativas.

2.3.3. Análisis de componentes principales

Como indica Jolliffe (2011) “La idea central del análisis de componentes principales (PCA) es reducir la dimensionalidad de un conjunto de datos que consiste en un gran

número de interrelaciones entre variables, conservando la mayor cantidad posible de la variación presente en el conjunto de datos" (p. 1). Para ello se recurre a un nuevo conjunto de variables que tienen dos características: no están correlacionadas y están ordenadas para que los primeros retengan la mayor proporción de variabilidad. Este nuevo conjunto de variables se define como componentes principales.

Por tanto, el análisis de componentes principales (PCA) es una técnica de reducción de dimensionalidad de una matriz de datos. En este caso, dado que se caracterizará a los usuarios por los atributos definidos en el análisis de datos y ratificados por el modelo multinomial ordenado, se tienen 6 atributos para cada uno de los usuarios. Recurriendo a PCA, se pretende entender mejor las variables, así como reducir el número de atributos para así poder realizar una representación gráfica de los distintos cluster que se obtendrán en el apartado siguiente. Con lo cual se persiguen dos objetivos: visualización de datos multidimensionales y mejor comprensión de los mismos.

Los componentes principales permitirán por tanto, resumir el conjunto de datos con un número menor de variables representativas que explicarán la mayor parte de la variabilidad del conjunto original. En esta técnica la dirección de la primera componente principal será aquella a lo largo de la que se manifieste una mayor variabilidad de los datos. El segundo componentes principal será descontando el primero, el segundo que más variabilidad recoja y así sucesivamente hasta el 6º componentes principal que sería el máximo al que se podría aspirar en este caso, si se quisiera recoger toda la información disponible en los datos.

La incertidumbre radica en la decisión de qué número de componentes principales ha de utilizarse, es decir, ¿cuántos componentes principales se necesitan para obtener una buena comprensión de los datos? Si bien es cierto que no hay una respuesta única y universal para esa pregunta, el gráfico de sedimentación puede ayudar en dicho camino. El gráfico de sedimentación representa la proporción de varianza explicada por cada componente principal. El objetivo es buscar un punto en el que la proporción de varianza explicada por los siguientes componentes principales sea mínima, ese punto recibe el nombre de codo de sedimentación y ayudará a identificar el número más adecuado de componentes principales, aunque se recalca que no hay una forma objetiva de concluir dicha decisión. Antes de proceder a realizar una primera aproximación, se destaca que todas las variables serán estandarizadas con objeto de que las unidades de

medida no tengan influencia en los resultados. Se muestra a continuación el gráfico de sedimentación.

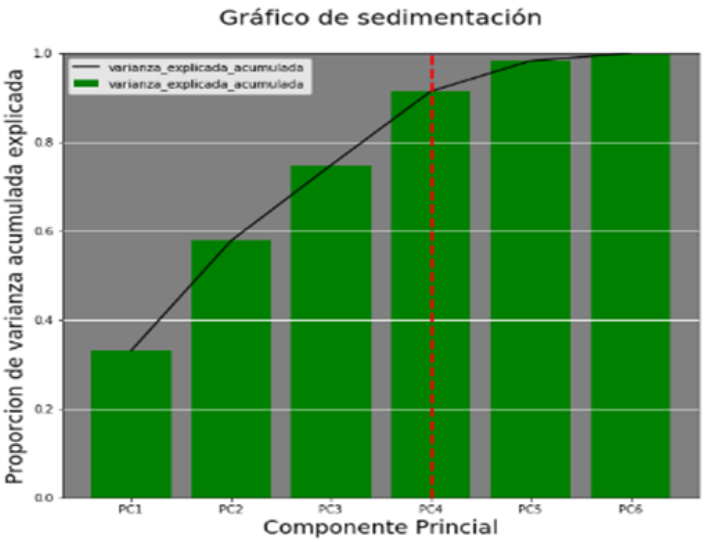

Gráfico 8. Gráfico de sedimentación. Fuente: Elaboración propia.

Como se puede observar, el codo de sedimentación se intuye en el cuarto componente principal, por tanto, con los tres primeros componentes principales se explica aproximadamente el 75% de la variabilidad de los datos, ganando también capacidad de visualización. Se representan, con la ayuda del software R, los datos en función de sus 3 componentes principales a continuación:

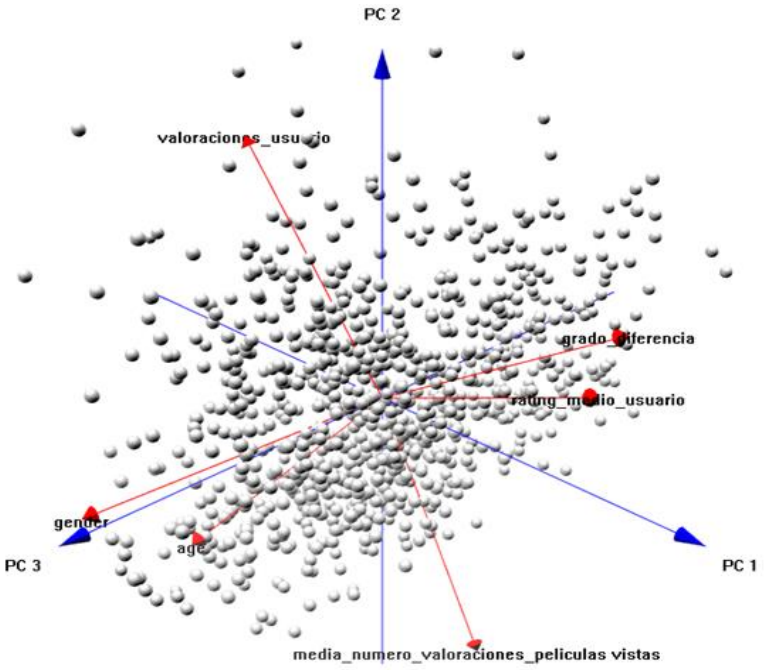

Gráfico 9. 3D Biplot PCA. Fuente: Elaboración propia.

Como se puede observar el primer componente principal coloca pesos altos en rating medio del usuario y grado de diferencia. También coloca pesos altos, aunque en menor medida, en la media del número de valoraciones de las películas vistas y, con signo opuesto, en las valoraciones del usuario. Se podría resumir el primer componente principal como aquel que recoge toda la información posible de ratings y, en menor medida, valoraciones.

El segundo componente principal, coloca pesos mayoritarios en valoraciones del usuario y con signo opuesto en la media del número de valoraciones de las películas vistas. Menor peso aunque también considerable centra en rating medio del usuario y grado de diferencia. Por tanto, el segundo componente es el caso inverso del primero, recogiendo sobre todo información de valoraciones y en menor medida de ratings.

Por último el tercer componente coloca su peso mayoritario en género y en edad. Este componente recoge "características personales" de los usuarios.

Como conclusiones destacan que valores muy por debajo de la media en el primer componente principal suelen ir acompañados de valores muy extremos en el segundo componente principal. Es decir, recordando los pesos de cada componente, se interpreta que ratings inferiores a la media vienen acompañados de un número de valoraciones del usuario superior a la media (un usuario muy activo y exigente) y una media del número de valoraciones de películas vistas inferior a la media (es decir, películas poco vistas). Cambiando la posición de los ejes, se intuye más claro en el gráfico siguiente:

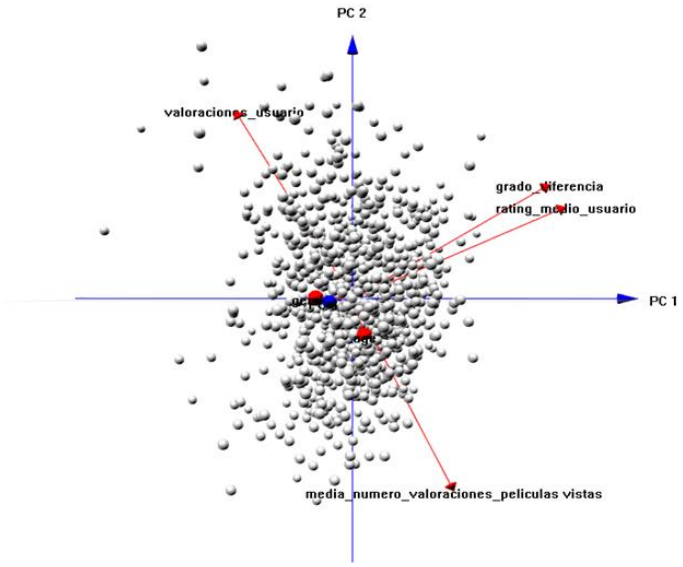

Gráfico 10. 3D Biplot PCA. Fuente: Elaboración propia.

Además también se observa que los usuarios que ven películas con valoraciones muy inferiores a la media, asignan ratings muy cercanos a la media y en la mayoría de los casos inferiores a la misma. Es decir, valores extremos inferiores a la media en el segundo componente principal vienen acompañados de valores cercanos o inferiores a la media en el primer componente principal.

Por último, se observa como valores muy inferiores a la media en el primer componente principal corresponden con valores muy cercanos o inferiores a la media en el tercer componente principal, siendo por tanto los más jóvenes quienes realizan rating más bajos, como se observa a continuación con mayor facilidad.

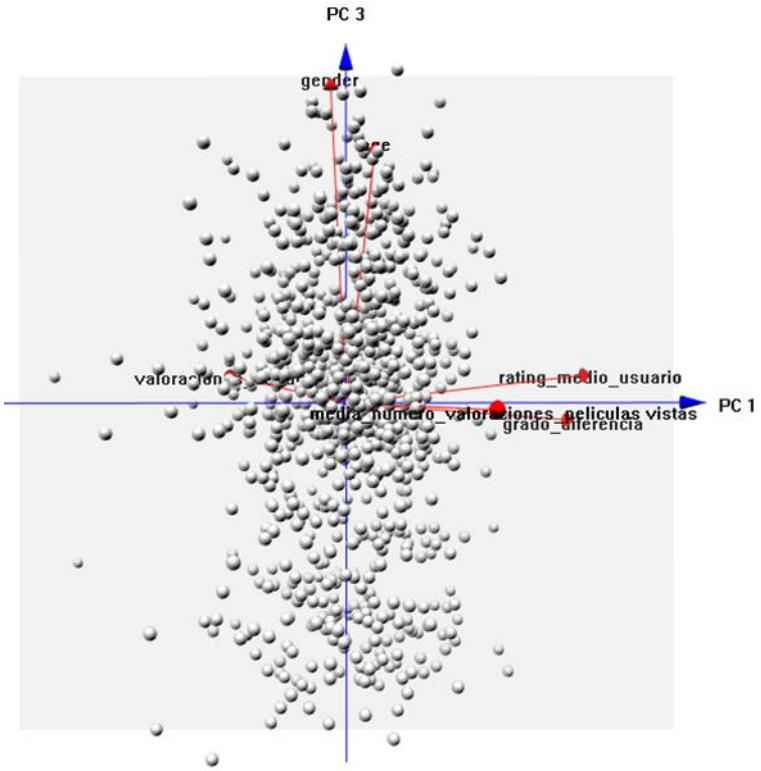

Gráfico 11. 3D Biplot PCA. Fuente: Elaboración propia.

2.3.4. Clustering

En este punto en base a los atributos detectados anteriormente, se realizará un clustering de usuarios, que se podrá visualizar con ayuda de sus tres componentes principales estudiados en el punto anterior. La idea que radica detrás de ello, es separar usuarios con características muy extremas que no "sumen" información al conjunto global, usuarios muy extraños o distintos que distorsionan más que ayudan. Supongamos que un usuario A es muy exigente y valora 20 películas con 3 estrellas, y

son sus 20 películas favoritas, mientras que otro usuario B valora las mismas 20 películas con 3 estrellas y le gustan regular, técnicas como k-vecinos más cercanos basado en usuarios los definirían como vecinos muy cercanos cuando la realidad es que sus gustos son distintos. De esta forma cuando un el usuario A vea una película que le encanta y de un rating de 5 estrellas, probablemente el usuario B le de menos de 3 estrellas. Por tanto, mismas puntuaciones esconden gustos distintos y el objetivo es encontrar clusters de usuarios extremos, en cuanto a los atributos definidos, para tratarlos de manera separada a la hora de hacer recomendaciones.

El análisis de Clusters (AC) hace referencia a todas aquellas técnicas que permiten realizar búsqueda de agrupaciones en un set de datos. El objetivo es que los individuos dentro de un grupo sean muy similares[13] o parecidos entre ellos y distintos a los individuos de otros clusters. El método que se utilizará es hierarchical clustering, y será el propio dendograma que se obtiene de esta técnica el que ayude a identificar el número de clusters más apropiado[14] para este caso en particular.

Para dar forma al cluster jerárquico, es decir a la agrupación de usuarios, se deben tener en cuenta ciertas cuestiones, que guiarán su desarrollo:

1.- ¿Se deben estandarizar las variables?

2.- ¿Qué medida de disimilitud o distancia se debe escoger?

3.- ¿Qué tipo de vinculación se debe realizar?

4.- ¿Dónde se debe cortar el dendograma?

Con respecto a la estandarización de variables, si se va a realizar una tipificación de las variables. Esto se debe a que, como se vio al principio, la media de valoraciones suele ser alta mientras que, por ejemplo, el rango de la edad o del rating medio están limitados. El objetivo no es que 1 variable marque diferencias por tener valores naturales mayores, sino encontrar patrones en las variables (todas ellas con el mismo peso) que conduzcan a la detección de la tipología de usuarios.

En cuanto a la medida de disimilitud, se escogerá la distancia euclidea, es decir, no se pretenden detectar como similares aquellos usuarios correlacionados, pero que podrían

[13] El concepto de similitud debe volver a ser definido para este caso concreto.
[14] Destacando que no hay una respuesta absoluta ni universal.

estar en distintas fases de su vida (por ejemplo por ser más jóvenes salen más y ven menos películas, por ser mayores se relacionan menos con la tecnología…), ya que esto puede dar lugar a usuarios que van a seguir mismo patrón de evolución pero que en este momento valoran de manera distinta. Caso contrario pasaría si por ejemplo, se pretende detectar pacientes que tienen una misma enfermedad, probablemente la mejor opción en ese caso sea la correlación puesto que existen distintos grados de afección en enfermedades que darán lugar a atributos muy distintos, pero dichos atributos seguirán un mismo patrón de descompensación. Por tanto, es la distancia euclidea la que tiene que detectar usuarios muy cercanos en características.

No obstante, hay otro tema importante por tratar, ¿En base a qué se vinculan dos clusters A y B? Es decir, para vincular dos individuos, la distancia euclidea mandará, pero para vincular dos clusters, ¿qué referencia será la guía? Las opciones son varias como por ejemplo el caso del vecino más cercano, que calcula todos los pares de disimilitudes entre los grupos A y B y se queda con la menor de ellas, o el caso contrario, el vecino más lejano que se queda con la mayor de ellas. No obstante, se consideran esos casos como extremos, para ocasiones en las que se pretende una agrupación predefinida. Por ejemplo, el vecino más cercano agrupará individuos muy cercanos entre sí pero distintos al resto, un agrupación "estrecha y alargada", mientras que el vecino más cercano llevará a cabo clusters más "redondeados". Por ello, en cuanto a la vinculación, para este caso concreto, se utilizará la disimilitud intercluster media. Se ha considerado mejor elección puesto que presenta la ventaja de aprovechar la información de todos los miembros de los dos clusters que se comparan.

Formalmente, como indica Gallardo (2011), la distancia o similitud entre dos clusters cualesquiera vendrá dada por la media aritmética entre la similitud de las componentes de dicho cluster. Por tanto, si suponemos que el cluster C_i integrado por n_i elementos y compuesto por dos clusters denotados por C_{i1} y C_{i2}, y el cluster n_j está integrado por j elementos, entonces la distancia o similitud entre ellos vendrá dada por la siguiente fórmula:

$$d(C_i, C_j) = \frac{d(C_{i1}, C_j) + d(C_{i2}, C_j)}{2}$$

Para dar respuesta a la última pregunta, primero se muestra el dendograma obtenido, tras aplicar la técnica de hierarchical clustering (con distancia de similitud euclidea y

vinculación intercluster media) al set de datos (estandarizado) que engloba a los 943 usuarios, caracterizados por los atributos que se extrajeron del análisis de datos y que se recuerdan a continuación:

1.- Valoraciones del usuario.

2.- Rating medio del usuario.

3.- Edad.

4.- Género.

5.-Media del número de valoraciones de todas las películas vistas.

6.- Grado de diferencia.

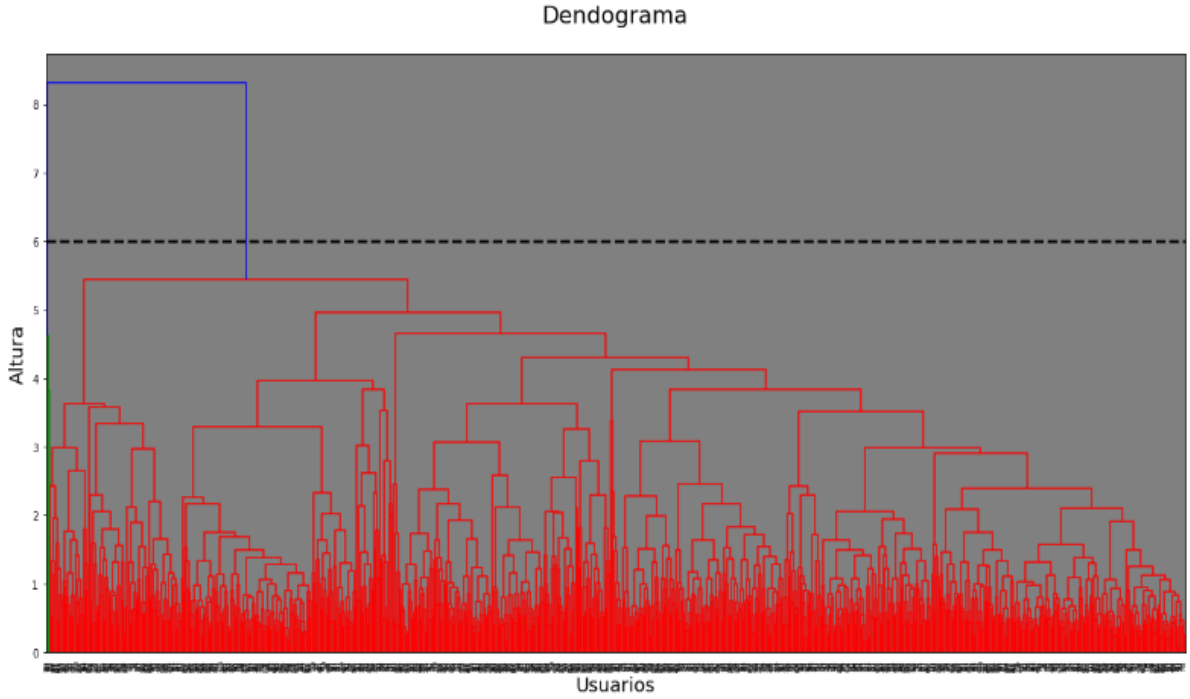

Gráfico 12. Hierarchical clustering: Dendograma. Fuente: Elaboración propia.

Cada una de las hojas, es decir, el eje x o la base del dendograma, muestra cada uno de los 943 usuarios analizados. A medida que se avanzando en altura algunas hojas comienzan a fusionarse en lo que podríamos denominar ramas que engloban a usuarios que son similares entre sí. Cuanto antes se produce la unión (es decir, a menor altura) más parecidos son los usuarios. En el dendograma se observa como a una altura de 5,5 prácticamente se han fusionado todos los individuos ya, no obstante queda un pequeño grupo a la izquierda representado en verde, que se fusiona con cluster global a una altura de 8.5, por tanto ese cluster contiene un mínimo de usuarios pero que discrepa

completamente del gran grueso restante. Estos usuarios son los que se cree que pueden introducir más ruido que información a la hora de realizar las técnicas de recomendación y, por tanto, se van a tratar todas sus valoraciones de manera separada del resto en el escenario de mejora. Gráficamente y en base a sus 3 componentes principales, la segmentación o agrupación queda de la siguiente manera:

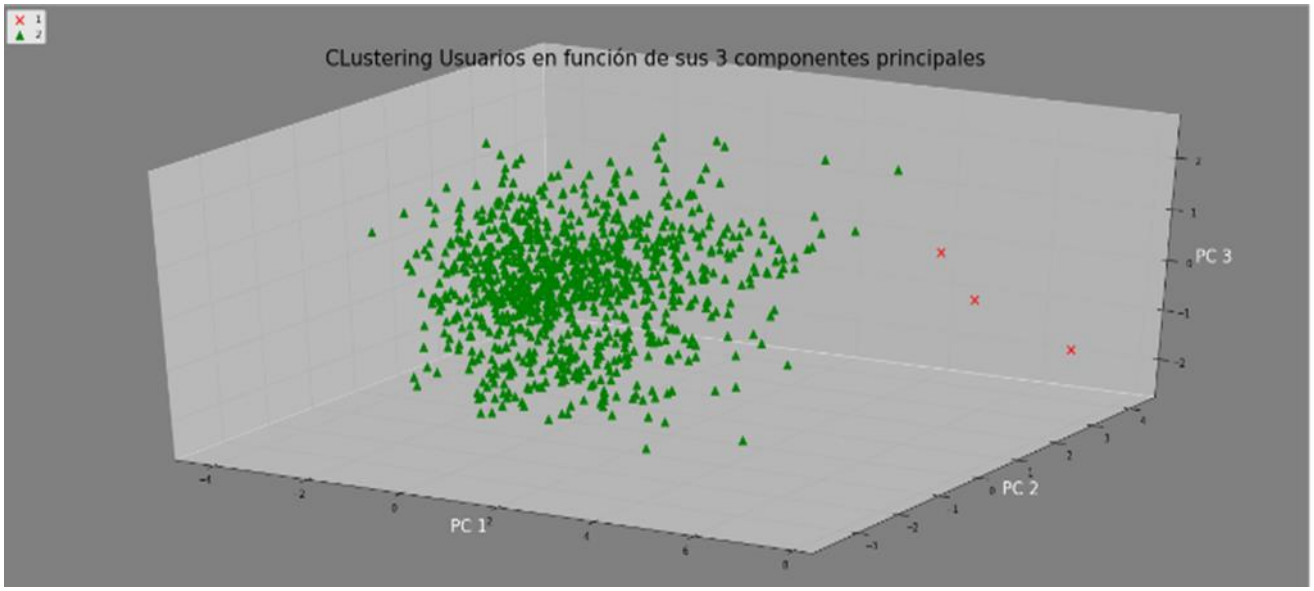

Gráfico 13. Hierarchical clustering: gráfico en función de los 3 componentes principales. Fuente: Elaboración propia.

Es claro que los 3 usuarios identificados, distan mucho del resto de usuarios, no obstante se observa simplemente la media de los atributos de cada uno de los clusters para tratar de tener una primera intuición de la agrupación:

	Media Cluster 1	Media Cluster 2
Número de usuarios en el cluster	3	940
Valoraciones del usuario.	619	104.41
Rating medio del usuario.	2.07	3.59
Edad.	32.66	34.05
Media del número de valoraciones de todas las películas vistas.	81.28	190.73
Grado de diferencia.	-1.05	+0.05

Tabla 9. Media de cada atributo por Cluster. Fuente: Elaboración propia

Como se puede observar, el Cluster 1 sólo tiene 3 usuarios, no obstante esos 3 usuarios suponen 1.857 valoraciones a estudiar, sin embargo el Cluster 2 engloba 940 usuarios que suponen 98.143 valoraciones. Según muestra la tabla 9 parece que los usuarios del Cluster 1 valoran una cantidad desorbitada de películas, que a su vez son poco valoradas por el resto de usuarios. En principio, parece que tienen unos gustos extraños y son muy exigentes puesto que en media, valoran 1 punto aproximadamente por debajo de la media. En edad son en media 2 años más jóvenes que en el caso del Cluster 2. En el caso del Cluster 2 valoran películas mucho más valoradas, realizan menos valoraciones y son menos exigentes.

Como conclusión se deduce que, en la propuesta de mejora, las 1.857 valoraciones del Cluster 1 se van a tratar de manera aislada del grueso de valoraciones, puesto que, en principio, parecen usuarios extraños que pueden distorsionar las predicciones globales.

2.4. Predicción

Como se ha dicho en anteriores puntos, se van a realizar dos escenarios distintos en este punto. En el primero de ellos, en los datos originales se realiza la separación necesaria del conjunto de datos original en entrenamiento y test, para posteriormente aplicar los algoritmos al conjunto de entrenamiento, y comparar los resultados predichos para el set de test con los resultados reales. Aquel algoritmo que tenga un menor MAE será el escogido. Se refleja el procedimiento a continuación en el gráfico 14.

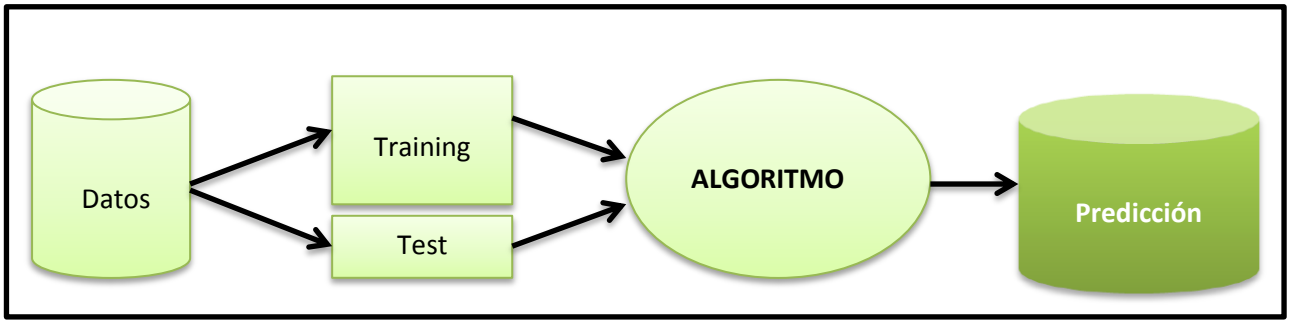

Gráfico 14. Procedimiento Escenario base. Fuente: Elaboración propia.

En el segundo escenario, no se tratan todos los datos del conjunto original de manera global, es decir, se realiza un preprocesamiento de los datos iniciales de manera que se identifican usuarios que son muy extremos y que en un principio se cree que añadirían ruido a las predicciones. Por tanto, mediante un cluster jerárquico realizado en el punto anterior, se dividen los datos originales en Cluster1 y Cluster2. A su vez, para aplicar la

técnica con garantías se divide Cluster1 en training1 y test1. Una vez entrenados los algoritmos con los datos de training1, se cotejan las predicciones, con los datos contenidos en test1. Se escoge aquel algoritmo que da lugar a menor MAE y se guardan sus predicciones (predicción 1 en el gráfico). Exactamente el mismo procedimiento se realizará para Cluster2 concluyendo con las predicciones del algoritmo que menos MAE haya presentado para ese caso. Una vez definidas tanto la predicción 1 como la predicción 2 se unirán en un único set de datos, y se calculará su MAE global con el fin último de comprarlo con la misma métrica pero asociada al escenario base. La ventaja de ello es que cada training set, tanto del cluster1 como del cluster 2, podrá adaptarse al mejor algoritmo en términos de error, permitiendo que si los datos discrepan mucho, no se vean "forzados" a adaptarse a un mismo algoritmo. Si bien es cierto, que en general y salvo discrepancias, cuanto mayor sea el set de datos mejor será el entrenamiento, en este caso, se pretende compensar la división de un set de datos global en dos más pequeños justificando que hay una parte de los datos que solo añade "ruido". Se refleja el proceso, a continuación, en el gráfico 15.

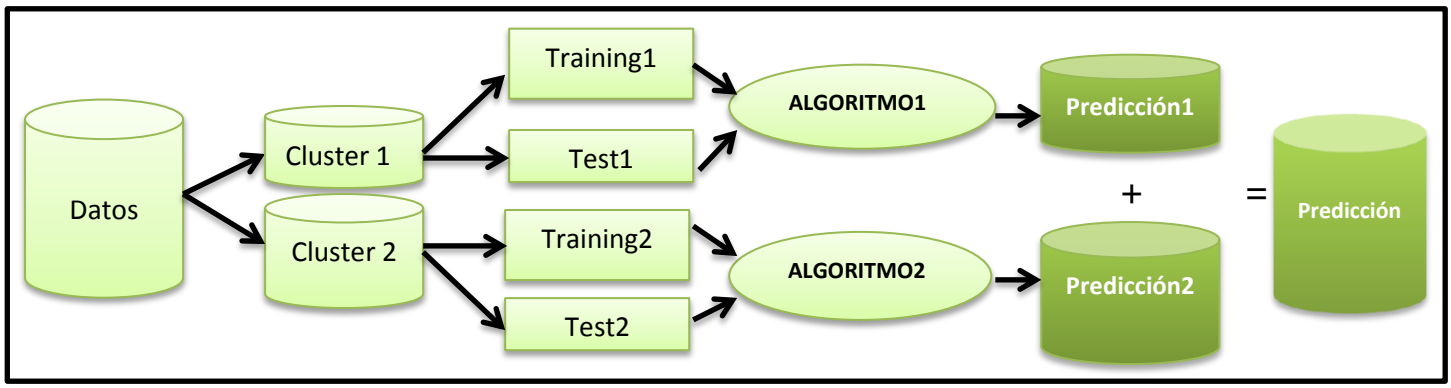

Gráfico 15. Procedimiento Propuesta de mejora .Fuente: Elaboración propia.

2.4.1. Librería

Las predicciones y aplicaciones de los algoritmos KNN, SlopeOne y SVD se realizarán con la ayuda de Surprise. Surpise es una librería de código abierto de Python enfocada a la construcción de sistemas de recomendación. Es desarrollada y mantenida por Nicolas Hug, ingeniero de software de formación, con Master's Degree en inteligencia artificial y PhD en Aprendizaje Automático del laboratorio IRIT, en Toulouse. Su PhD además ha sido focalizado en sistemas de recomendación y cuenta con varias publicaciones en el campo de Machine Learning.

2.4.2. Escenario base

La información necesaria para aplicar los algoritmos es la matriz que contiene la siguiente estructura:

USER	ITEM	RATING
181	1081	1
181	220	4

Tabla 10. Estructura datos de entrada algoritmo. Fuente: Elaboración propia

Es decir, cada fila contendrá qué usuario valora, qué item está valorando y cuántas estrellas asigna. Por tanto, la tabla global contiene 100.000 filas o instancias, de las cuales el 80% se utilizan para training y el 20% restante para test.

La intuición de KNN que reside tras esta tipología de algoritmos radica en buscar usuarios similares para el caso de KNN-user e items similares para el caso de KNN-item, recordando que la similitud vendrá impuesta por la correlación. Se realiza el método KNN en el dataset mencionado anteriormente, para todos los posibles valores que puede tomar k, acotándolos en un rango adecuado. Posteriormente, se calcula la diferencia entre los valores reales contenidos en el conjunto de test y valores predichos para cada rating, y se toma valor absoluto para mitigar el efecto del signo. Finalmente se escoge el valor de k que minimiza dicho error.

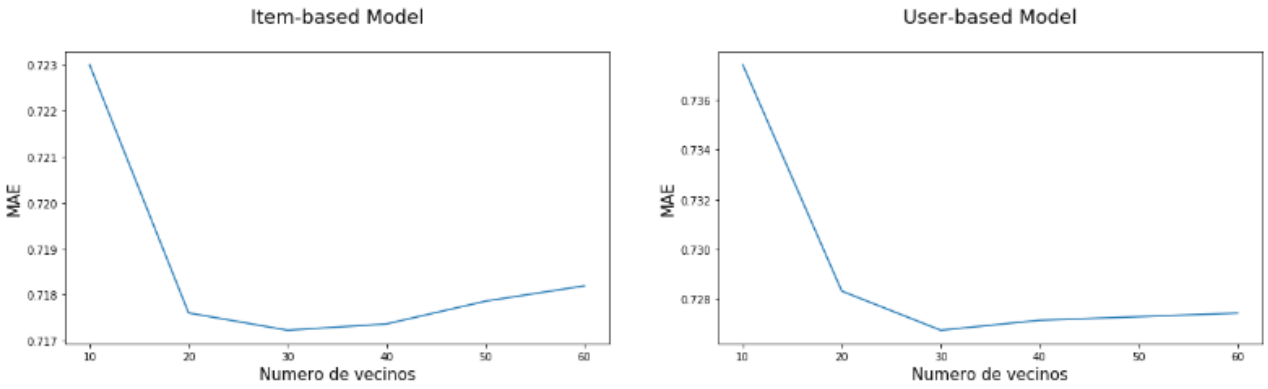

Gráfico 16. MAE para distintos valores de k en KNN-item y KNN-user .Fuente: Elaboración propia.

En este caso concreto, el valor k que minimiza el valor de MAE para ambos algoritmos es 30. Definido ese valor, se recupera el valor de MAE para ambos casos. No obstante,

como en los casos siguientes, los algoritmos se repetirán varias veces, dejando que los datos de test y entrenamiento se escojan aleatoriamente en cada una de ellas, con objeto de asegurar la generalidad de los resultados y evitar, por tanto, la dependencia de los resultados del set de entrenamiento y test. Los resultados para dichos 5 casos son los siguientes:

	KNN-Item valor de k	KNN-Item MAE	KNN-User valor de k	KNN-User MAE
CASO 1	30	**0.7168**	40	0.7275
CASO 2	30	**0.7183**	30	0.7270
CASO 3	30	**0.7119**	40	0.7219
CASO 4	40	**0.7224**	40	0.7301
CASO 5	30	**0.7167**	30	0.7275

Tabla 11. Resultados Escenario base para KNN-item y KNN-user. Fuente: Elaboración propia

Como se puede observar, en todos los casos el mejor resultado de ambos viene dado por KNN-item. Lo cual indica que para este caso concreto las características y similitudes entre películas ayudan más a definir el rating que la similitud entre usuarios. Esto es curioso puesto que todos los usuarios valoran al menos 20 veces, no obstante hay películas que solo han sido valoradas una vez, con lo cual el enfoque de items, aparentemente debería tener más problemas de arranque en frío[15]. No obstante los datos de test muestran que obtiene mejores resultados.

Si se centra la atención en el siguiente algoritmo, SlopeOne, la intuición general reside en dar respuesta a la siguiente pregunta, para poder calcular los ratings: ¿Cuánto más es valorada una película con respecto a otra?. Por tanto, este algoritmo puede ser aplicado directamente sin necesidad de ajustar ningún tipo de parámetros. El procedimiento será igual que en el caso anterior, ajustar el algoritmo con los datos de entrenamiento para validar resultados con los de test. Así mismo, se repite 5 veces obteniendo los siguientes resultados:

[15] El arranque en frío en los sistemas de recomendación es un problema típico derivado de la falta de información inicial, que no garantiza consistencia de resultados

	SlopeOne MAE
CASO 1	0.7413
CASO 2	0.7390
CASO 3	0.7327
CASO 4	0.7435
CASO 5	0.7383

Tabla 12. Resultados Escenario base para SlopeOne. Fuente: Elaboración propia

Por último, SVD como se explicó anteriormente, trata de encontrar factores que se deducen directamente de los ratings realizados y que caracterizan tanto usuarios como items. Por tanto, se centra la atención en encontrar el número de factores que produce el resultado más óptimo en términos de error[16].A continuación se muestran los resultados:

Gráfico 17. MAE en función del número de factores considerados .Fuente: Elaboración propia.

Como se observa en el gráfico la variación en el error es ínfima ya que oscila entre 0.733 y 0.739. Aun así, el número de factores que provoca el menor valor en MAE es 20, para estos datos concretos (CASO 1). Se procede de la misma forma que en los casos anteriores, obteniendo los siguientes resultados:

	Factores	SVD MAE
CASO 1	20	0.7367
CASO 2	70	0.7345
CASO 3	20	0.7298
CASO 4	60	0.7368
CASO 5	70	0.7322

Tabla 13. Resultados Escenario base SVD. Fuente: Elaboración propia

[16] Los valores asociados al problema de optimización, se dejan por defecto tal y como vienen programados. Estos valores son para el tamaño del paso 0.005 y para el parámetro de la penalización 0.02.

No obstante, para tener una guía y cotejar que todo sigue los cauces adecuados se ha confiado en la información recogida en el siguiente enlace https://www.librec.net/release/v1.3/example.html. Dicha información incluye el valor de MAE para cada una de las técnicas estableciendo el valor de los parámetros fijado. Se destaca la discrepancia propia de la división aleatoria del set de datos. Además también se han revisado estudios que profundizan en la predicción de recomendaciones para el mismo set de datos, con el fin de cotejar resultados. Por ejemplo, You, Li, Wang, & Zhao (2015) realizan una propuesta de mejora del predictor SlopeOne, plasmando que el resultado de aplicar SlopeOne anterior a la mejora de su algoritmo es aproximadamente del 0.75 en términos de MAE, algo que encaja con los resultados obtenidos anteriormente. Así mismo Sarwar, Karypis, Konstan & Riedl (2001) en su estudio item-based Collaborative Filtering Recommendation Algorithms, muestran el siguiente gráfico:

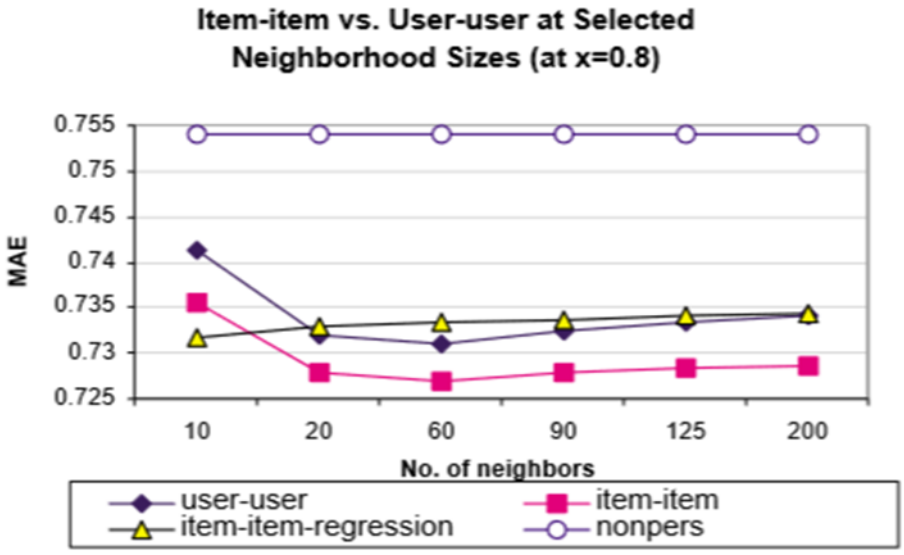

Gráfico 18. Item-item vs. User-user at Selected Neighborhood Sizes (at x=0.8).Fuente: Collaborative Filtering Recommendation Algorithms

Si se centra la atención en user-user e item-item se aprecia claramente como los mejores resultados, vienen dados por el enfoque basado en items. Así mismo, los valores están en consonancia con los arrojados en el estudio presente, anteriormente.

Por tanto se concluye, que en este escenario, en todos los casos, el algoritmo con mejores resultados es KNN-item.

2.4.3. Propuesta de mejora

Se seguirán exactamente los mismo pasos que en el caso anterior, pero separando los datos en Cluster 1 y Cluster 2, como bien indicaba el gráfico 15. Se destaca que los datos que componen entrenamiento en el escenario base, surgen de hacer una división aleatoria en el Cluster1, otra en el Cluster 2 y unirlos. Es decir, la idea que se pretende recalcar es que en los dos escenarios, los datos con los que se trabaja en entrenamiento y test son los mismos, sólo que en este no se trabajan globalmente, y por supuesto para cada uno de los 5 casos la división es aleatoria. El objetivo de ello es poder comparar finalmente, los resultados de ambos escenarios.

Centrando la atención en KNN y en el Cluster 1, que se compone de 3 usuarios y 1.857 observaciones, los resultados obtenidos son los siguientes:

	KNN-Item valor de k	KNN-Item MAE	KNN-User valor de k	KNN-User MAE
CASO 1	3	1.2657	3	**0.9058**
CASO 2	3	1.2038	3	**0.9250**
CASO 3	3	1.2093	3	**0.8925**
CASO 4	3	1.1993	3	**0.8887**
CASO 5	3	1.2024	3	**0.8912**

Tabla 14. Resultados Cluster 1 para KNN-item y KNN-user. Fuente: Elaboración propia

Se muestran los mismos datos para el Cluster 2, compuesto por 940 usuarios y 98.143 ratings:

	KNN-Item valor de k	KNN-Item MAE	KNN-User valor de k	KNN-User MAE
CASO 1	30	**0.7116**	40	0.7275
CASO 2	20	**0.7141**	30	0.7293
CASO 3	30	**0.7081**	30	0.7253
CASO 4	40	**0.7196**	40	0.7292
CASO 5	30	**0.7116**	30	0.7282

Tabla 15. Resultados Cluster 2 para KNN-item y KNN-user. Fuente: Elaboración propia

Como se observa en el Cluster1, KNN-user obtiene mejores resultados que KNN-item. Esto no es de extrañar si suponemos que dichos usuarios tenían perfiles muy distintos al resto de usuarios pero muy parecidos entre ellos según se observó en la estadística descriptiva. Por tanto, son usuarios similares que contienen información útil para

predecir ratings del resto dentro del mismo cluster. Es un cluster muy específico y con muy pocos usuarios.

A continuación se procede a mostrar los resultados para el algoritmo SlopeOne del Cluster 1 y Cluster 2 respectivamente:

	Cluster1 SlopeOne MAE	**Cluster2 SlopeOne MAE**
CASO 1	1.1161	0.7396
CASO 2	1.1135	0.7384
CASO 3	1.0839	0.7316
CASO 4	1.0747	0.7414
CASO 5	1.0168	0.7367

Tabla 16. Resultados Propuesta de mejora para SlopeOne. Fuente: Elaboración propia

Por último se muestran los resultados para ambos clusters y para el algoritmo SVD:

	Cluster 1 Factores	Cluster 1 SVD MAE	Cluster 2 Factores	Cluster 2 SVD MAE
CASO 1	50	0.7408	20	0.7371
CASO 2	50	0.7684	30	0.7371
CASO 3	10	0.7624	40	0.7253
CASO 4	50	0.8028	30	0.7380
CASO 5	30	0.7607	70	0.7297

Tabla 17. Resultados Propuesta de mejora SVD. Fuente: Elaboración propia

Con lo cual para el caso del Cluster 1 la técnica que menor error genera es SVD, mientras que en el Cluster 2 los datos se predicen mejor con KNN-item. A continuación, se procede a fusionar las predicciones derivadas de ambas técnicas en un único set de datos y a calcular el MAE de dicho conjunto. Así mismo se recuerdan los resultados obtenidos en ambos clusters:

	Cluster 1 SVD	Cluster2 KNN-Item	Set Global
CASO 1	0.7408	0.7116	**0.7121**
CASO 2	0.7684	0.7141	**0.7151**
CASO 3	0.7624	0.7081	**0.7091**
CASO 4	0.8028	0.7196	**0.7211**

CASO 5	0.7607	0.7116	**0.7124**

Tabla 18. Resultados finales Propuesta de mejora. Fuente: Elaboración propia

2.5. Resultados

Centrando la atención en el escenario base, en los 5 casos propuestos los mejores resultados, en términos de MAE, vienen dados por KNN-item, por tanto las predicciones derivadas de dicha técnica serán las empleadas en el escenario base. Con respecto a la propuesta de mejora, para el Cluster 1 el vencedor siempre es SVD, mientras que para el Cluster 2 lo es KNN-item. Por tanto, la propuesta de mejora será la fusión de las predicciones derivadas de dichas técnicas. Se procede a contrastar resultados de la propuesta de mejora con respecto al escenario base:

	Escenario base MAE	Media base MAE	Propuesta de mejora MAE	Media mejora MAE
CASO 1	0.7168	0.9484	**0.7121**	0.9330
CASO 2	0.7183	0.9422	**0.7151**	0.9272
CASO 3	0.7119	0.9374	**0.7091**	0.9230
CASO 4	0.7224	0.9474	**0.7211**	0.9362
CASO 5	0.7167	0.9436	**0.7124**	0.9286

Tabla 19. Comparación Escenario base VS Propuesta de mejora. Fuente: Elaboración propia

Como se observa, en los 5 casos los resultados de la propuesta de mejora introducen un pequeño beneficio en términos de MAE con respecto al escenario base. Además se han incluido los resultados derivados de aplicar como predictor simplista la media en cada uno de los dos escenarios. Para el escenario base se imputa la media de los datos de entrenamiento a todos los usuarios y se obtiene el error medio en test (Media base MAE). De la misma manera, para la propuesta de mejora, en el Cluster 1 se aplica la media del Cluster1 en entrenamiento para cada individuo y en el Cluster 2 su media también en entrenamiento. Posteriormente, se fusionan ambos clusters y se obtiene el error medio global en test (Media mejora MAE). Como conclusión destaca que el tratamiento separado de individuos en dos clusters mejora también el predictor media en términos de MAE, y que si se compara la media con los predictores seleccionados en cada escenario, sin duda las técnicas aplicadas mejoran la predicción con respecto a ella, que si bien es simple en ocasiones es difícil de derrotar. Se concluye además que como se quería comprobar el resultado es independiente de los datos englobados en test y

entrenamiento. Por tanto, aclarado lo anterior se procede a estudiar más detenidamente sólo uno de los casos, concretamente el Caso 5.

Recopilando para el caso 5, se recuerdan los resultados del escenario base en términos de MAE conjuntamente con la propuesta de mejora.

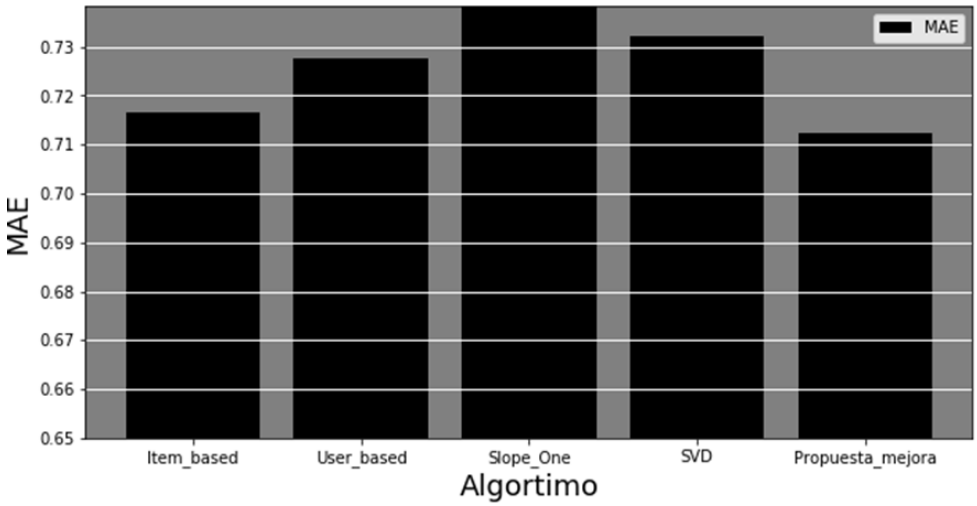

Gráfico 19. Propuesta de mejora VS Escenario Base .Fuente: Elaboración propia.

Se observa como la opción más beneficiosa en el escenario base es KNN- item based mientras que la peor es SlopeOne. La propuesta de mejora, combinando KNN-item y SVD para clusters de usuarios definidos, logra mejorar el resto de opciones. No obstante, los resultados que arrojan los algoritmos, no están expresados en números enteros, por lo que habría que realizar un redondeo del rating predicho. Haciendo esto último y centrando la atención sólo en el mejor resultado para el escenario base y la propuesta de mejora, las diferencias se reducen mucho, pero aun así la propuesta de mejora es ínfimamente mejor.

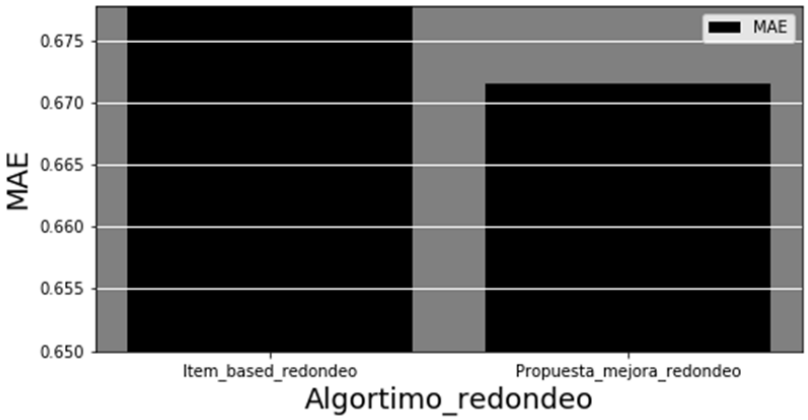

Gráfico 20. Propuesta de mejora con redondeo VS Escenario base con redondeo. Fuente: Elaboración propia.

Por último, se expone comparativamente en un gráfico la distribución de los errores para ambos algoritmos. En el eje x se muestran las estrellas de desviación de la predicción con respecto a los datos reales y en el eje y el % de predicciones en las que se comete ese error.

Gráfico 21. Distribución de los errores en estrellas Propuesta de mejora VS Escenario base. Fuente: Elaboración propia.

Como se puede observar ambos algoritmos, con las predicciones ya redondeadas, muestran la misma distribución, es decir, el 90% de las predicciones cometen como máximo 1 punto de error. El caso en el que el error es pleno es decir, de 4 puntos solo representa un 0.05% aproximadamente de las predicciones. Aun así la diferencia entre ambos algoritmos es que la propuesta de mejora comete más errores de 1 punto y más aciertos. En cuanto a los errores de 2 y 3 puntos, el escenario base los comete en un mayor porcentaje. No obstante cometen el mismo porcentaje de errores de 4 puntos, y además es mínimo.

Finalmente una vez obtenidas las predicciones, que derivan de la propuesta de mejora el último paso sería simplemente filtrar las predicciones y obtener solo aquellas que superan cierto umbral[17]. En este caso, todas aquellas predicciones que superan el 3.5, y por tanto que estimamos que van a ser de 4 o 5 estrellas serán recomendadas. La tabla final de recomendaciones quedaría como sigue:

[17] Es un umbral subjetivo, que puede variar en función de las necesidades.

	user	item	rating_estimado
0	308	11	4.0
1	676	172	5.0
2	350	183	4.0
6	479	182	4.0
10	426	968	4.0
11	764	121	4.0
13	328	51	4.0
14	642	168	4.0
15	59	132	4.0
16	796	525	4.0
17	649	127	4.0
18	398	82	4.0

Tabla 20. Estructura predicciones. Fuente: Elaboración propia

Como refleja la tabla inmediatamente anterior, al usuario 308 se le recomendaría la película 11, puesto que para ella el algoritmo ha predicho un 4. No obstante, aquí tendríamos usuarios para los que no recomendaríamos nada por falta de datos iniciales o desconocimiento de sus gustos, el conocido problema del arranque en frío. Una manera de corregirlo podría ser complementar el sistema de recomendación de filtrado colaborativo con un sistema de recomendación basado en contenido, actuando este último en aquellos casos donde la información proporcionada por el usuario fuese ínfima o insuficiente.

3. Nueva propuesta.

En vista a los resultados obtenidos en el punto 2, parece que podría tener sentido que la predicción final fuera una combinación de diversas técnicas. Esto se deduce puesto que la propuesta de mejora, con respecto al escenario base, simplemente le ha dado cierto peso a SVD dentro del set de datos y se han combinado KNN-item y SVD.

La idea que se persigue es que cada técnica encuentre su peso en la predicción final, de manera que el rating estimado se estructure de la siguiente manera:

$$P_{FINAL} = \alpha_1 * P_{KNN-ITEM} + \alpha_2 * P_{KNN-USER} + \alpha_3 * P_{SLOPEONE} + \alpha_4 * P_{SVD}$$

Se define P como la matriz de dimensión 20.000x1 que contiene los ratings reales del set de test para cada individuo, Ω como la matriz de dimensión 20.000*4 que contiene las predicciones de KNN-item, KNN-user, SlopeOne y SVD respectivamente en cada fila para cada una de las 20.000 observaciones y α como la matriz de dimensión 4x1 que engloba los coeficientes α_i que ponderan el peso de las predicciones en el resultado final. Definidas dichas matrices el problema a resolver es el siguiente:

$$Min_{\alpha} \ \|P - \Omega\alpha\|_2^2$$

Por tanto, aplicando el estimador MCO se desvelaría el valor de la matriz α de la siguiente manera:

$$\alpha^* = (\Omega'\Omega)^{-1}\Omega'P$$

Con la ayuda del software Python se resuelve el problema de optimización, obteniendo los siguientes valores para los coeficientes:

COEFICIENTE	VALOR
$\boldsymbol{\alpha_1}$	*0.523160*
$\boldsymbol{\alpha_2}$	0.238455
$\boldsymbol{\alpha_3}$	-0.091457
$\boldsymbol{\alpha_4}$	0.323497

Tabla 21. Propuesta de mejora 2: coeficientes. Fuente: Elaboración propia.

Los resultados dan el peso principal en la estimación a KNN-item seguido de SVD, lo cual era esperado puesto que indirectamente así lo indicaban los resultados obtenidos en la propuesta de mejora del apartado 2. Se hará referencia a las estimaciones derivadas de la resolución de este problema con el nombre de propuesta de mejora 2.

El siguiente paso a dar para obtener la propuesta de mejora 3, complica el problema de optimización, puesto que requiere añadir dos restricciones: que todos los coeficientes sean mayores que 0, cosa que no ocurre con α_3 en la propuesta de mejora 2 y que la suma de todos ellos sea igual a la unidad. Con la ayuda de Python y de la librería Scipy se da solución al problema anterior, contemplando dichas restricciones de igualdad y desigualdad, obteniendo los siguientes resultados:

COEFICIENTE	VALOR
α_1	0.4949002
α_2	0.21867639
α_3	0.00000000
α_4	0.28074687

Tabla 22. Propuesta de mejora 3: coeficientes. Fuente: Elaboración propia.

Se proceder a comparar estas dos propuestas de mejora con el escenario base y la propuesta de mejora del apartado inmediatamente anterior. Los resultados brutos sin redondear indican los siguientes valores de MAE:

Escenario analizado	MAE
Escenario base	0.716687
Propuesta de mejora	0.712465
Propuesta de mejora 2	0.714560
Propuesta de mejora 3	0.714733

Tabla 23. Comparación escenarios brutos. Fuente: Elaboración propia

Los mejores resultados son los que se obtienen de tratar los dos cluster identificados por separado y dejar que se adapten a técnicas distintas, es decir, de la propuesta de mejora. Si se procede a redondear los resultados de las predicciones obtenidas por el problema de minimización para que sean discretos, y se comparan con las del escenario base y la propuesta de mejora, redondeadas también como ya se trataron en puntos anteriores, se obtiene lo siguiente:

Escenario analizado	MAE
Escenario base	0.677766
Propuesta de mejora	0.671566
Propuesta de mejora 2	0.672916
Propuesta de mejora 3	0.672066

Tabla 24. Comparación escenarios redondeados. Fuente: Elaboración propia

En este caso los mejores resultados continúan siendo aportados por la propuesta de mejora, aunque se destaca que la discrepancia es mínima. No obstante, destaca que esto

no ocurre con carácter general. Lo que si ocurre con carácter general es que las propuestas de mejora, reducen MAE con respecto al escenario base. Gráficamente:

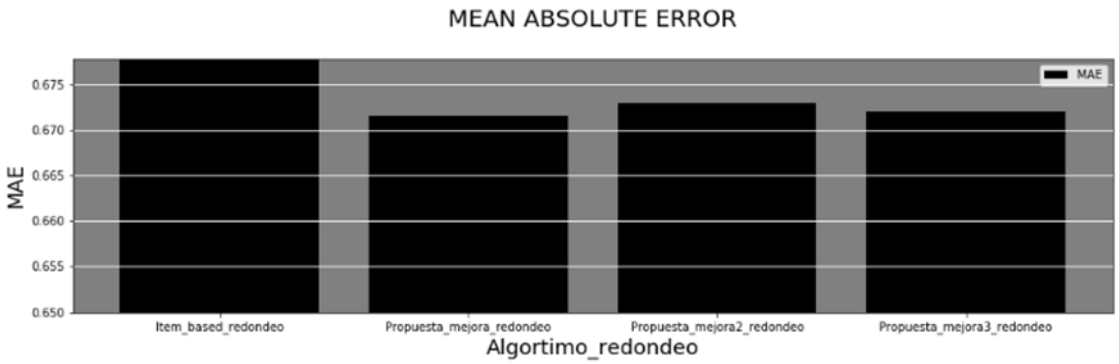

Gráfico 22. Comparación Escenario base VS Propuestas de mejora

4. Conclusiones.

Como conclusión se expone la capacidad de mejora de las técnicas de recomendación basada en una buena compresión de los datos. Gracias al análisis inicial se definen una serie de atributos, que se corroboran en base a un modelo multinomial ordenado, y con la ayuda de las técnicas PCA y Hierarchical clustering, se segmentan dos tipologías extremas de usuarios. Esa segmentación permite que cada grupo de usuarios se adapte a técnicas distintas, concretamente KNN-item y SVD. La idea detrás de todo ello es que ciertos usuarios con características extremas pueden expresar gustos distintos pese a un mismo comportamiento en términos de ratings. Por ello, utilizar todos los usuarios conjuntamente en las predicciones globales puede introducir "ruido" que distorsione la realidad. Tras indicar los resultados, que efectivamente la recomendación mejoraba en términos de MAE al considerar los dos clusterS por separado, aplicar técnicas distintas y fusionar finalmente las predicciones, en lugar de aplicar directamente las técnicas al conjunto global de datos, se intenta materializar la idea de que cada técnica tenga un peso en la predicción. Para ello, se propone una regresión simple del rating real con respecto a los ratings estimados en el escenario base para cada una de las técnicas propuestas: KNN-item, KNN-user, SlopeOne y SVD. Los resultados muestran que está visión también mejora el escenario base. Así mismo, la regresión tal como se esperaba vuelca pesos altos en KNN-item y SVD lo cual concuerda con la evolución seguida por la propuesta de mejora inicial.

Como vía de investigación futura se plantea el estudio del problema del arranque en frío, complementando posiblemente el sistema de recomendación de filtrado colaborativo con un sistema de recomendación basado en contenido, para focalizar este último en aquellos casos en los que se haya recogido poca información de usuarios. Así mismo, sería interesante profundizar en otras técnicas de clustering, para ver si otras posibles segmentaciones, no tan extremas, producirían mejoras en el escenario base.

5. Referencias Bibliográficas

INE. (2017, 5 octubre). Notas de prensa [Comunicado de prensa]. Recuperado 25 junio, 2018, de http://www.ine.es/prensa/tich_2017.pdf

Ricci, P. B., Rokach,F., Shapira, B., & Kantor, L.(2011). Recommender Systems Handbook.

M. D., Riedl, J. T., & Konstan, J. A. (2011). Collaborative filtering recommender systems. Foundations and Trends® in Human–Computer Interaction, 4(2), 81-173.

Lemire, D., & Maclachlan, A. (2005, April). Slope one predictors for online rating-based collaborative filtering. In Proceedings of the 2005 SIAM International Conference on Data Mining (pp. 471-475). Society for Industrial and Applied Mathematics.

Koren, Y., & Sill, J. (2013, August). Collaborative Filtering on Ordinal User Feedback. In IJCAI (pp. 3022-3026).

Donate, M. C. R., & Hernández, J. J. C. (2007). Modelos de elección discreta y especificaciones ordenadas: una reflexión metodológica. Universidad de La Laguna, Facultad de Ciencias Económicas y Empresariales, San Cristóbal de la Laguna.

Jolliffe, I. (2011). Principal component analysis. In International encyclopedia of statistical science. Springer, Berlin, Heidelberg.

Gallardo, J. (2011). Métodos Jerárquicos de Análisis Clúster. Disponible desde Internet en: www. ugr. es/~ gallardo/pdf/cluster-3. pdf.(Con acceso el 28/08/2018)

You, H., Li, H., Wang, Y., & Zhao, Q. (2015, March). An improved collaborative filtering recommendation algorithm combining item clustering and slope one scheme. In Proceedings of the International MultiConference of Engineers and Computer Scientists (Vol. 1, pp. 313-316).

Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2001, April). Item-based collaborative filtering recommendation algorithms. In Proceedings of the 10th international conference on World Wide Web (pp. 285-295). ACM.

Samsung Display. (2017, Jan). Top 5 Digital Signage Trends for 2018. Recuperado de https://pid.samsungdisplay.com/en/learning-center/blog/top-5-digital-signage-trends-2018