UNIVERSIDAD TECNOLÓGICA DE LA MIXTECA

Modelo recomendador para plataforma de streaming basado en ratings y en métricas modificadas de Hausdorff

TESIS
PARA OBTENER EL TÍTULO DE:

Licenciado en Matemáticas Aplicadas

PRESENTA:

Hael Bravo Ramírez

DIRECTOR DE TESIS:

Dr. Tomás Pérez Becerra

CO-DIRECTOR DE TESIS:

Dr. Saúl Solorio Fernández

HUAJUAPAN DE LEÓN, OAXACA.
—Mayo del 2025—

Para Guillermina,
mi madre.

Agradecimientos

En primer lugar, quiero agradecer a Dios por cada momento, tanto los buenos como los malos, que han sido parte de mi camino y han hecho posible este logro.

A mi madre, por darme la vida y sacrificar la suya por mí; por enseñarme que hay belleza incluso en el dolor, y que el amor trasciende el tiempo y el espacio. Espero, de corazón, que se sienta orgullosa de sus hijos, al menos en una medida cercana en la que lo estamos de usted.

A mi hermana Vania, por acompañarme incondicionalmente a lo largo de mi vida, incluso cuando la distancia nos separó. Quiero que sepa cuánto la amo y que siempre podrá contar conmigo.

A mis amigos de la universidad —Mendiola, Raymundo, Luz y Yulisa—, así como a todos mis compañeros de salón de clases, por llevar nuestra amistad más allá de las aulas y hacer que este recorrido fuera más llevadero y significativo.

A Danna, por creer en mí incluso cuando yo dudaba, por escucharme, acompañarme y apoyarme en cada paso.

A María y a Kelly, por tenerme paciencia e iluminar este camino con su alegría contagiosa. Con su apoyo incondicional, este logro es más dulce.

A mi director de tesis, el Dr. Tomás, por confiar en mi capacidad para llevar a cabo esta investigación, por su guía paciente y dedicada durante todo el proyecto, y por su disponibilidad constante para ayudarme a crecer como profesional e investigador.

A mi codirector, el Dr. Saúl, por su invaluable aporte, sus consejos certeros y su compromiso con el rigor académico, que sin duda enriquecieron profundamente este trabajo.

A mis sinodales, por aceptar evaluarme, y cuyas observaciones, preguntas críticas y retroalimentación durante mi examen profesional ampliaron mi perspectiva y fortalecieron este trabajo. Agradezco su tiempo, su rigor y su compromiso con la excelencia académica.

A la Universidad Tecnológica de la Mixteca, en especial al Instituto de Física y Matemáticas, por brindarme una formación sólida y las herramientas necesarias para enfrentar desafíos intelectuales con determinación, y sobre todo, por despertar en mí la pasión por las Matemáticas.

A los profesores de esta institución, cuyo profesionalismo, dedicación y pasión por la enseñanza han dejado una huella imborrable en mi formación. Su ejemplo ha sido fundamental no solo en mi crecimiento académico, sino también en mi visión del conocimiento como un compromiso con la excelencia y la sociedad.

Prefacio

El auge de la ciencia de datos en las últimas décadas no es un fenómeno aislado, sino el resultado de una convergencia histórica entre la estadística, la computación y las necesidades de la era digital. Sus raíces pueden rastrearse hasta los primeros sistemas expertos de los años 70, pero fue con el surgimiento del Big Data y el aprendizaje profundo que adquirió un papel fundamental en la toma de decisiones automatizada, en busca de resolver problemas cada vez más complejos en nuestra sociedad.

En la era digital, el acceso a información y entretenimiento ha crecido de manera exponencial, dando lugar a desafíos en la organización, selección y personalización del contenido. Las plataformas de streaming han revolucionado la forma en que los usuarios consumen películas, series, música y otros medios de entretenimiento digital, generando una necesidad creciente de sistemas inteligentes que faciliten la recomendación de contenido relevante. En este contexto, los modelos recomendadores han surgido como una de las aplicaciones más impactantes de la ciencia de datos.

Los sistemas de recomendación nacieron en la década de los 90, con proyectos pioneros como Tapestry (Xerox PARC) [Rawn, 2024] y también GroupLens (Universidad de Minnesota) [Resnick et al., 1994], que introdujeron el filtrado colaborativo basado en *ratings*. Hoy, se han convertido en el corazón de plataformas como Netflix, Spotify o Amazon, donde generan billones de dólares en valor, determinando no solo qué contenidos se promueven, sino también la experiencia personalizada de millones de usuarios. Sin embargo, la calidad de estas recomendaciones depende críticamente de cómo se midan las similitudes entre perfiles de usuarios y cómo se manejen datos imperfectos, como ratings dispersos o sesgados.

Este trabajo aborda ese desafío mediante un modelo recomendador innovador que combina técnicas de filtrado colaborativo con una adaptación de las métricas modificadas de Hausdorff, estas métricas han sido diseñadas para capturar relaciones de proximidad más robustas en espacios de preferencias heterogéneas. La métrica de Hausdorff, utilizada originalmente en geometría para medir la distancia entre conjuntos, se ha adaptado en este contexto para mejorar la precisión y relevancia de las recomendaciones. Al modificar esta métrica, se busca capturar mejor las similitudes entre los perfiles de usuario y los patrones de preferencia, proporcionando una experiencia de recomendación más personalizada y efectiva.

La motivación para este trabajo surgió al identificar que muchas plataformas dependen de algoritmos que no suelen ser *robustos* a la dispersión natural de los *ratings* (como usuarios que

califican pocos ítems o lo hacen de manera extrema). La métrica propuesta aquí busca resolver ese vacío, ofreciendo una alternativa escalable y matemáticamente fundamentada. Así mismo, el propósito de este estudio es explorar y evaluar el impacto de estas modificaciones en la calidad de las recomendaciones, comparando su desempeño con modelos convencionales y estableciendo un marco metodológico que pueda ser aplicado a diferentes plataformas de contenido digital, esperando contribuir al avance de los sistemas recomendadores y ofrecer nuevas perspectivas sobre el uso de métricas geométricas en el análisis de preferencias de los usuarios.

Se le invita al lector a adentrarse en este estudio con la misma curiosidad y entusiasmo con los que fue concebido, esperando que los hallazgos sean de utilidad para quien desee adentrarse en este apasionante tema.

Índice general

Índice de figuras

Lista de tablas

Introducción

La Ciencia de Datos es un campo multidisciplinario que se enfoca en extraer información valiosa y conocimientos a partir de grandes cantidades de datos. Combina diversas disciplinas como matemáticas, estadística, programación y análisis de datos para identificar patrones, hacer predicciones y tomar decisiones informadas.

La investigación tratada en esta tesis se enmarca dentro de la Ciencia de Datos Aplicada. Esta área ha cobrado gran relevancia durante los últimos años, ayudando a resolver de forma rápida y confiable problemas que tienen como materia prima grandes cantidades de información. En el ámbito empresarial, por ejemplo, ha sido ampliamente utilizada con fines de logistica y mercadotecnia; se ha implementado tambien en las redes sociales como herramientas de segmentación de clientes; así como para plataformas de transmisión de series y películas (*streaming*) y demás aplicaciones para dispositivos móviles y páginas web.

La presente investigación se enfoca en los modelos recomendadores para plataformas de streaming. Escencialmente, un *modelo recomendador* es un sistema automático que muestra productos o servicios al usuario de una aplicación con el objetivo de que los consuma. Puede estar basado en búsquedas realizadas por el usuario, en compras de distintos productos, en gustos y preferencias expresados, en locaciones o ubicaciones visitadas, en interacciones en redes sociales, entre otros. El funcionamiento de dichos modelos consiste en realizar tres tipos de filtrados de datos:

- *Basados en contenido.* Los cuales crean predicciones sobre atributos que pueden ser comparados entre artículos directamente. El sistema ofrece recomendaciones basadas en las preferencias específicas de cada usuario. Una de las frases que más caracteriza a este tipo de filtrado es “Algunos productos similares al que viste son...”.

- *Filtrados colaborativos.* Utilizan datos de comportamiento de los usuarios y artículos en la plataforma para determinar si a un usuario le gustará cierto artículo. Comparan sus preferencias con las de otros usuarios, y posteriormente, generan las recomendaciones. Por ejemplo, si un usuario observa un artículo en un sitio web, se le recomendarán productos que otros usuarios compraron al buscar dicho artículo. Una frase comúnmente utilizada en estos sistemas es “Personas que buscaron este artículo también compraron...”.

- *Métodos híbridos.* Combina los sistemas anteriores. El filtrado basado en contenido funciona bien para las sugerencias explícitas de los intereses de un usuario. Sin embargo, no pueden

predecir con precisión recomendaciones fuera de dichas preferencias documentadas. En un sistema de filtrado híbrido, este déficit se cubre mediante filtrado colaborativo, ya que es capaz de sugerir productos relacionados que no se encuentran dentro del perfil establecido de un usuario al basar las recomendaciones en las preferencias y comportamientos de una cohorte similar. Alternativamente, el filtrado basado en contenido ayuda a llenar los vacíos creados por los sistemas colaborativos. Si no existen datos comparativos para grupos similares, el recomendador buscará de forma predeterminada una coincidencia basada en etiquetas de atributos para encontrar un resultado adecuado.

En esta tesis, se analizará un método de filtrado colaborativo, basado específicamente en *ratings*, (calificaciones, puntajes o valoraciones). Un *rating* es una calificación que le otorga el usuario a un producto o servicio a manera de medición de la calidad de este; en la actualidad, en aplicaciones como Uber, Google Maps, Amazon, Facebook y YouTube, entre otras, se solicita al usuario que asigne un puntaje, este puede ser un valor discreto (por ejemplo, enteros entre 0 y 5), binario ("Me gusta"-"No me gusta") o incluso continuo en un intervalo fijo. Los ratings asignados a cada producto o servicio, que de ahora en adelante llamaremos *ítem*, ofertado en las plataformas digitales, son recopilados constantemente, con lo que se produce una cantidad enorme de información o *data* en intervalos de tiempo cada vez más cortos, esto permite que los recomendadores se ajusten continuamente a las preferencias de los usuarios; no obstante, supone retos cada vez mayores para el tratamiento y uso de la información que se genera. Estos sistemas utilizan los *ratings* para determinar tres tipos de similitudes:

- Entre productos, denotados como *Ítem-Ítem (articulo-articulo)*. Cuando un usuario elige un ítem, se le recomienda un conjunto de ítems similares al elegido inicialmente, por ello, se tiene la necesidad de determinar cuan similar es uno de ellos con el resto.
- Determinar el "gusto" de un producto al usuario, son denominados *Ítem-User (usuario-artículo)*.
- Determinar la similitud entre dos usuarios para poder mostrar productos al usuario que el otro usuario similar a él eligió. Este sistema es llamado *User-User (usuario-usuario)*.

El procesamiento de los ratings, mediante técnicas de minería de datos, da lugar a una *Matriz de utilidad*, donde cada una de sus entradas se compone de la calificación que el usuario asigna acada producto. Esta matriz es de mucha ayuda para poder organizar la información registrada en las bases de datos de las empresas. Sin embargo, dada su propia naturaleza, representa también un reto el poder trabajar con ella de forma óptima, aunque existen procesos de transformación y mejoramiento de la matriz de utilidad que sirven para aumentar su usabilidad y su fiabilidad (véase la sección 1.2, 2.2).

Por otro lado, los modelos matemáticos que dan origen a los sistemas recomendadores se conocen como *motores de recomendación*. Dichos motores pueden ser de diversos tipos; ya sea basados en factorización matricial de la matriz de utilidad o en modelos de aprendizaje máquina

no supervisados de agrupamiento, entre otros. Para el caso de este proyecto, se han analizado estos últimos. Dichos modelos son dependientes del tipo de ítem. Por ejemplo, si lo que se busca recomendar son destinos turísticos, entonces el motor de recomendación será distinto a si se desea recomendar automóviles. Para delimitar el trabajo de esta tesis, se planteará un modelo para recomendadores de películas de *streaming* basado en ratings. Este tipo de sistemas está basado en determinar la matriz de utilidad para luego aplicar un modelo que determine la similitud deseada: ítem-ítem, ítem-user o user-user, El presente proyecto se enfoca en los de tipo ítem-ítem.

Para lograr determinar la similitud entre los ítems y realizar la recomendación ítem-ítem se debe plantear la "cercanía" o el "parecido" entre ellos. Una forma de determinar esta cercanía o parecido es a través de los *ratings* o calificaciones, los cuales son capaces de proporcionar una manera de medir las preferencias de cada usuario y fijar un nivel de similitud. Por ejemplo, analizando patrones de calificación hacia los ítems. Para ello, se consideran estos registros de las calificaciones como vectores sobre un espacio vectorial y se dota de una métrica para instaurar vecindades de grupos "parecidos", lo que conduce a considerar modelos basados en k-vecindades cercanas (*KNN* por sus siglas en inglés). Usualmente, los modelos *KNN* utilizan distintas métricas para construir las vecindades, como la euclidiana, de Minkowsky, L_p, de Hausdorff, entre otras.

Una problemática frecuente al realizar este tipo de mediciones es la sensibilidad de los modelos a los datos atípicos, es decir, aquellos valores que difieren significativamente del resto de las observaciones. Por ejemplo, la presencia de un solo dato que no concuerde con los demás puede afectar la medida de similitud, llevando a concluir erróneamente que dos ítems son distintos cuando en realidad comparten muchas características (véase [Wang and Tan, 2012]). En este contexto, se dice que los modelos no son robustos (véase la sección 1.5).

En [Wang and Tan, 2012], este problema se aborda en el contexto de sistemas de reconocimiento de imágenes mediante la implementación de una métrica modificada de Hausdorff. Esta modificación permite robustecer el modelo al introducir variantes del algoritmo de k vecinos cercanos basadas en dicha métrica, logrando resistencia a datos atípicos (como imágenes con altos niveles de ruido) en dos bases de datos distintas.

Sin embargo, con base en el análisis bibliográfico realizado, se identifica que actualmente no existen modelos de sistemas recomendadores que empleen este tipo de métricas modificadas.

En esta ocasión, es de interés comprobar si se puede robustecer un sistema recomendador basado en ratings utilizando las métricas utilizadas en [Wang and Tan, 2012], específicamente para películas, ya que se cuenta con una base de datos liberada por la compañía *Netflix*, y esta base cuenta con datos atípicos.

Esta tesis resolverá parcialmente esta cuestión al proporcionar un modelo matemático específico para un sistema recomendador de películas basado en ratings y en k-vecindades cercanas. Se plantea además que implemente dichas métricas y validarlo con una muestra de la base con la que se cuenta, lo que la hace ideal para el proyecto. El problema general sobre la demostra-

ción de que un sistema de este tipo es robusto para cualquier base no será abordado en esta investigación.

Basado en la problemática antes descrita, el objetivo general de esta investigación, es crear un modelo matemático para un sistema recomendador robusto para películas basado en k-vecindades más cercanas y métricas de Hausdorff modificadas, además de realizar su simulación y validarlo. Los objetivos específicos son:

1. Diseñar el modelo y algoritmo de k-vecindades más cercanas para un sistema recomendador basado en ratings, utilizando la base de datos de Netflix.

2. Implementar diversas métricas modificadas de Hausdorff al modelo y realizar las simulaciones mediante la programación en Python en un Notebook de Google Collaboratory, basado en el lenguaje Python.

3. Validar el modelo y determinar su robustez en la base de datos particular propuesta.

Para lograr los objetivos específicos, y con ello alcanzar el general, esta tesis se distribuye de la siguiente manera:

Capítulo 1: Preliminares Se presenta una revisión teórica de los conceptos fundamentales que sustentan esta investigación, lo que se busca en este capítulo, es proporcionar al lector el contexto necesario para comprender el desarrollo y alcance de este trabajo.

Capítulo 2: Preprocesamiento de la base de datos Se lleva a cabo un análisis exploratorio de la base de datos con la que se cuenta para esta investigación, incluyendo la verificación de las distribuciones, detección de datos atípicos (aspecto crítico para la fase experimental) y la construcción de la matriz de utilidad. Adicionalmente, también se realizan adaptaciones a esta matriz para facilitar su procesamiento en la etapa experimental y contrastar los resultados.

Capítulo 3: Modelos recomendadores Se implementan los modelos de recomendación basados en las variantes de las métricas de Hausdorff y en las modificaciones aplicadas a la matriz del Capítulo 2; se evalúan mediante simulaciones y se compara el desempeño de cada uno. Finalmente, se busca determinar cuál de todos los métodos es el que mejor comportamiento tiene.

Se espera que los resultados obtenidos en esta investigación no solo validen la metodología propuesta, sino que también contribuyan al avance del conocimiento en el área de los sistemas de recomendación. En particular, se aspira a que este trabajo sirva como punto de partida para nuevas líneas de investigación, al ofrecer enfoques alternativos y soluciones potenciales a problemáticas reales y actuales del campo. Asimismo, se confía en que los hallazgos aquí presentados resulten de interés y utilidad para investigadores, desarrolladores y cualquier lector interesado en la temática. Con este propósito, a continuación se presenta el desarrollo de esta investigación.

Capítulo 1

Preliminares

Los sistemas recomendadores basados en ratings se componen de una serie de ecuaciones y procesos matemáticos, por lo que es indispensable iniciar con la introducción de algunos conceptos matemáticos y resultados previos, por esta razón, en este capítulo se muestran las definiciones de los modelos recomendadores, la definición y construcción de la matriz de utilidad de forma generalizada y sus propiedades, los conceptos de similitud y distancias (desde el punto de vista matemático), el modelo de k vecinos cercanos y las métricas de Hausdorff. Cabe mencionar que una motivación de esta investigación fue el tratamiento de datos atípicos, por lo que también se abordará en este capítulo preliminar.

1.1. Sistemas recomendadores

Los sistemas recomendadores son sistemas o técnicas que sugieren o recomiendan un producto, un servicio o una entidad a un usuario en particular. Estos sistemas son clasificados en dos categorías basadas en la manera en la que proveen recomendaciones, y su diseño tiene como base la Minería de Datos y los motores de recomendación.

Un *motor de recomendación* es un conjunto de herramientas y técnicas utilizadas para analizar grandes volúmenes de datos utilizando la información de productos y de usuarios. El objetivo principal de un sistema recomendador es proporcionar sugerencias relevantes a usuarios en línea para tomar las mejores decisiones entre un conjunto de artículos disponibles. Estos sistemas realizan las recomendaciones utilizando las *huellas* dejadas por el usuario dentro o fuera de la plataforma, ya sea su información demográfica, reacciones a publicaciones, o las interacciones con productos, tales como especificaciones y gustos (véase [Gorakala, 2016]).

Técnicamente, un problema de motor de recomendación consiste en desarrollar un modelo matemático o función objetivo, el cual pueda predecir cuánto le gusta un producto a un usuario. Si $U = \{Usuarios\}$ e $I = \{Items\}$, entonces el modelo matemático buscado es una función:

$$F : U \times I \to R,$$

que mide la utilidad del producto I al usuario U, donde $R = \{productos\ recomendados\}$. En específico, para cada $u \in U$, se desea elegir el ítem $i \in I$ que optimice la función objetivo F tal que:

$$I_u = \arg\max_u F(u, i).$$

Construir un buen motor de recomendación plantea retos a los dos actores del sistema: los consumidores y los vendedores. Para el consumidor es fundamental recibir sugerencias relevantes de una fuente confiable para llevar a cabo la toma de decisiones. Por eso, el motor de recomendación debe ser construido de tal forma que se gane la confianza de los consumidores ofreciendo "buenas" recomendaciones. Por otra parte, para el vendedor es más importante generar recomendaciones "relevantes" para los consumidores de forma personalizada.

Con el auge de las ventas en línea, las grandes empresas están recopilando grandes volúmenes de registros de interacciones transaccionales de los usuarios para analizar sus comportamientos con mayor profundidad y precisión. Es aquí donde aparecen los problemas al construir un sistema recomendador, pues existen enormes cantidades de datos surgiendo a cada instante, se recolectan grandes volúmenes de registros de información (llamados *logs* en informática) de usuarios interactuando con productos y servicios a través de las redes. Su análisis en tiempo real es otro factor determinante, ya que la recomendación debe ser ofrecida al momento en que se requiera. Tan sólo en el 2020 se estimaba que había 26,000 millones de dispositivos conectados a internet que generarían, aproximadamente 300,000 millones de datos (véase [Jofra and Gómez, 2019]) y, como se sabe, su crecimiento ha sido exponencial. Estos datos por sí solos no presentan una utilidad directa. Sin embargo, al poder discernir la información relevante de la que no lo es, se pueden convertir en una fuente de conocimiento para la toma de decisiones que optimicen la cadena de suministro (véase [Jofra and Gómez, 2019]). Para lograr pulir estos datos, se deben utilizar herramientas basadas en ciencia de datos, tal como *Big data* y el *Machine Learning*, debido a que con ellas se determina el comportamiento de los usuarios y, así, se ajustan modelos matemáticos adecuados al objetivo final: vender un producto o servicio. Un buen motor de recomendación debe ser fiable, escalable[1], altamente disponible y capaz de ofrecer recomendaciones personalizadas en tiempo real a la gran base de usuarios que contiene.

El papel del *Big Data* y las mejoras tecnológicas, tanto en el ámbito del software como del hardware, va más allá del mero suministro de datos masivos. También proporciona datos significativos y procesables, y proporciona la configuración necesaria para procesar rápidamente los datos en tiempo real. Actualmente, se está avanzando hacia una mayor personalización de aspectos como la dimensión temporal y las formas ubicuas de recomendación. En el aspecto tecnológico, las recomendaciones están pasando de enfoques de aprendizaje automático a enfoques más avanzados de aprendizaje profundo y redes neuronales [Gorakala, 2016].

Como se mencionó en la introducción, existen al menos dos tipos de sistemas recomendadores: por filtrado colaborativo y basados en contenido. En los motores de recomendación del primer tipo, el filtrado de elementos se realiza a partir de un conjunto de preferencias de los mismos

[1]Capacidad de un sistema para adaptarse y crecer ante demandas cambiantes.

usuarios, bajo la hipótesis de que si dos usuarios compartieron los mismos intereses en el pasado, también tendrán gustos similares en el futuro. De esta clase se derivan dos subtipos: el filtrado colaborativo basado en el usuario y el basado en elementos o ítems.

En el filtrado colaborativo basado en usuarios, las recomendaciones se generan teniendo en cuenta las preferencias del usuario y se realizan en dos pasos:

1. Identificar usuarios similares, tomando como referencia las preferencias que comparten.

2. Recomendar nuevos artículos a un usuario activo, basados en la valoración dada por usuarios similares a los artículos no valorados por el usuario activo.

En el caso del filtrado colaborativo basado en ítems, las recomendaciones se generan utilizando una *vecindad* de los artículos. A diferencia del filtrado colaborativo basado en el usuario, primero se encuentran similitudes entre los artículos y, posteriormente, los elementos no valorados que son similares a los que el usuario activo ha valorado en el pasado son ofrecidos al usuario. Los sistemas de recomendación basados en ítems se construyen en dos pasos:

1. Calculan la similitud de los artículos en función de las preferencias del usuario activo.

2. Encuentran los artículos más similares a los no valorados por el usuario activo y luego los recominedan.

Uno de los principales problemas que generan estos sistemas es el del llamado *Cold Start*, que significa que los sistemas de filtrado colaborativo no recomiendan a los usuarios cuya información no esté disponible en el sistema.

Para el caso de los sistemas basados en contenido, sólo se toma en cuenta las preferencias que los usuarios han expresado de forma explícita, y se crean sistemas de recomendación con base en ellas. Aunque este enfoque es preciso, tiene más sentido si se toman en cuenta las propiedades del usuario y las del artículo a la hora de crear motores de recomendación. Otra desventaja a considerar sobre estos sistemas, es que no son capaces de ofrecer recomendaciones que escapan a los gustos expresados por el usuario, lo que dificulta su escalabilidad.

En este contexto, el modelo presentado en esta tesis se considera un híbrido entre las dos clases del filtrado colaborativo, es decir, un sistema basado en elementos y en usuarios. Este motor de recomendación estará alimentado por una clase de matrices conocidas en mercadotecnia como *matrices de utilidad*, las cuales se describen a detalle en la siguiente sección.

1.2. Matriz de utilidad

Desde la popularización de las transacciones por Internet, cada vez es más fácil recopilar datos sobre los usuarios de las plataformas que usan. Estos datos incluyen información sobre perfiles de usuario, intereses, comportamiento de navegación, comportamiento de compra y valoraciones sobre diversos artículos, etc. Es natural aprovechar estos datos para hacer recomendaciones a los clientes sobre posibles intereses de compra. En el problema de la recomendación, los pares *usuario-artículo* tienen valores de utilidad asociados. Así, para m usuarios y n artículos, se obtiene una matriz D de $n \times m$ de valores de utilidad, la cual también se denomina *matriz de utilidad*. El valor de utilidad de un par usuario-artículo puede corresponder al comportamiento de compra o a las valoraciones del usuario sobre el artículo.

La matriz de utilidad tiene una influencia significativa en la elección del algoritmo de recomendación, por ejemplo:

- *Preferencias positivas únicamente*: La matriz de utilidad especificada sólo contiene preferencias positivas. Por ejemplo, la especificación de una opción de "me gusta" en una red social, la búsqueda de un artículo en un sitio web o la compra de una cantidad determinada de un artículo corresponden a una preferencia positiva. Estas funciones suelen ser especificadas por el analista de forma heurística en función de la aplicación.

- *Preferencias positivas y negativas (calificaciones o ratings)*: En este caso, el usuario especifica las valoraciones que representan su agrado o desagrado por el artículo. La incorporación de la aversión del usuario en el análisis es significativa porque hace que el problema sea más complejo y a menudo requiere algunos cambios en los algoritmos subyacentes.

Es común que la matriz de utilidad sea rala (también llamada escasa, dispersa o *sparse* por su traducción en inglés), es decir, con un porcentaje alto de entradas vacías, debido a que los usuarios no califican a todos los artículos, por lo que las entradas que faltan corresponden a preferencias no especificadas. No es raro que los valores de n y m en dicha matriz D superen las 10^5 entradas; así también, un usuario típico puede haber especificado no más de 10 valoraciones de un universo de más de 10^5 elementos. Esta diferencia cambia significativamente los algoritmos utilizados en los dos casos (véase la referencia [Aggarwal, 2015]).

La matriz de utilidad para entrenar sistemas recomendadores basados en ratings se construye mediante los siguientes pasos:

1. Se definen las entidades clave:

    - Usuarios (U): Personas que califican los ítems.
    - Ítems (I): Elementos que los usuarios pueden calificar (películas, productos, libros, etc.).
    - Ratings (R): Calificaciones dadas por los usuarios a los ítems.

2. Estructura de la matriz de utilidad: Se define una matriz D de tamaño $n \times m$, donde cada columna representa un usuario u_i, cada fila representa un ítem i_j y la celda r_{ji} contiene el rating que el usuario u_i dio al ítem i_j, o un valor vacío (nulo) si no ha sido calificado. Con lo anterior, se obtiene una matriz ejemplificada como:

$$D = \begin{bmatrix} 5 & 3 & ? & 1 \\ 4 & ? & 2 & ? \\ ? & 5 & 3 & 4 \\ 2 & ? & 1 & 5 \end{bmatrix},$$

   donde el símbolo "?" en la matriz D indica una calificación desconocida.

3. Obtención de los ratings: Existen dos tipos de ratings, los explícitos e implícitos. En los explícitos los usuarios proporcionan calificaciones a los ítems directamente, por ejemplo, puntuaciones de 1 a 5 estrellas para películas y series en Netflix; mientras que los implícitos se infieren a partir del comportamiento del usuario, por ejemplo, el tiempo de reproducción de una película. En esta investigación, los explícitos serán obtenidos a partir de la base de datos, aquellos implícitos serán obtenidos mediante un modelo de regresión en la sección de validación.

4. Manejo de valores faltantes: Como la matriz suele ser dispersa, se utilizan técnicas de imputación como:

   a) Imputación simple con algún valor aleatorio, fijo o por la media por usuario o ítem.

   b) Factorización de matrices, tales como la descomposición en valores singulares (SVD), la factorización no negativa (NMF), entre otras.

   c) Modelos de aprendizaje automático, como los basados en vecindades (KNN), redes neuronales, regresiones lineales y logísticas, entre otros.

   En esta investigación, se realizaron los tres pasos para la construcción de la matriz de utilidad, y se encuentran descritos en la sección 2.2.

5. Uso en sistemas recomendadores: Es en este paso donde la matriz de utilidad alimenta a algún sistema recomendador de alguna de las siguientes maneras:

   a) Filtrado colaborativo: Se usa la matriz para encontrar similitudes entre usuarios o ítems.

   b) Modelos de aprendizaje profundo: Se entrenan redes neuronales para predecir ratings faltantes.

   c) Descomposición en factores latentes: Métodos como "ALS" (Alternating Least Squares) extraen representaciones de usuarios e ítems.

En este trabajo de tesis, se utilizó la matriz de utilidad para el caso a) del punto 5. En específico, se buscó encontrar similitudes entre artículos; sin embargo, es relevante primero introducir un concepto apto de similitud en el contexto de este trabajo, el cual se describe a detalle en la siguiente sección.

1.3. Similitud y distancias

En diversas aplicaciones de minería de datos se requiere determinar objetos, patrones y atributos similares o distintos en los datos, por lo que es necesario introducir los conceptos de *similitud* y *distancias*. Prácticamente todos los problemas de minería de datos, como la agrupación (*clustering*), la detección de valores atípicos (*outlier detection*) y la clasificación (*classification*), requieren el cálculo de la similitud. Esto también es la esencia en los sistemas recomendadores.

Un enunciado formal para el problema de la cuantificación de la similitud o la distancia es el siguiente:

Dados dos objetos O_1 y O_2, determinar un valor de la similitud $Sim(O_1, O_2)$ (o distancia $Dist(O_1, O_2)$) entre los dos objetos.

En las funciones de similitud, valores más altos implican una mayor similitud, mientras que en las funciones de distancia, los valores menores implican una mayor cercanía. En algunos dominios, como los datos espaciales, es más natural hablar de funciones de distancia, mientras que en otros dominios, como el texto, es más natural hablar de funciones de similitud (véase el capítulo 3 de la referencia [Aggarwal, 2015] para un contexto más amplio).

Respecto a las funciones de distancia, su correcta elección es fundamental para el diseño eficaz de algoritmos de minería de datos, pues una mala elección de dichas funciones puede ser perjudicial para la calidad de los resultados. Se debe tener en cuenta que una mala elección puede generar sesgos en las conclusiones de los análisis, dependiendo del dominio de la aplicación, pues las funciones de distancia suelen ser sensibles a la distribución, la dimensión y el tipo de datos. En algunos tipos de datos, como los multidimensionales, es más sencillo definir y calcular funciones de distancia que en otros, como los datos de series temporales (véase [Aggarwal, 2015]).

Una de las funciones más comunes de distancia para datos cuantitativos (datos cuyos campos contienen valores numéricos) es la norma L_p. La norma L_p, para $1 \leq p < \infty$, entre dos puntos $X = (x_1, ..., x_d)$ y $Y = (y_1, ..., y_d)$ se define como:

$$Dist_p(X, Y) = \left(\sum_{i=1}^{d} |x_i - y_i|^p \right)^{1/p}.$$

Dos casos especiales de la norma L_p son las métricas euclidiana ($p = 2$) y Manhattan ($p = 1$). Una propiedad de la primera es que es invariante de la rotación, porque la distancia en línea recta entre dos puntos de datos no cambia con la orientación del sistema de ejes. Sin embargo, surge una dificultad en el caso de dimensiones altas (d relativamente grande), ya que estas funciones influyen en las métricas de eficiencia de los recomendadores cuando los datos son de dimensión alta debido al impacto variable de la escasez de datos, la distribución, el *ruido* (véase la sección 1.5) y la relevancia de las características. La distancia L_p generalizada es la más adecuada para esta situación y se define de forma similar a la norma L_p, con la salvedad de que se asocia un coeficiente a_i a la i-ésima característica, este coeficiente se utiliza para ponderar el componente

de la característica correspondiente en la norma L_p, y se define como:

$$Dist_{Gen}(X,Y) = \left(\sum_{i=1}^{d} a_i |x_i - y_i|^p \right)^{1/p},$$

esta distancia también es referida como la generalización de la distancia de Minkowsky.

En conclusión, la norma L_p generalizada puede utilizarse para reducir el problema de la alta dimensionalidad, por lo que se comprueba que algunas modificaciones a las métricas pueden ayudar a resolver algunos problemas que surgen en la práctica.

En el caso de la similitud, se suele tratar como la medida que cuantifica qué tan parecidos son dos ítems o usuarios en función de ciertas características o comportamientos. Esta medida es fundamental para hacer recomendaciones, ya que permite identificar ítems o usuarios similares y, con base en ello, sugerir contenido relevante.

Para los sistemas recomendadores, la idea es escoger dos objetos x y y, aislar a los usuarios que los han calificado en común y luego aplicar una técnica para determinar la similitud. Por otra parte, para que pueda calcularse la similitud entre dos usuarios u_i y u_j, deben cumplirse dos condiciones: ambos usuarios deben haber calificado al menos un ítem y, además, deben tener al menos un ítem calificado en común; si alguna de estas condiciones no se cumple,será necesario recurrir a otras técnicas que permitan generar recomendaciones ([Olguín et al., 2019]).

En este marco, dos usuarios son *muy diferentes* si asignan calificaciones opuestas a un mismo ítem o si no tienen ítems calificados en común. En cambio, estos se consideran *muy similares* al tener expresada una preferencia positiva parecida para los mismos productos. Sin embargo, incorporar tanto preferencias positivas como negativas suele complicar el proceso desde el punto de vista de la recopilación de datos. Además, inferir preferencias negativas resulta más desafiante cuando estas se deducen del comportamiento de los usuarios en lugar de basarse directamente en las calificaciones que asignan a los ítems ([Aggarwal, 2015]).

En recomendaciones que se basan en el contenido, tanto los usuarios como los artículos son asociados a descripciones basadas en sus características. Los perfiles de los artículos se pueden determinar utilizando el texto de la descripción del mismo, por lo que se utilizan funciones de similitud para texto. Por otro lado, un usuario también puede haber especificado explícitamente sus intereses en alguna red social, por lo que es posible inferir su perfil a partir de su comportamiento de compra o de navegación. Así, cuando las descripciones textuales de los elementos coinciden con el perfil del usuario (inferido o no), esto puede considerarse como un indicador de similitud.

En los filtrados colaborativos, el objetivo consiste en aprovechar las preferencias de los usuarios en forma de valoraciones o comportamiento de compra de forma "colaborativa", en beneficio de toda la red. La matriz de utilidad se utiliza para determinar usuarios de relevancia para ciertos artículos en específico, o artículos relevantes para algunos usuarios particulares en el proceso de recomendación. Un paso intermedio clave en este enfoque es la determinación de grupos similares de artículos y usuarios. Los patrones de estos grupos proporcionan el conocimiento colaborativo

necesario en el proceso de recomendación.

Los métodos basados en el contenido tienen la ventaja de que no requieren una matriz de utilidad y aprovechan la información sobre el contenido específico del ámbito del dominio; por otro lado, la información de contenido sesga la recomendación hacia los elementos descritos con palabras clave similares a los que el usuario ha visto en el pasado. Por contraparte, los métodos de filtrado colaborativo suelen estar entre los modelos más utilizados, debido a que trabajan directamente con la matriz de utilidad y, por tanto, pueden evitar estos sesgos, aunque tienen la desventaja de que las matrices de utilidad utilizadas en los algoritmos de filtrado colaborativo son exageradamente grandes y dispersas, como se vio en la sección anterior.

En un nivel básico, el filtrado colaborativo puede verse como un problema de estimación de valores perdidos o de completación de matrices, en el que se especifica una matriz de utilidad incompleta de $n \times m$, y se desean estimar las entradas que faltan. Sin embargo, los problemas de filtrado colaborativo presentan un caso especial desafiante en términos de tamaño de los datos y la escasez de estos últimos ([Aggarwal, 2015]).

Cabe aclarar que los dos modelos no son excluyentes. A menudo es posible combinar los métodos basados en el contenido con métodos de filtrado colaborativo para crear una puntuación de preferencia combinada ([Aggarwal, 2015]). En esta investigación se define la similitud entre usuarios mediante la determinación de sus preferencias al aplicar una métrica L_p generalizada a la matriz de utilidad, esto conduce a un modelo de filtrado colaborativo basado en la teoría de k vecinos más cercanos, que se definen en la siguiente sección.

1.4. Modelo de K vecinos más cercanos (KNN)

Los sistemas recomendadores realizan sugerencias o recomendaciones considerando las preferencias o gustos de la comunidad de *vecinos* de un usuario activo. Estos vecinos son elegidos mediante métricas de similaridad y/o distancia. La idea es relativamente sencilla: dadas las valoraciones del usuario, encontrar todos los usuarios con preferencias similares a él en el pasado y, a continuación, hacer predicciones sobre todos los productos desconocidos que el usuario activo no ha valorado, pero que sí han sido valorados por sus vecinos. Dado que se realizan cálculos de similitud, estos sistemas de recomendación también se denominan *sistemas de recomendación basados en la similitud.*

El algoritmo de los K vecinos más cercanos (KNN) es un algoritmo de clasificación supervisada. Este algoritmo trata de utilizar la similitud usuario-usuario o la similitud elemento-elemento para hacer recomendaciones a partir de la matriz de utilidad y clasificar dichas entidades en clases, utilizando un conjunto de datos de los cuales ya se conoce su clase, este será el conjunto de entrenamiento. Primero, se calcula la distancia entre el usuario objetivo u_i, luego se calcula su similitud con todos los demás usuarios del conjunto de entrenamiento y se selecciona a los K con menor distancia o mayor similitud; estos elementos serán sus K *vecinos más cercanos.*

En el caso de una matriz basada en *ratings*, el cálculo de la similitud tiene mayor complejidad debido a que los distintos usuarios pueden tener diferentes escalas de valoración. Un usuario puede estar predispuesto a que le gusten la mayoría de los ítems, y otro puede estar predispuesto a que no le agraden. Además, distintos usuarios pueden haber valorado ítems distintos.

Para introducir las métricas de similitud y distancias, es necesario considerar a los usuarios y sus calificaciones como vectores en un espacio métrico. Para esto, se definen un par de conceptos:

Definición 1.1. *Sean U el conjunto de usuarios, I el de ítems y D la matriz de utilidad de dimensión $n \times m$. Para cada usuario $u \in U$, se define el vector de calificaciones del usuario u como*

$$r_u = (r_{u_1}, \cdots, r_{u_n}),$$

donde r_{u_j}, es la entrada en la posición (j, u) de la matriz D.

El vector r_u representa todos los ratings que el usuario u le ha dado a los ítems i_j en D, con $i_j \in I$, $j \in \{1, \cdots, n\}$. Dado que el usuario no necesariamente califica a todos los ítems, se tiene que el vector de calificaciones no nulas (aquellas que son explícitas) del usuario tiene una dimensión menor que n.

De forma similar se define ahora:

Definición 1.2. *Sean U el conjunto de usuarios, I el de ítems y D la matriz de utilidad de dimensión $n \times m$. Para cada ítem $i \in I$, se define el vector de calificaciones del ítem i como*

$$r'_i = (r_{i_1}, \cdots, r_{i_m}),$$

donde r_{i_k}, es la entrada en la posición (i, k) de la matriz D.

En este caso, el vector r_i representa todos los ratings que el ítem i ha recibido de los usuarios u_k en D, con $u_k \in U$, $k \in \{1, \cdots, m\}$. Dado que el ítem no es visto ni calificado por todos los usuarios, se tiene que el vector de calificaciones no nulas del ítem tiene una dimensión menor que m.

Definición 1.3. *Sean $u \in U$, y r_u el vector de calificaciones del usuario u. Para un valor fijo de $k \in \mathbb{N}$, se define el conjunto de distancias*

$$Dist(u) = \{Dist_*(r_u, r_x) : \ x \in U\}.$$

Donde $Dist_*(\cdot)$ representa una función de distancia o similitud.

Para los k valores más pequeños del conjunto de distancias $Dist(u)$ se define como $\mathcal{N}_K(u)$ al conjunto de índices de los registros correspondientes a los k valores más pequeños.

Así también, para un ítem se tiene:

Definición 1.4. *Sean $i \in I$, y r_i el vector de calificaciones del ítem i. Para un valor fijo de $k \in \mathbb{N}$, se define el conjunto de distancias*

$$Dist'(i) = \left\{Dist_*(r'_i, r'_y) : \ y \in I\right\}.$$

Donde $Dist_*(\cdot)$ representa una función de distancia o similitud.

Para los k valores más pequeños del conjunto de distancias $Dist'(i)$ se define como $\mathcal{N}'_K(i)$ al conjunto de índices de los registros correspondientes a los k valores más pequeños.

La métrica más simple que mide la distancia entre dos vectores, que en este caso, son los vectores de calificaciones de dos ítems, es la métrica euclidiana, la cual se define como:

$$d(i'_x, i'_y) = \sqrt{\sum_{j=1}^{m} (i_{x_j} - i_{y_j})^2}$$

donde $i'_x = (i_{x_1}, \ldots, i_{x_m})$ y $i'_y = (i_{y_1}, \ldots, i_{y_m})$ son los vectores de calificaciones, en donde algunas entradas son no nulas, y la diferencia $(x_i - y_i)$ representa la distancia entre las coordenadas correspondientes de los vectores. Se elevan al cuadrado, se suman y se extrae la raíz cuadrada para obtener la distancia.

En notación de norma, se expresa como:

$$d(i'_x, i'_y) = \|x - y\|_2$$

donde $\|\cdot\|_2$ representa la norma euclidiana.

Por otra parte, una medida que captura la similitud entre dos vectores de ratings de dos usuarios x y y es el *coeficiente de correlación de Pearson* que se define como sigue:

Sean $r_x = (r_{x_1}, \cdots, r_{x_s})$ e $r_y = (r_{y_1}, \cdots, r_{y_s})$ los vectores de las calificaciones comúnmente especificadas de un par de usuarios, con medias muestrales $\hat{x} = \sum_{i=1}^{s} r_{x_i}/s$ y $\hat{y} = \sum_{i=1}^{s} r_{y_i}/s$, respectivamente, el coeficiente de Pearson se define como:

$$Pearson(r_x, r_y) = \frac{\Sigma_{i=1}^{s}(r_{x_i} - \hat{x}).(r_{y_i} - \hat{y})}{\sqrt{\sum_{i=1}^{s}(r_{x_i} - \hat{x})^2} \cdot \sqrt{\sum_{i=1}^{s}(r_{y_i} - \hat{y})^2}}.$$

Alternativamente, la valoración media de un usuario se calcula promediando todas sus valoraciones especificadas, en lugar de utilizar sólo los elementos valorados coincidentemente por el par de usuarios en cuestión. Esta forma alternativa de calcular la media es más común y puede afectar significativamente al cálculo de Pearson por pares ([Aggarwal, 2015]).

El coeficiente de Pearson se calcula entre el usuario objetivo y todos los demás usuarios. El grupo de pares del usuario objetivo se define como los usuarios con el coeficiente de Pearson más alto con él. Los usuarios con correlaciones muy bajas o negativas se eliminan del grupo y las valoraciones medias de cada uno de los elementos (especificados) de este grupo se devuelven como valoraciones recomendadas. Para lograr una mayor robustez, también se pondera cada valoración con el coeficiente de correlación de Pearson de su propietario mientras se calcula la media. Esta valoración media ponderada puede proporcionar una predicción para el usuario objetivo y los artículos con las valoraciones más altas se recomiendan al usuario. El principal problema de este enfoque es la diferencia en las escalas de valoraciones que diferentes usuarios pueden proporcionar a diferentes ítems. Por tanto, es necesario normalizar las calificaciones brutas antes de determinar la valoración media (ponderada) del grupo de pares. La calificación normalizada de un usuario se define restando la valoración media de cada una de sus valoraciones. Como antes, la media ponderada de la calificación normalizada de un elemento en el grupo de pares se determina como una predicción normalizada. La valoración media del usuario objetivo se añade de nuevo a la predicción de la valoración normalizada para obtener una predicción de calificación no ponderada ([Aggarwal, 2015]).

La principal diferencia conceptual con el enfoque basado en el usuario es que los grupos de pares se construyen en términos de artículos y no de usuarios, por lo que las similitudes deben calcularse entre elementos (filas de la matriz de utilidad). Para esto, antes de calcular las similitudes entre las filas se normaliza la matriz de utilidad, esto es, se resta la media de cada fila de la matriz de valoraciones de esa columna. A continuación, el coseno define la similitud entre las valoraciones normalizadas $U = (u_1 \cdots u_d)$ y $V = (v_1 \cdots v_d)$, de un par de de artículos, mediante la función:

$$Cos(U, V) = \frac{\Sigma_{i=1}^{d} u_i \cdot v_i}{\sqrt{\Sigma_{i=1}^{d} u_i^2} \cdot \sqrt{\Sigma_{i=1}^{d} u_i^2}}.$$

Esta similitud se denomina *similitud del coseno ajustada*, porque los ratings se normalizan antes de calcular el valor de la similitud.

Consideremos el caso en el que hay que determinar la valoración del artículo j para el usuario i. El primer paso consiste en determinar los k artículos más similares al artículo j basándose en

la similitud del coseno ajustada. Entre los k artículos más parecidos al artículo j, se determinan aquellos para los que el usuario i ha especificado valoraciones. El valor medio ponderado de estas puntuaciones se presenta como el valor previsto. El peso del artículo r en esta media es igual a la similitud coseno ajustada entre el artículo r y el artículo j. La idea básica es aprovechar las propias valoraciones del usuario en el paso final de la predicción y es la que da origen al modelo propuesto en esta investigación y que esta basado en la teoría de vecinos cercanos.

1.5. Datos atípicos (outliers)

En este trabajo, se analiza la influencia de los datos atípicos en la eficiencia de los modelos recomendadores, lo que se conoce como *robustez*. Informalmente, la robustez puede definirse como la capacidad de un software para mantener un comportamiento "aceptable" a pesar de condiciones de ejecución excepcionales o imprevistas (como la falta de disponibilidad de recursos del sistema, fallos de comunicación, entradas no válidas o estresantes, entre otros). Esta característica es especialmente importante para las aplicaciones cuyo entorno de ejecución no puede ser completamente prevista en el momento del desarrollo (véase [Fernandez et al., 2005]).

Así, la robustez de los sistemas recomendadores a datos atípicos se refiere a la capacidad que tiene un sistema recomendador para dar buenas recomendaciones de ítems (recomendaciones que el usuario elija) a pesar de que la base de datos con la que se este trabajando tenga datos atípicos.

Un valor atípico (o dato atípico) es un punto en el conjunto de datos que difiere significativamente del resto. Según Hawkins, citado en [Aggarwal, 2015, p. 17], define un valor atípico de la siguiente manera: *"Un valor atípico es una observación que se desvía tanto de las otras observaciones que suscita sospechas de que fue generado por un mecanismo diferente."* En la minería de datos y la estadística, los valores atípicos también se denominan anormalidades, discordantes, desviantes o anomalías. En la mayoría de las aplicaciones, los datos son creados por uno o más procesos generadores que pueden reflejar la actividad en el sistema u observaciones recogidas sobre las entidades que influyen en la generación de los datos.

En la mayoría de las aplicaciones, los datos tienen una distribución estadística *normal*, y las anomalías se reconocen como desviaciones de este modelo. Los datos que siguen este modelo son llamados *normales*, *valores típicos* o *inliers*. En algunas aplicaciones, como la detección de intrusiones o fraudes, los valores atípicos corresponden a secuencias de múltiples puntos de datos en lugar de puntos de datos individuales. Estas anomalías también suelen ser llamadas *anomalías colectivas*, porque sólo pueden deducirse colectivamente de un conjunto o secuencia de puntos de datos, ya que generan patrones anómalos de actividad.

La detección de valores atípicos comienza con un enfoque intuitivo, donde el analista, con base en su experiencia y conocimiento del dominio, identifica observaciones que parecen desviarse de lo esperado. Este método es subjetivo pero útil en etapas iniciales, ya que no requiere herramientas complejas y permite una rápida identificación de patrones inusuales. Sin embargo, en muchos casos, los datos están inmersos en *ruido*, que representa observaciones que se encuentran en el límite entre lo normal y lo anómalo, lo que dificulta la detección intuitiva.

Para abordar estas limitaciones, se utilizan métodos estadísticos, que ofrecen un marco más estructurado y objetivo. La mayoría de los algoritmos de detección de valores atípicos utilizan alguna medida cuantificada de la *atipicidad* de un elemento en el conjunto de datos, como la dispersión de la región subyacente, la distancia basada en el vecino más cercano o el ajuste a la distribución de datos subyacente. Cada punto de datos se encuentra en un espectro continuo que va de los datos normales al ruido y, por último, a las anomalías.

Métodos de detección de valores atípicos basados en estadística: Las técnicas estadísticas de detección de valores atípicos suponen que los puntos de datos normales aparecen en las regiones de alta probabilidad de un modelo estocástico, mientras que los valores atípicos aparecen en las regiones de baja probabilidad. Para una revisión general de los métodos estadísticos de detección de valores atípicos se puede consultar [Hodge and Austin, 2004]. Existen dos categorías principales: los *métodos basados en pruebas de hipótesis*, que calculan un estadístico de prueba para determinar si se rechaza la hipótesis nula (ausencia de valores atípicos), y los *métodos de ajuste de distribución*, que deducen una función de densidad de probabilidad para identificar datos con baja probabilidad como atípicos. Un análisis completo y comparación entre ambos enfoques se puede encontrar en [Lehmann and Lösler, 2016, Boukerche et al., 2020, Smiti, 2020].

Las ventajas que presentan las técnicas basadas en estadística son varias. En primer lugar, si los datos subyacentes siguen una distribución específica, estas técnicas pueden ofrecer una interpretación de los valores atípicos. Además, suelen proporcionar una puntuación o un intervalo de confianza para cada dato, en lugar de limitarse a una decisión binaria. Esta puntuación puede ser útil como información adicional al tomar decisiones sobre un dato de prueba. Otra ventaja es que suelen funcionar de manera no supervisada, es decir, sin necesidad de datos de entrenamiento etiquetados.

No obstante, dichas técnicas presentan algunas desventajas. Una de ellas es que suelen basarse en el supuesto de que los datos se generan a partir de una distribución determinada, lo cual no siempre es cierto, especialmente en conjuntos de datos reales de gran dimensión. Incluso cuando este supuesto puede justificarse, la elección del mejor estadístico para detectar anomalías no es una tarea sencilla. Construir pruebas de hipótesis para distribuciones complejas que se ajusten a conjuntos de datos de gran dimensión resulta particularmente complicado.

Aunado a lo anterior, en muchas de las aplicaciones reales, la interpretación de un modelo de detección de valores atípicos es relevante desde la perspectiva del analista y del contexto del problema que se aborda. A menudo es necesario justificar por qué un dato en concreto debe considerarse atípico, ya que proporciona al analista más pistas sobre el diagnóstico necesario en un escenario específico. Para esto, existe el concepto de valores atípicos contextuales.

Los valores atípicos contextuales, son observaciones que se desvían significativamente del comportamiento esperado dentro de un contexto específico. A diferencia de los valores atípicos globales, que son anomalías en todo el conjunto de datos, los valores atípicos contextuales solo son considerados anomalías en relación con un contexto particular. Este enfoque es especialmente útil en datos con estructuras complejas, como series temporales o datos espaciales, donde el contexto juega un papel crucial en la interpretación de las anomalías (Ver [Alimohammadi and Chen, 2022]). A diferencia de los métodos estadísticos tradicionales, que se centran en la distribución global de los datos, los métodos contextuales permiten identificar anomalías que solo son relevantes dentro de un marco específico.

Recientemente, se han hecho muchas propuestas sobre el descubrimiento de valores atípicos contextuales con atributos generales del entorno; algunas de estas se dirigen al caso de uso de

la exploración de datos, mientras que otras se dirigen al caso de explicación y diagnóstico. En el caso de las propuestas orientadas a la exploración de datos, algunas asumen que los atributos del entorno y de comportamiento se proporcionan como entrada, mientras que otras exploran el espacio de posibles atributos del entorno y de comportamiento para identificar valores atípicos contextuales. Este último enfoque es particularmente útil cuando no se tiene un conocimiento previo claro de qué atributos definen el contexto, permitiendo al analista descubrir relaciones ocultas o no evidentes. Para más detalles consulte [Singh and Upadhyaya, 2012].

El objetivo de las propuestas dirigidas al caso de uso de explicación y diagnóstico es descubrir los atributos del entorno y de comportamiento en los que un objeto dado es un valor atípico. En [Ilyas and Chu, 2019], se presenta un algoritmo de ejemplo para cuando se dan los atributos del entorno y de comportamiento y otro para cuando no se dan. La idea es relativamente sencilla: calcular una correlación entre los atributos del entorno y los de comportamiento proporcionados, y definir los valores atípicos contextuales como aquellos datos que se desvían de la correlación aprendida. La correlación se modela utilizando un modelo de mezcla gaussiano, y los parámetros del modelo se aprenden utilizando métodos de maximización de expectativas (*Expectation maximization*).

La simple idea de encontrar una correlación entre los atributos de entorno y los atributos de comportamiento, sin la necesidad de la modelación, puede ayudar demasiado en el criterio del analista a la hora de determinar si un dato es atípico o no. Esto se debe a que, en muchos casos, el analista puede identificar patrones o desviaciones significativas mediante un análisis exploratorio inicial, utilizando herramientas visuales o técnicas estadísticas básicas. Por ejemplo, al graficar los atributos de comportamiento en función de los atributos de entorno, el analista puede observar tendencias o puntos que se desvían claramente de la norma. En definitiva, siempre será crucial el criterio del analista y su conocimiento profundo del problema específico, desde la detección temprana de anomalías hasta la selección del método estadístico o el marco contextual más adecuado.

En resumen, la detección de valores atípicos puede abordarse desde tres niveles: la intuición del analista, los métodos estadísticos y el análisis contextual. Cada enfoque tiene sus fortalezas y limitaciones, pero combinados proporcionan un marco robusto para identificar y comprender anomalías en diversos tipos de datos.

1.6. Métricas de Hausdorff

La distancia de Hausdorff (HD por sus siglas en inglés) es una medida útil para determinar en qué medida una forma es similar a otra, sobre todo en la Visión por Computadora, Análisis de Imágenes y Reconocimiento de Patrones ([Wang and Tan, 2012]). Sin embargo, la HD es sensible a los valores atípicos. Muchos investigadores han propuesto modificaciones de la HD para proveerla de mayor robustez. La HD y sus modificaciones se basan en el cálculo de la distancia entre elementos de un conjunto y su punto más cercano en otro conjunto, denominadas colectivamente *distancias de Hausdorff basadas en el vecino más cercano* ($NNHD$).

Matemáticamente hablando, la distancia de Hausdorff es un operador no lineal que mide el desajuste entre dos conjuntos dados; esto es, para dos conjuntos de puntos finitos $M = \{m_1, m_2, \cdots, m_m\}$ y $T = \{t_1, t_2 \cdots, t_t\}$, la distancia de Hausdorff se define como:

$$H(M,T) = \text{máx}\{h(M,T), h(T,M)\},$$

con

$$h(M,T) = \max_{a \in M} \min_{b \in T} \{\|a - b\|\},$$

donde la norma $\|\cdot\|$ puede ser Euclidiana, Minkowski o, más general, alguna L_p, $1 \leq p < \infty$. La función $H(M,T)$ es llamada *distancia de Hausdorff no dirigida* y la función $h(T,M)$ es llamada *distancia de Hausdorff dirigida*.

La distancia de Hausdorff mide la similitud extendida de un punto a otro, a diferencia de la mayoría de otras formas de comparación, la medida de Hausdorff no está basada en encontrar el modo correspondiente; así, es más tolerante con las perturbaciones (o ruido) de locaciones de puntos, debido a que considera la proximidad de los mismos por sobre la superposición exacta. Sin embargo, la distancia de Hausdorff tiene la desventaja de ser sensible a los valores atípicos ([Wang and Tan, 2012]). Por ejemplo, dados dos conjuntos de puntos A y B, consideremos a B como el conjunto A más un único punto que está alejado de otros puntos de A. Así, la distancia de Hausdorff entre A y B está determinada por la distancia del punto único a otras partes. La sensibilidad a los valores atípicos es fatal para la medición de la similitud. Por lo tanto, muchos investigadores proponen modificaciones de la distancia de Hausdorff original para reducir la influencia de los valores atípicos. Por ejemplo, en [Wang and Tan, 2012] proponen la *distancia de Hausdorff dirigida* (RHD) como:

$$h^p_{RHD}(M,T) = p^{th} \max_{a \in M} \min_{b \in T} ||a - b||,$$

la cual calcula la p-ésima máxima distancia. Cuando $p = 1$, RHD es igual a HD, esta definición selecciona automáticamente la p^{va} "mejor coincidencia" de puntos de M para ignorar los valores atípicos obvios y minimiza la distancia Hausdorff.

En [Wang and Tan, 2012] realizaron un experimento sobre comparación de imágenes binarias de líneas (imágenes en blanco y negro que sólo contienen puntos y líneas). Propusieron

24 funciones de distancia basadas en la métrica de Hausdorff entre dos conjuntos de puntos; dichas funciones fueron tomadas de cuatro funciones de distancia no dirigida y seis funciones de distancia dirigida, a continuación, crearon una combinación de cada entre ellas y revisaron el comportamiento de dichas métricas con el ruido de las imágenes de testeo correspondientes para cada una de las imágenes. El ruido fue distribuido de forma aleatoria de una distribución normal.

Para el experimento, los patrones de líneas aleatorias fueron generados en imágenes de 256×256, y crearon aleatoriamente las ubicaciones de los puntos extremos de una cantidad aleatoria de líneas. A continuación se generó un conjunto de 50 imágenes con ruido a partir de cada uno de estos patrones de líneas con las siguientes técnicas:

1. Perturbar aleatoriamente los puntos finales de cada una de las líneas.
2. Añadir aleatoriamente características de línea.
3. Eliminar aleatoriamente características de línea.
4. Voltear pixeles aleatoriamente.

Para evaluar el comportamiento de las funciones que crearon, se utilizaron dos características:

1. Debe tener un gran poder de discriminación, es decir, una gran capacidad para notar las diferencias entre las imágenes de prueba.
2. El valor se debe incrementar a medida que la diferencia de los objetos aumenta (se agregan mayores niveles de ruido).

Al término del experimento, notaron que la métrica que mejor comportamiento tuvo fue aquella creada con la métrica dirigida:

$$d(\mathcal{A}, \mathcal{B}) = \frac{1}{N_A} \sum_{a \in A} d'(a, \mathcal{B}),$$

y la métrica no dirigida:

$$f_2(d(\mathcal{A}, \mathcal{B}), d(\mathcal{B}, \mathcal{A})) = \text{máx}((\mathcal{A}, \mathcal{B}), d(\mathcal{B}, \mathcal{A})).$$

De manera análoga, en esta investigación se ha propuesto usar esta métrica para plantear un modelo KNN para recomendadores de películas de Netflix basados en ratings (véase la sección 3.3).

Capítulo 2

Preprocesamiento de la base de datos

El preprocesamiento de datos es el conjunto de técnicas y prácticas que se aplican a los datos antes de ser utilizados en un proyecto de Ciencia de Datos. Un correcto preprocesamiento de los datos garantiza que estos sean transparentes y comprensibles, lo que facilita la interpretación de los resultados y permite a los científicos de datos explicar por qué un modelo toma decisiones específicas.

El análisis descriptivo examina los datos para obtener información sobre lo que ha ocurrido u ocurre en el entorno de datos. Se caracteriza por las visualizaciones de datos, como los gráficos circulares, de barras o líneas, las tablas o las narraciones generadas. Un buen análisis descriptivo es capaz de impactar positivamente en el desarrollo de un modelo predictivo que puede llegar a ser relevante en una área de la ciencia.

En este capítulo se realiza un análisis exploratorio (descriptivo) de los datos y la construcción de la matriz de utilidad. En la primera sección, se exhibe el análisis exploratorio de los datos, se realiza la descripción del tipo de datos y la dispersión que poseen; se realiza una búsqueda de datos atípicos; se normalizan los datos y se hace una comparación con los datos sin normalizar.

En la segunda sección, se describe la construcción de la matriz de utilidad de los datos desde una perspectiva matemática, que es un objeto que utilizan los modelos KNN para poder generar las recomendaciones, así como dos modificaciones a la misma para su posterior uso en los modelos KNN del capítulo 3.

2.1. Análisis exploratorio de los datos

Netflix es una empresa fundada en 1997 por Reed Hastings y Marc Randolph, cuya idea surgió cuando Hastings fue multado por devolver un videocasete de Blockbuster tarde. Originalmente, Netflix ofrecía el alquiler de DVD que se entregaban por correo postal, pero a medida que fue creciendo su éxito, migraron a un servicio de streaming que permite ver películas y series en línea

(véase [Datalaria, 2019]). Funciona por suscripción y está disponible en casi cualquier dispositivo conectado a internet.

Según datos publicados en [Technology, 2023], el primer trimestre de 2024 Netflix contaba con alrededor de 282,680,000 usuarios en todo el mundo. Dicho éxito se debe al buen modelo de negocios que manejan y en la importancia que le dan a los datos que generan sus usuarios. En palabras del Director de Comunicaciones Globales de Netflix en 2019, *"existen 33 millones de versiones diferentes de Netflix"*, pues Netflix personaliza sus contenidos y decisiones para cada uno de sus usuarios.

Netflix recopila datos como, por ejemplo:

- Tasa de usuarios que han comenzado una serie y han acabado todos sus capítulos.
- Cuáles son los puntos más frecuentes de corte en los que se deja de ver una serie.
- Tiempos que los usuarios tardan en visualizar 2 capítulos seguidos de la misma serie.
- Cuando se pausa, se rebobina o se adelanta un capítulo.
- Días de la semana donde se ven más capítulos, por ejemplo, Netflix ha comprobado que se ven más series los días laborables y más películas los fines de semana.
- Ubicación, fecha y hora de visualización de cada contenido por cada usuario.
- Dispositivo utilizado para visualizar el contenido.
- Las valoraciones a los contenidos (4 millones cada día).
- Resultados de las búsquedas (3 millones cada día).
- Datos dentro de los contenidos, como cuando empiezan los créditos para controlar que hacen los usuarios justo después.

Gracias al análisis de todos estos datos, Netflix comenzó a tomar decisiones siguiendo las premisas que reflejaban la fase de análisis prescriptivo ([Datalaria, 2019]).

Netflix hace mucho hincapié a la recomendación de películas y series desde su aplicación con el objetivo de sugerir de manera precisa contenidos a cada usuario y nunca se quede sin opciones de visualización (lo cual implicaría la cancelación de su suscripción). Y Netflix ha conseguido que un 75 % de los contenidos visualizados partan de sus recomendaciones ([Datalaria, 2019]).

En octubre de 2006, Netflix ofreció un concurso abierto al mejor algoritmo de filtrado colaborativo para predecir las valoraciones de los usuarios sobre películas, basado en valoraciones anteriores sin ninguna otra información sobre los usuarios o las películas, es decir, sin que los usuarios estuvieran identificados salvo por los números asignados para el concurso.

El 21 de septiembre de 2009, el gran premio de un millón de dólares fue otorgado al equipo Pragmatic Chaos de BellKor, que superó al propio algoritmo de Netflix para predecir las valoraciones en un 10,06 % ([Wikipedia, 2023]).

Para dicho concurso, Netflix liberó una base de datos de usuarios y películas, y gracias a la plataforma kaggle.com, es posible acceder a ella (https://www.kaggle.com/datasets/netflix-inc/netflix-prize-data). Kaggle es una plataforma que tiene como finalidad, entre otras, buscar o publicar bases de datos, trabajar y conectar a profesionales y aficionados; esta plataforma también realiza competencias y retos sobre temas innovadores en cuanto a Ciencia de Datos.

Por otro lado, en esta investigación se ha utilizado el lenguaje de programación Python y un Notebook de Google Colaboratory, que es herramienta gratuita que permite ejecutar código en la nube de Google, con la ventaja de tener CPU y RAM virtual para llevar a cabo proyectos que requieran de cálculos complejos, es especialmente popular para la Ciencia de Datos.

Las *bibliotecas* necesarias para llevar a cabo un mejor manejo de los datos utilizadas en este proyecto son:

- **Pandas**: para analizar, limpiar, explorar y manipular datos.
- **Numpy**: para realizar cálculos matemáticos y científicos, también para manejar datos y realizar operaciones matemáticas complejas.
- **Sklearn**: crear modelos de aprendizaje automático y ciencia de datos.
- **Seaborn**: para crear gráficos estadísticos.
- **Mathplotlib**: crear visualizaciones de datos, como gráficos, histogramas, diagramas de barras y gráficos de dispersión

Con las herramientas anteriores, se realizó una primera exploración a la base de datos y se observó, entre otras cosas, que cuenta con dos sub-bases de información, que llamaremos datasets, en formato .csv, que es un tipo de archivo de texto que almacena datos en filas y columnas separadas por comas, puntos y comas u otros caracteres.

El primer dataset, el cual es representado en la Tabla 2.1, contiene 4 columnas y $100,836$ filas. Las columnas son: *userId*, que representa el número de identificación (o *id*) de cada usuario, y son datos de tipo entero; *moveId*, que representa el *id* de cada película, y sus entradas son números enteros; *rating*, que representa las calificaciones que el usuario le ha dado a la película correspondiente, con entradas de números decimales de 0 a 5; por último, *timestamp*, que representa la fecha y hora en que dichas calificaciones fueron asignadas, con entradas de números enteros.

Por otra parte, el segundo dataset, representado en la Tabla 2.2, cuenta con $9,472$ filas y 3 columnas, la primer columna se llama *movieId* y representa el id de la película, con entradas de tipo entero; la segunda se llama *title* y representa el título de las películas, con datos de tipo *pandas.core.series.Series*; por último, *genres* que representa a los géneros que pertenecen las películas, con datos de tipo *pandas.core.series.Series*, dicho tipo de dato es una estructura de datos unidimensional que utiliza la biblioteca Pandas.

userId	movieId	rating	timestamp
1	1	4.0	964982703
1	3	4.0	964981247
1	6	4.0	964982224
1	47	5.0	964983815
1	50	5.0	964982931
1	70	3.0	964982400
1	101	5.0	964980868

Tabla 2.1: Algunos registros del dataset que contiene ratings de películas.

movieId	title	genres
1	Toy Story (1995)	Adventure, Animation, Children, Comedy, Fantasy
2	Jumanji (1995)	Adventure, Children, Fantasy
3	Grumpier Old Men (1995)	Comedy, Romance
4	Waiting to Exhale (1995)	Comedy, Drama, Romance
5	Father of the Bride Part II (1995)	Comedy
6	Heat (1995)	Action, Crime, Thriller
7	Sabrina (1995)	Comedy, Romance

Tabla 2.2: Algunos registros del dataset de películas(títulos y géneros).

Al analizar ambos datasets, se tiene que existen $100,835$ ratings en total, con un número de $9,724$ películas únicas; con 610 usuarios únicos; con un promedio de ratings generados por usuario de 165.3 y 10.37 ratings en promedio recibidos por película.

De los $100,836$ ratings totales en la base de datos, se obtiene una media de ratings de 3.501 puntos por rating, lo que sugiere que, en general, los usuarios tienden a dar ratings ligeramente por encima del punto medio (2.5 puntos o estrellas), con una desviación estándar de 1.04 puntos, que se podría considerar relativamente baja, lo que indica que la mayoría de los ratings están concentrados cerca de la media (entre 2.46 y 4.54). Esto sugiere que los usuarios tienden a dar ratings consistentes, sin mucha variabilidad extrema. La mediana tiene un valor de 3.5, y es muy similar a la media, lo que indica que la distribución de los ratings es simétrica alrededor de 3.5; esto sugiere que no hay un sesgo fuerte hacia ratings altos o bajos. La calificación mínima que los usuarios pueden asignar a las películas es de 0.5, lo que indica que existen que algunas películas que recibieron puntuaciones (o estrellas) muy bajas. Por contraparte, la calificación más alta es 5, que indica que existen películas con la calificación más alta posible. El 50 % de los datos entre el segundo y tercer cuartil de los ratings está entre 3.0 y 4.0, lo que confirma que la mayoría de los ratings son moderadamente altos.

Por último, el hecho de que el mínimo sea 0.5 y el máximo 5.0 sugiere que podría haber algunos outliers en los extremos (ratings muy bajos o muy altos). Sin embargo, que la desviación

estándar sea baja indica que estos outliers no son demasiados.

Al visualizar la distribución de los ratings para las películas mediante un gráfico de barras (Figura 2.1) se observa una distribución sesgada a la derecha, pues la mayoría de las calificaciones se concentran en valores altos (3.0, 4.0 y 5.0). En el gráfico, el eje **X** representa los distintos ratings que los usuarios, en general, dieron a las películas; por contraparte, el eje Y representa la frecuencia en la que aparecen dichos ratings. En este caso, la calificación más frecuente es 4.0, con $26,818$ apariciones; también hay una cantidad significativa de ratings de 3.0 y 5.0, lo que refuerza la idea de que los usuarios son más propensos a calificar con puntajes altos, y que en general dichas valoraciones son contundentes, pues no otorgan valoraciones medias (aquellas con un 0.5 añadido). También puede reflejar un sesgo de selección, es decir, es más probable que los usuarios califiquen películas que realmente les gustan en lugar de aquellas que no les interesan.

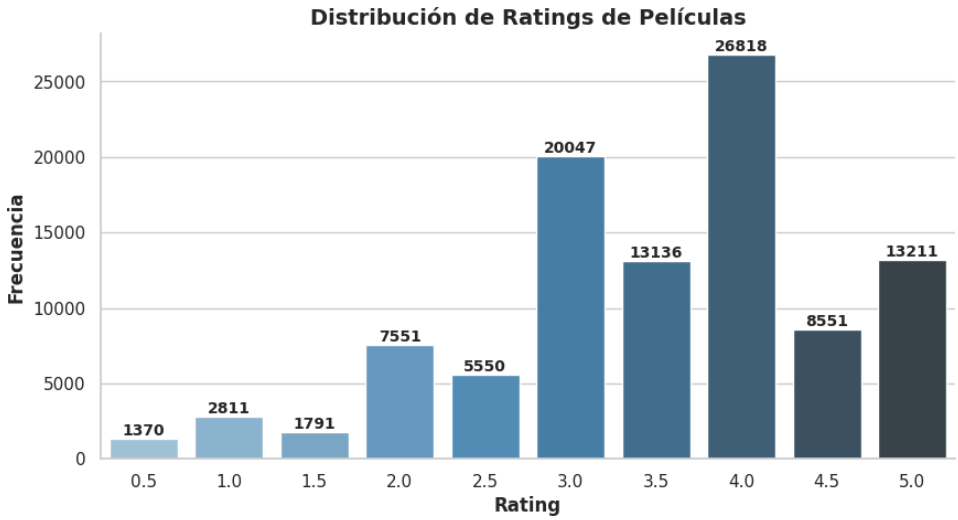

Figura 2.1: Distribución de ratings de películas.

Por otro lado, el rating 3.0 también tiene una frecuencia relativamente alta, lo que sugiere que algunos usuarios dan calificaciones más neutrales, mientras que las calificaciones inferiores a 2.0 son mucho menos frecuentes, lo que señala que los usuarios tienden a evitar ratings bajos o que las películas en el dataset son, en general, bien recibidas.

A continuación, se analiza la distribución de la cantidad de valoraciones que los usuarios otorgan a las distintas películas; es decir, a cuántas películas ha calificado un usuario, esto mediante el gráfico de la Figura 2.2. En ella, el eje **X** representa la cantidad de ratings, mientras que el eje **Y** indica la proporción de usuarios que han otorgado dicha cantidad de valoraciones. La línea vertical punteada representa la media total del número de valoraciones.

En la Figura 2.2, la media de ratings por usuario es de 165.3, o sea que, en promedio, cada usuario ha realizado aproximadamente 165 calificaciones, lo que indica que los usuarios están bastante activos al momento de realizar calificaciones. La distribución no está centrada en la media, de hecho, tiene una *cola larga* (en este caso, derecha), la cual sugiere una mayor

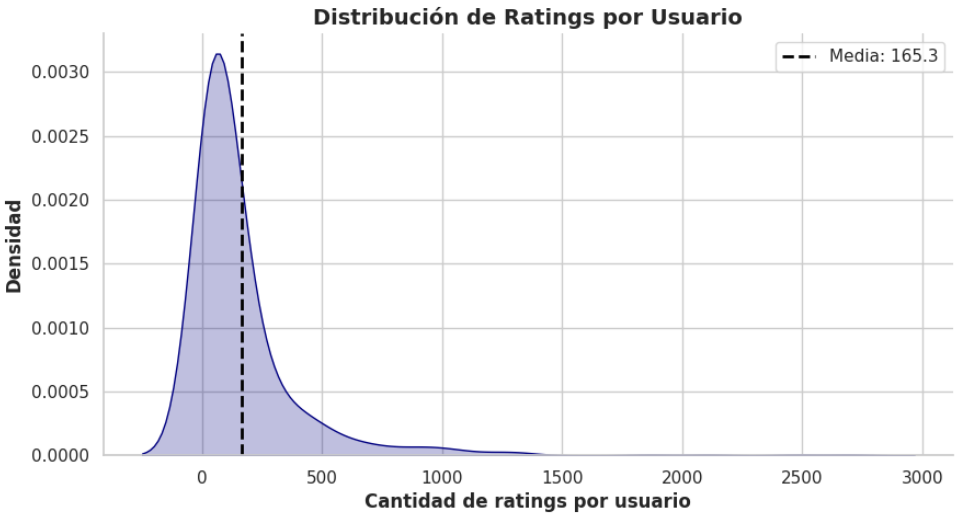

Figura 2.2: Distribución de ratings otorgados por los usuarios

variabilidad en el número de calificaciones por usuario, pues existen valores inusuales alejados de la media (hacia la derecha); es decir, que existe una proporción relativamente pequeña que, en este caso, ha otorgado demasiadas calificaciones a las películas, lo que aumenta la magnitud de la asimetría de la distribución. Debido a este sesgo, es imperativo echar un vistazo más profundo y revisar las particularidades de la base de datos; para esto, se buscan ahora valores extremos.

Se realiza primero una búsqueda de la película con el promedio de ratings más bajo, la cuál resulta ser "Gipsy", del año 1962; es un musical y cuenta únicamente con una calificación (0.5). Posteriormente, al buscar aquella que tiene el promedio de ratings más alto de todas las películas, esta es "Lamerica", del año 1994, perteneciente a los géneros de Aventura y Drama, con un promedio de calificación de 5.0, pero que cuenta únicamente con 2 valoraciones.

Es natural pensar que, para considerar una película como *buena*, es necesario que sea popular (es decir, conocida o calificada por muchas personas) y que tenga un promedio de calificaciones alto. En ese sentido, para la película mejor calificada, al solo contar con dos calificaciones, no puede ser un punto de referencia para considerarla buena, a pesar de ser la que posea el promedio más alto de las calificaciones. De forma similar se considera para la película con el promedio más bajo.

En este contexto, para obtener una visión más amplia de las películas y sus respectivas calificaciones, se realiza una tabla que incluye el promedio y la desviación estándar de las valoraciones de cada película. Esta tabla permite analizar, a grandes rasgos, la relación entre la cantidad de valoraciones recibidas, el promedio de calificación y la influencia de calificaciones extremas en dicho promedio (véase la Tabla 2.3).

Una vez realizada la Tabla 2.3, se crea la gráfica mostrada en la Figura 2.3, en la cual se analiza la cantidad de ratings recibidos por película (eje **X**) y se compara con el promedio de las calificaciones recibidas de cada una (eje **Y**). En esta gráfica, cada punto representa una película.

movieId	n_ratings	mean_ratings	std_ratings
1	215	3.920930	0.834859
2	110	3.431818	0.881713
3	52	3.259615	1.054823
4	7	2.357143	0.852168
5	49	3.071429	0.907148
⋮	⋮	⋮	⋮
193581	1	4.000000	NaN
193583	1	3.500000	NaN
193585	1	3.500000	NaN
193587	1	3.500000	NaN
193609	1	4.000000	NaN

Tabla 2.3: Resumen de ratings por película.

Se nota nuevamente que la mayoría de promedios de calificaciones que tienen las películas está entre 2.5 y 4.5. También es posible observar que son relativamente pocas las películas que cuentan con más de 100 calificaciones, y dichas películas cuentan con un promedio de calificación entre 3.0 y 4.5. Es prudente inferir que la gran mayoría de los usuarios otorgaron calificaciones a las mismas películas, es decir, a las más populares, y que estas películas populares tienen un promedio de calificación alto y que influyeron en el promedio general de todas las películas en la base de datos. También se observa que existen películas con casi ninguna valoración, y que esas películas cuentan con calificaciones que abarcan todo el espectro posible de las opciones de calificación, es decir, existen películas que cuentan con 0.5, 1.0, ... , 4.5, 5.0 en su promedio de calificación, pero que tienen menos de 5 calificaciones.

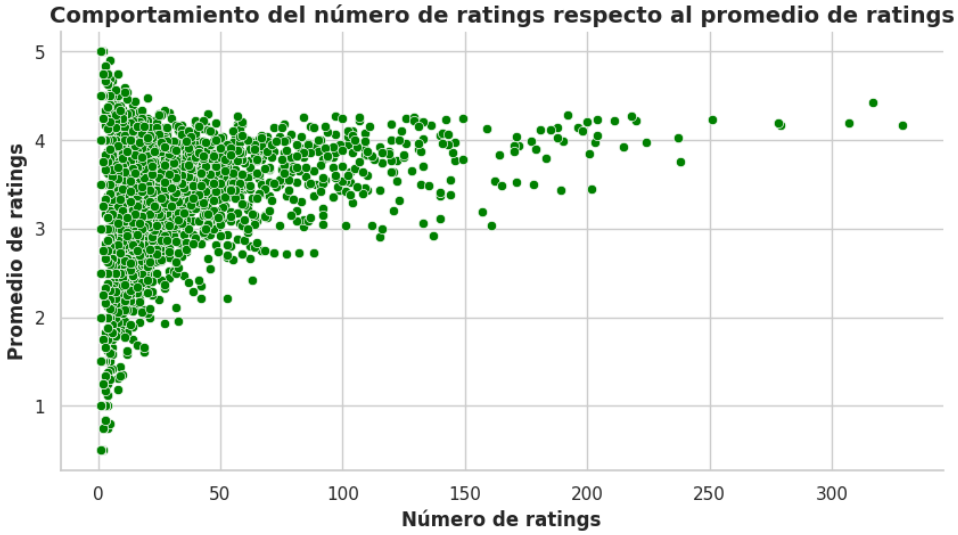

Figura 2.3: Comportamiento del número de ratings respecto a su promedio.

A continuación, se tiene la gráfica de la Figura 2.4 que, en este caso, el eje **Y** representa la desviación estándar de las valoraciones que han recibido cada una de las películas y el eje **X** representa el número de valoraciones que recibió cada película, y se observa que la gran mayoría de las películas posee una desviación estándar entre 0.5 y 1.5 puntos de valoración. Dicha desviación se mantiene tanto en películas con muchas valoraciones, como aquellas que poseen pocas. Además, se puede observar una cantidad pequeña de películas que poseen desviaciones en todo el espectro, pero que contienen muy pocas valoraciones. Se puede inferir que estas películas podrían ser de *nicho*, pues de las pocas valoraciones que tiene, o gustan mucho, o disgustan de igual forma.

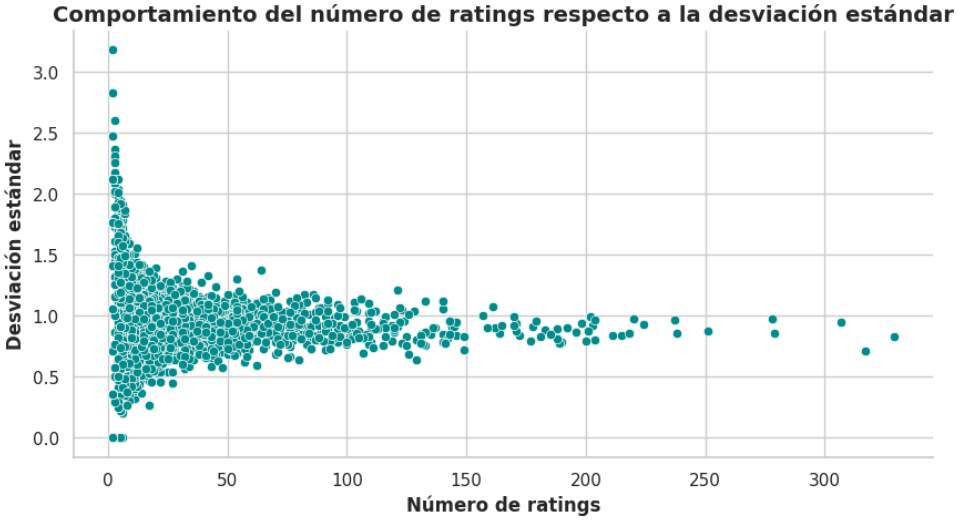

Figura 2.4: Comportamiento de la desviación estándar de los ratings.

Otro gráfico que ayuda a visualizar la distribución conjunta de dos variables, como lo pueden ser el número de ratings y el promedio de calificación de las películas, es la gráfica de densidad bivariada, cuya variación en la intensidad del color indica la variación de la concentración de los datos. Las áreas más claras indican menor densidad de datos; mientras que las más oscuras representan una mayor densidad de ellos. Por esta razón, se muestra la gráfica de la Figura 2.5, que exhibe dónde se concentran los ratings promedio que poseen cada una de las películas. Esta visualización es útil para identificar tendencias, patrones y posibles relaciones entre las variables, como si las películas con más ratings tienden a tener un promedio de calificación más estable o si hay un sesgo hacia ciertos valores de calificación. Más aún, se observa una alta densidad de puntos en la parte izquierda de la gráfica, lo que indica que la mayoría de las películas tienen pocos ratings (menos de 50); por otro lado, a medida que el número de ratings aumenta, la dispersión se reduce, reafirmando que pocas películas han sido calificadas por un gran número de usuarios, es decir, son populares. También existen algunas películas con ratings cercanos a 5, pero con pocos ratings, lo que sugiere que películas menos populares pueden haber recibido solo evaluaciones positivas. También hay algunas películas con ratings bajos, pero en menor medida.

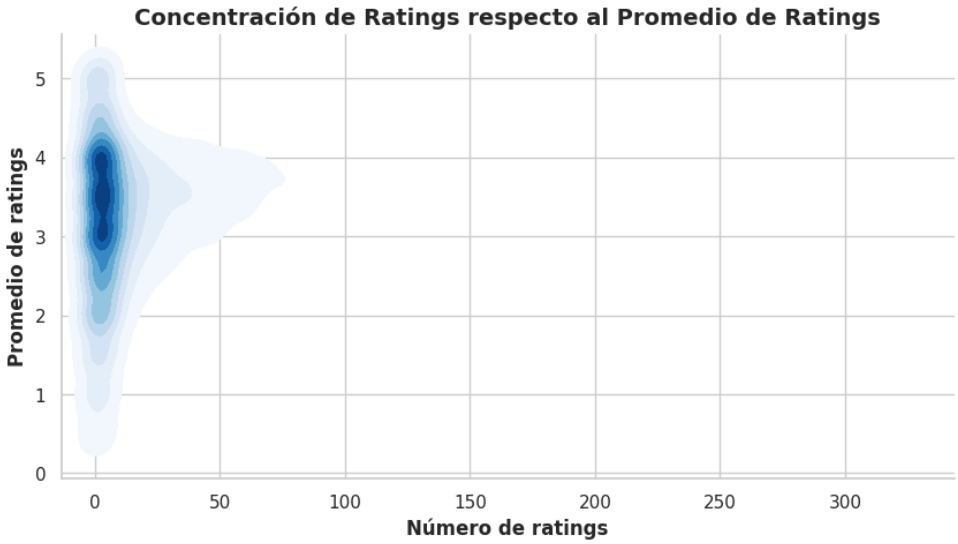

Figura 2.5: Concentración de los ratings.

Debido a lo anterior, es posible inferir que en la base de datos existen *outliers*, y que cuenta con *ruido*; una forma de tratar con ellos es usar técnicas de normalización. En esta investigación se ha considerado la estrategia de normalizar con el promedio bayesiano para continuar con el análisis de los datos. El promedio bayesiano es una media ajustada que se utiliza cuando los datos son pocos o están sesgados y cuando los promedios provienen de muestras que poseen tamaños distintos. También, reduce la influencia de valores atípicos que pueden distorsionar la percepción de calidad o rendimiento (como en este caso).

Se prevé que el promedio bayesiano evite los ratings sesgados, ya que el promedio bayesiano empuja los valores extremos hacia la media general; por otro lado, también se busca que se le dé más peso a películas con más ratings, pues cuanto más calificaciones tenga una película, más se confiará en su rating promedio real, mientras que las películas con pocas calificaciones se ajustarán más hacia la media general hasta que acumulen suficiente información, y con esto, reduzca la variabilidad en ratings bajos, ya que hay mayor dispersión de ellos, y el promedio bayesiano ayuda a suavizar estos valores para que sean más confiables.

En la Figura 2.6 se observa que existe una relación casi lineal entre los promedios para valores de rating promedio altos (mayores a 3). Esto sugiere que cuando una película tiene muchas calificaciones altas, el ajuste bayesiano no cambia demasiado el resultado. También, se observa que en los laterales de la gráfica, el promedio bayesiano no permite valores extremos tan fácilmente. Esto indica que este método pondera el número de ratings, evitando sesgos de películas con muy pocas calificaciones, pero con valores extremos.

Para analizar de mejor manera este fenómeno, se realiza una comparación en la que se restan la media y la media bayesiana, con el objetivo de verificar qué tanto se diferencian uno del otro; para esto, en la Figura 2.7 se representa la cantidad de ratings que obtiene una película (eje **X**) y la diferencia entre el promedio normal y bayesiano (eje **Y**). En el diagrama se observa que

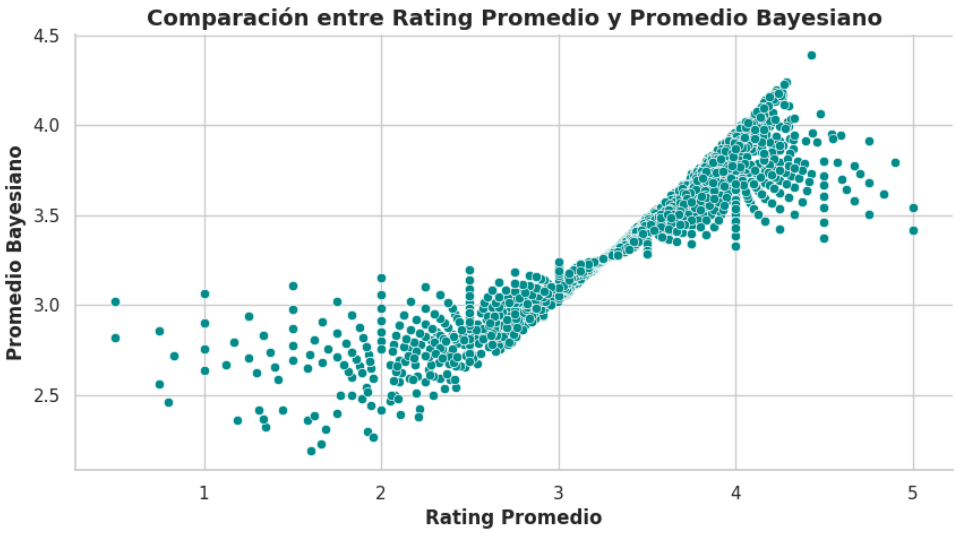

Figura 2.6: Comparación entre el promedio y promedio bayesiano.

cuando la cantidad de ratings es baja, la diferencia tiene una gran dispersión; esto indica que las películas con pocos ratings pueden tener una diferencia significativa entre su promedio de calificaciones y el promedio bayesiano; por otra parte, para películas con muchas calificaciones, su promedio y el promedio bayesiano son prácticamente iguales, lo que confirma que los ajustes bayesianos afectan principalmente a las películas con pocos ratings.

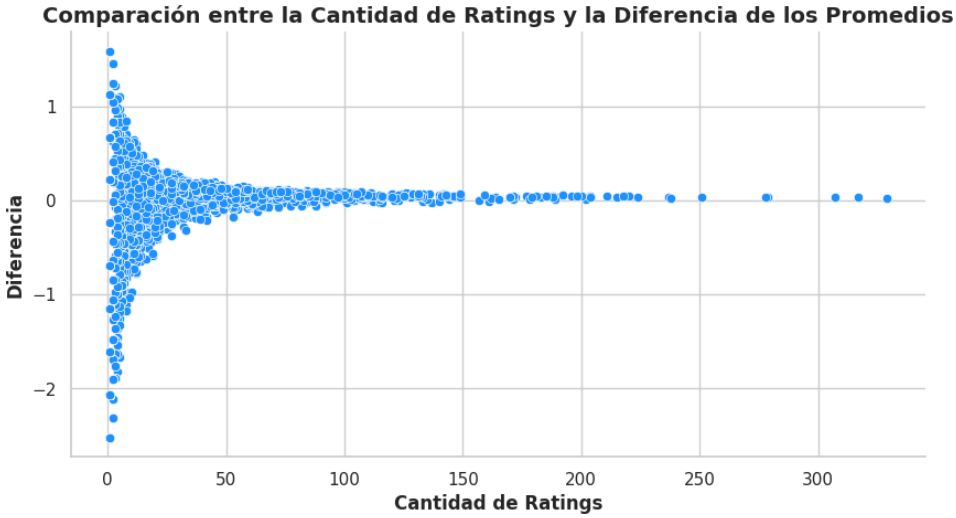

Figura 2.7: Diferencias entre los promedios.

Es necesario apuntar que la diferencia en los promedios para las películas con pocas valoraciones no es indicativa de un ajuste correcto, pues, como se intuyó en la Figura 2.1, puede suceder que los usuarios solo califican aquellos títulos que en verdad les gustan, por lo que, para una película con muy pocas valoraciones, y además, que dichas valoraciones sean bajas,

puede ocurrir que el promedio normal de la película sea el más bajo posible (0.5), y al utilizar el promedio bayesiano, podrían ser sumados cerca de 2.0 puntos; por lo que la película pasaría a tener un promedio bayesiano cercano a 2.5, muy cercano a la media general. Esto último puede provocar la sobreestimación de ciertas películas. De forma similar, aunque no tan extrema, se analiza para las películas mejor calificadas, provocando en este caso una subestimación de ciertos títulos.

Consideremos como ejemplo la película *Gypsy*, que fue la que peor promedio obtuvo (0.5) y con solo una valoración, al aplicar el promedio bayesiano, este título pasa a tener un promedio de calificación de 3.01. No obstante, bajo este mismo promedio, la película peor calificada pasa a ser *Speed 2: Cruise control (Máxima velocidad)* del año 1997, cuyo promedio bayesiano es de 2.19, y un total de 19 calificaciones, esta misma película tuvo un promedio de 1.61. De forma similar para los títulos con mejor calificación en promedio; anteriormente se observó que la película mejor calificada fue *Lamerica*, con únicamente 2 calificaciones y un promedio de 5.0 en sus valoraciones; en contraste, el promedio bayesiano de esta película baja a 3.54. Ahora bien, la película mejor calificada bajo el promedio bayesiano es *The Shawshank Redemption (Sueño de fuga)* del año 1994, con 4.39 y un total de 317 valoraciones, esta misma película obtuvo un promedio normal de calificación de 4.42.

En este punto, se cuenta con la base de datos normalizada, por lo que se procede a construir la matriz de utilidad. El proceso constructivo se exhibirá con detalle en la siguiente sección.

2.2. Construcción de la matriz de utilidad

La matriz de utilidad es una forma compacta y estructurada de representar los ratings que los usuarios asignaron a las distintas películas. Este objeto es utilizado en los sistemas de recomendación y ayuda a la detección de patrones y a predecir calificaciones futuras. Un tema importante en este objeto es el manejo de datos faltantes, ya que se pueden mejorar las predicciones y generar recomendaciones más precisas gracias al manejo inteligente de ellos.

La matriz de utilidad se forma a través de *diccionarios*, que son funciones que se encargan de *mapear*[1] los ids de los ítems o usuarios, y asignarlos a los índices de los renglones y columnas de la matriz.

Tomemos los conjuntos $U = \{j : j \text{ es un id de usuario}\}$, $I = \{i : i\text{es un id de película}\}$ y $R = \{x_{i,j} \in D\}$. Se define la función $\psi : I \times U \to R$, que representa el mapeo de índices, como:

$$\psi(i,j) = (\psi_1(i), \psi_2(j)) = (x'_i, x''_j) = x_{i,j},$$

donde $\psi_1(i)$ da la posición del vector fila i en D correspondientes a una película, y $\psi_2(j)$ da la posición del vector columna j en D, por lo que $\psi(i,j)$ nos dará el rating del usuario j a la película i, en otras palabras:

- $\psi_1(i) = x'_i$ se encarga de rastrear la fila x' correspondiente en D a partir del id de la película i.
- $\psi_2(j) = x''_j$ se encarga de rastrear la columna x'' correspondiente en D a partir del id del usuario j.

Ahora, es necesaria una función que a partir de la posición de un rating nos devuelva los ids de usuarios y películas, es decir, la inversa de ψ, donde $\psi^{-1} : R \to I \times U$ y se define como

$$\psi^{-1}(x_{i,j}) = (\psi_1^{-1}(x'_i), \psi_2^{-1}(x''_j)) = (i,j),$$

que al igualar componente a componente obtenemos lo siguiente:

- $\psi_1^{-1}(x'_i) = i$ rastrea el id de la película a partir de la posición de la fila que indica x_{ij} en la matriz D.
- $\psi_2^{-1}(x''_j) = j$ rastrea el id del usuario a partir de la posición de la columna que indica x_{ij} en la matriz D.

Con estas funciones es posible crear la matriz de utilidad, en la cual, los índices de sus renglones hacen referencia a los ids de las películas y los índices de sus columnas representan los ids de los usuarios; en este caso, la matriz obtenida para esta investigación tiene dimensiones de 9724×610. Dichos ids no son mapeados de forma consecutiva, ni tienen una secuencia en

[1] Asignar datos de origen a campos de destino.

específico, su ordenamiento en la matriz depende propiamente de los métodos usados para crear la matriz.

El proceso de la creación de la matriz sigue una serie de aspectos para su correcta creación, como el cálculo de las dimensiones de la matriz, en donde se determina el número único de películas (n) y usuarios (m) únicos en los datasets utilizados, y a través de ellos, se determinan dichas dimensiones. A partir de lo anterior, son utilizados los diccionarios y sus inversos, para empezar a rellenar la matriz, en donde las columnas **userId** y **movieId** de los datasets se transforman en listas de índices numéricos utilizando los mapeos creados. Estas listas indican la posición de cada usuario y película en la matriz. Cada celda contiene las calificaciones que los usuarios asignan a las películas, y las coordenadas de estos valores son especificados como una tupla[2]. La matriz resultante es dispersa, lo que optimiza el uso de memoria al almacenar solo los valores distintos de cero.

Sin embargo, a pesar de su utilidad, trabajar con una matriz de utilidad presenta diversos desafíos. Uno de los principales retos es la gran cantidad de valores faltantes, ya que es común que muchos usuarios califiquen un número muy reducido de elementos de un número extenso de productos. Como resultado, la matriz suele estar mayormente incompleta, lo que dificulta su análisis y procesamiento. Esta escasez puede obstaculizar el rendimiento de los algoritmos de recomendación, dificultando la búsqueda de patrones significativos. Además, se genera el problema del arranque en frío o *cold start*, que surge cuando se introducen nuevos usuarios o elementos en el sistema, ya que no hay datos suficientes para hacer predicciones precisas. Abordar estos desafíos a menudo requiere la implementación de técnicas más sofisticadas para el manejo de los datos.

Para medir la *dispersión* de la matriz, se utiliza el cociente creado por el número de elementos que no están vacíos entre el número de elementos totales, es decir:

$$S = \frac{\text{Número de elementos no vacíos}}{\text{Número de elementos totales}}.$$

En este caso, la dispersión es de 1.7 %, es decir, que el 1.7 % de las celdas en la matriz están no vacías. Dada la estructura de la matriz, las celdas vacías son llenadas con ceros, aunque realmente lo correcto es manejarlas vacías o aplicar alguna técnica de imputación, pero debido a que se trabajará con el concepto de distancia, en esta ocasión es conveniente hacerlo de esta manera.

A partir de la matriz resultante, es posible obtener otras dos matrices; la primera, se obtiene al normalizar la matriz original; la segunda, se obtiene al normalizar la original con el promedio bayesiano, ya que de acuerdo a lo observado en la sección 2.1, es posible tener la creencia *a priori* de que las películas con pocas valoraciones contienen una mayor variación en cuanto a las valoraciones obtenidas de los usuarios. Una solución que se propone es trabajar con la media y con la media bayesiana. A continuación se describe el proceso para la creación de ambas matrices.

[2]Colección de elementos ordenados que no se pueden modificar.

Se define la media bayesiana para cada película con índice i como:

$$b_i = \frac{C_i \cdot \mu_i + s_i}{C_i + n_i},$$

donde n_i representa el número de calificaciones que cada película ha recibido en su respectivo índice, es decir, para cada película con índice i, se calcula el número de calificaciones no nulas de esta, o sea, el número de usuarios que han calificado la película con índice i de la siguiente forma:

$$n_i = \sum_{j=1}^{m} \mathbb{I}(x_{i,j} \neq 0),$$

donde $\mathbb{I}(\cdot)$ es una función indicadora que vale 1 si la condición es verdadera y 0 en caso contrario.

Posteriormente, para cada película con índice i, se calcula la suma de todas las calificaciones que ha recibido:

$$s_i = \sum_{j=1}^{m} x_{i,j}.$$

Ahora, para cada película i, se calcula la calificación promedio:

$$\mu_i = \frac{s_i}{n_i}.$$

Posteriormente se crea un vector $\boldsymbol{\mu} \in \mathbb{R}^n$, donde cada elemento μ_i representa la calificación promedio de la película i, es decir:

$$\mu = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_n \end{bmatrix}.$$

Para el caso de la media bayesiana, se propone que $m = \mu_i$ sea la media de los promedios de las calificaciones y, por la creencia *a priori*, C_i representa el número de observaciones del dataset. Entonces, $C = C_i$ puede ser cuántas calificaciones en promedio tiene una película.

Es posible entender el promedio bayesiano como un promedio ponderado, donde el factor de ponderación esta dado por el promedio de ratings en todas la películas C , mientras que la otra ponderación esta dada por los ratings de la película que se este analizando.

Así, para esta matriz, se redefine la media bayesiana como:

$$b_i \ = \frac{C \cdot m + s_i}{C + n_i},$$

Con lo anterior se crea el vector columna b de la forma:

$$\mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}.$$

Por último, se crean dos matrices X_{mean} y X_{bayesian}, ambas de tamaño $n \times m$. Cada fila de X_{mean} es una copia del vector $\boldsymbol{\mu}$, y cada fila de X_{bayesian} es una copia del vector $\mathbf{b}$:

$$X_{\text{mean}} = \begin{bmatrix} \mu_1 & \mu_1 & \cdots & \mu_1 \\ \mu_2 & \mu_2 & \cdots & \mu_2 \\ \vdots & \vdots & \ddots & \vdots \\ \mu_M & \mu_M & \cdots & \mu_M \end{bmatrix},$$

y

$$X_{\text{bayesian}} = \begin{bmatrix} b_1 & b_1 & \cdots & b_1 \\ b_2 & b_2 & \cdots & b_2 \\ \vdots & \vdots & \ddots & \vdots \\ b_M & b_M & \cdots & b_M \end{bmatrix}.$$

Una vez se han creado estas matrices, se aplica una función que permita que sean dispersas y, puesto que se crearon a partir de la matriz original de este proyecto, todas ellas tienen la misma forma y estructura, con la salvedad de que cambian los valores de las entradas no vacías de cada una.

Ahora bien, de X_{mean} y X_{bayesian} se crean las dos matrices con las que se trabajará en esta investigación, las cuales son:

$$D_{\text{norm}} = D - X_{\text{mean}}$$

y

$$D_{\text{bayesian}} = D - X_{\text{bayesian}}$$

Estas últimas, junto a D (creada en la sección 1.2) han sido los objetos de trabajo para este proyecto.

Capítulo 3

Modelos recomendadores

El modelo ***k-Nearest Neighbors (KNN)*** es un algoritmo de aprendizaje automático no paramétrico[1] que se ha utilizado de forma extensa en problemas de clasificación y regresión en Ciencia de Datos.

En este capítulo, se abordan tres variantes de este modelo, en específico, el modelo que maneja por defecto la biblioteca *scikit-learn*. La primera sección explica el funcionamiento del modelo con la métrica euclidiana; la segunda sección explica cómo funciona el modelo utilizando la métrica de Hausdorff, y la tercera sección explica cómo se ha modificado el modelo para poder utilizar en él la métrica de Hausdorff modificada vista en la sección 1.6.

Por último, se realizan una serie de simulaciones que pretenden evaluar y comparar los modelos anteriores.

3.1. Modelo KNN con métrica euclidiana

Respecto a los sistemas de recomendación basados en el filtrado colaborativo, el KNN se emplea para identificar usuarios o ítems (en este caso, películas) similares a un ítem dado. Este algoritmo se basa en la idea de que ítems con características o patrones de interacción similares (como los ratings que los usuarios les asignan) tienden a ser relevantes entre sí. La similitud se calcula mediante una métrica de distancia entre los vectores de características de los ítems, lo que permite encontrar los "vecinos más cercanos" en el espacio de características.

Formulación Matemática

Una vez dada la matriz de utilidad $D \in \mathbb{R}^{n \times m}$, donde n es el número de ítems (películas) únicos de la base de datos, y m es el número de usuarios únicos, cada fila $\mathbf{x}_i \in \mathbb{R}^n$ representa las calificaciones de los usuarios para el ítem i. El objetivo es encontrar los k ítems más cercanos a un ítem dado $\mathbf{x}_q$, utilizando alguna métrica de distancia $d(\mathbf{x}_i, \mathbf{x}_q)$.

[1] Que no hace suposiciones específicas sobre la forma funcional de la distribución subyacente de los datos.

En el caso particular de esta investigación se ha utilizado la distancia euclidiana, definida como:

$$d(\mathbf{x}_i, \mathbf{x}_q) = \sqrt{\sum_{j=1}^{n} (x_{i,j} - x_{q,j})^2}. \tag{3.1}$$

Sin embargo, esta métrica no puede ser utilizada de forma directa en los vectores de calificaciones resultantes de la matriz de utilidad. Antes de aplicar la métrica, se requieren funciones previas para un mejor manejo de los datos.

En primer lugar, se obtiene el índice de la película de la que se quieren obtener los vecinos más cercanos, con ayuda de los diccionarios que se han creado anteriormente. Posteriormente, se obtiene de la matriz el vector de calificaciones de la película.

Una vez hecho lo anterior, se crea el modelo KNN por medio de una función de la biblioteca *sklearn* (scikit-learn) de python, en donde se especifica el número de vecinos, la métrica con la que se evaluará la cercanía, e incluso, se tiene un parámetro específico que permite al algoritmo seleccionar automáticamente la mejor estrategia para calcular los vecinos más cercanos, en función de las características de los datos y los parámetros proporcionados. Esto resulta muy útil, ya que optimiza el rendimiento del modelo sin que el usuario tenga que elegir manualmente el algoritmo más adecuado.

El paso siguiente es crucial para el modelo KNN, y es el entrenamiento. Se debe ajustar el modelo a la matriz en particular. La misma biblioteca ya maneja esa función.

Una vez creado el modelo y ajustado a la matriz, se pueden obtener los k vecinos más cercanos a la película de interés con la métrica que se ha especificado antes, utilizando el vector de calificaciones y, si es necesario, se redimensiona para hacer más fácil su manipulación.

Luego de obtenidos los índices de las películas más cercanas a la película de interés, se utilizan las funciones inversas de los diccionarios y se mapean los índices con sus respectivos ids. Así, con el uso de otro diccionario que contiene los títulos de las películas, se mapean los ids obtenidos anteriormente y la función devuelve, por último, la lista con los títulos de las k películas más cercanas a la película de interés.

Como nota adicional, se debe tener cuidado con el número de vecinos obtenidos, pues es necesario excluir la película con la que se está trabajando, ya que no existirá alguien más cercano a la película que la propia película en sí. Es necesario aumentar a k en 1 y posteriormente eliminar la película de la lista de los vecinos más cercanos.

Para optimizar y ampliar la funcionalidad del sistema, se pueden considerar distintas mejoras en varios aspectos del sistema, con el uso de métricas alternativas, como la distancia cosenoidal, o la correlación de Pearson. Aunado a esto, otro punto a considerar pudiera ser la normalización de los datos, por ejemplo, en un rango entre $[0, 1]$ o implementar técnicas para la reducción de la dimensionalidad de la matriz; y por supuesto, la implementación de mejoras en el modelo para su escalabilidad.

El modelo KNN es una herramienta poderosa y flexible para sistemas de recomendación,

especialmente en el contexto de filtrado colaborativo. Su simplicidad y efectividad lo convierten en una excelente opción para encontrar ítems similares, como películas, basándose en las interacciones de los usuarios. Sin embargo, es importante considerar las limitaciones del modelo, como su escalabilidad y el manejo de datos dispersos, y complementarlo con técnicas avanzadas para mejorar su rendimiento.

3.2. Modelo KNN con métrica de Hausdorff

Una vez que se ha descrito cómo funciona el modelo KNN en la sección anterior (3.1), y continuando con los objetivos de esta investigación, se propone modificar el modelo KNN, en este caso, con métricas alternativas. El proceso es similar al KNN común, con la única diferencia en la métrica que se utiliza, que en este caso, es la métrica (común) de Hausdorff, que mide la proximidad entre dos conjuntos de puntos.

Como se vio en la Sección 1.6, la distancia de Hausdorff entre dos conjuntos finitos A y B se define como:

$$HD(A,B) = \max\left(\max_{a\in A}\min_{b\in B} d(a,b), \max_{b\in B}\min_{a\in A} d(b,a)\right),$$

donde el conjunto A tiene como puntos las calificaciones no nulas de la película de interés y B representa cualquier otra película; $d(a,b)$ es la distancia entre los puntos a y b (calificaciones no nulas de ambas películas). En este caso, se usa la norma euclidiana $d(a,b) = \|a-b\|$.

En términos simples:

1. Para cada calificación no nula en A, se encuentra la calificación más cercana en B y se toma la máxima de estas distancias.

2. Para cada calificación no nula en B, se encuentra la calificación más cercana en A y se toma la máxima de estas distancias.

3. La distancia de Hausdorff es el máximo de estos dos valores.

En conclusión, se utiliza la distancia de Hausdorff para medir la similitud entre películas basándose en sus calificaciones no nulas. La distancia de Hausdorff captura la máxima distancia mínima entre los conjuntos de calificaciones de dichas películas, lo que permite identificar películas con patrones de calificación similares y, finalmente, se seleccionan las k películas con la menor distancia de Hausdorff como las recomendadas.

3.3. Modelo KNN con métrica modificada de Hausdorff

La modificación que se se ha hecho en esta investigación al modelo KNN, es la aplicación de la métrica de Hausdorff modificada, vista en la sección 1.6, es decir:

La métrica de Hausdorff creada con la métrica dirigida:

$$d(\mathcal{A}, \mathcal{B}) = \frac{1}{N_A} \sum_{a \in A} d'(a, \mathcal{B}),$$

y la métrica no dirigida:

$$f(d(\mathcal{A}, \mathcal{B}), d(\mathcal{B}, \mathcal{A})) = \text{máx}(d(\mathcal{A}, \mathcal{B}), d(\mathcal{B}, \mathcal{A})).$$

Aquí, el conjunto A tiene como puntos las calificaciones no nulas de la película de interés y B representa cualquier otra película.

El procedimiento para obtener la distancia utilizando la métrica de Hausdorff modificada es como sigue:

1. Se miden las distancias $d'(a, b)$ (en este caso, se usa la norma euclidiana $\|a - b\|$), y se toma el mínimo de los valores, esa será $d'(a, \mathcal{B})$.
2. El procedimiento se realiza para cada $a \in \mathcal{A}$
3. Se suman todas las $d'(a, \mathcal{B})$ y se dividen entre la cardinalidad de $\mathcal{A}$, (N_A), esa será $d(\mathcal{A}, \mathcal{B})$
4. Se repiten los tres pasos anteriores utilizando $\mathcal{B}$, se obtendrá $d(\mathcal{B}, \mathcal{A})$
5. La distancia de Hausdorff es el máximo de estos dos valores.

Lo que se busca con esta modificación a la métrica es que se haga más robusto al modelo a datos atípicos, que en este caso son las películas con muy pocos ratings.

3.4. Validación

Una vez que han sido creados y modificados los modelos KNN, es necesario tener un punto de referencia que permita comparar y evaluar el desempeño de estos, para poder determinar el que cumple de mejor forma el objetivo de esta investigación. La calidad de un algoritmo de recomendación puede evaluarse mediante distintos tipos de medidas, que pueden ser la precisión o cobertura. El tipo de métrica utilizada depende del tipo de técnica de filtrado.

La *precisión* (accuracy, en inglés) es la fracción de recomendaciones correctas del total de recomendaciones posibles, mientras que la *cobertura* (coverage, en inglés) mide la fracción de objetos en el espacio de búsqueda que el sistema es capaz de ofrecer como recomendaciones. La idoneidad de cada métrica depende de las características del conjunto de datos y del tipo de tareas asignadas al sistema de recomendación.

Las métricas para el accuracy evalúan la precisión de una técnica de filtrado comparando los ratings predichos por el modelo con los ratings reales.

Lo anterior presenta un nuevo reto: la correcta precisión de las recomendaciones. De acuerdo con la Sección 1.1, los modelos de filtrado colaborativo logran captar las interacciones entre

usuarios e ítems que se generan por las valoraciones que los usuarios dan a los ítems. No obstante, muchas de estas valoraciones son producto de efectos asociados de forma unilateral a los usuarios o a los ítems independientemente de su interacción; es decir, en muchas ocasiones, tanto las valoraciones que otorga un usuario en particular o que recibe algún ítem en específico presentan tendencias sistemáticas que son producto de sesgos propios del usuario y del ítem, lo que presenta una mayor complejidad a la hora de predecir el rating que un usuario específico puede dar a un ítem en particular.

En [Koren, 2009], se propone una forma para aislar estos efectos que no implican la interacción *usuario-artículo*. Utilizan *baseline predictors*, que son modelos simples que se utilizan como punto de referencia para comparar el rendimiento de modelos más complejos.

Dado que estos predictores tienden a capturar gran parte de la señal observada, es fundamental modelarlos con precisión. Esto permite aislar de forma más precisa la interacción *usuario-artículo*, y someterla a modelos de preferencias de usuario más adecuados. A continuación se explica la propuesta del predictor introducido en [Koren, 2009], que resultó ser el idóneo en esta investigación.

Denotando primero a μ como la media total de las valoraciones, se define b_{ui} como la predicción para la valoración r_{ui} que el usuario u otorga al ítem i de la siguiente forma:

$$b_{ui} = \mu + b_u + b_i,$$

en donde los parámetros b_u y b_i indican las desviaciones de la media observadas del usuario u y del ítem i respectivamente, es decir:

$$b_i = \frac{\sum_{u \in R(i)} (r_{ui} - \mu)}{\lambda_1 + |R(i)|},$$

$$b_u = \frac{\sum_{i \in R(u)} (r_{ui} - \mu - b_i)}{\lambda_2 + |R(u)|},$$

de donde $R(i)$ representa al conjunto de todos los usuarios que han calificado a i y $R(u)$ a todos los ítems que han sido calificados por u; además, λ_1 y λ_2 son parámetros de regularización que fueron determinados por un conjunto de prueba en el experimento de [Koren, 2009].

Una vez definido el predictor que se ha utilizado en en esta investigación, es posible utilizar distintas métricas, como el error medio absoluto (MAE), el error cuadrático medio ($RMSE$) y la correlación, que suelen utilizarse como métricas estadísticas del accuracy. El MAE es el más popular y es una medida de la desviación de la recomendación con el valor específico del usuario. Se define como sigue ([Jofra and Gómez, 2019]):

$$MAE = \frac{1}{N} \sum_{u,i} |b_{ui} - r_{ui}|,$$

donde b_{ui} es el rating predicho por el modelo para el usuario u del ítem i, r_{ui} es el rating real y

N es el número total de ratings del ítem i. Cuanto menor sea el MAE, mayor será la precisión con la que el motor de recomendación prediga los ratings de los usuarios.

El error cuadrático medio (RMSE) viene dado por:

$$RMSE = \sqrt{\frac{1}{N}\sum_{u,i}(b_{ui} - r_{ui})^2}.$$

El RMSE pone más énfasis en el mayor error absoluto y cuanto menor sea, mejor será la precisión de la recomendación.

Otro tipo de métricas son las conocidas como *Decision support accuracy metrics* y las más utilizadas son la tasa de reversión (Reversal rate), los errores ponderados (Weighted errors), la curva ROC (Receiver Operating Characteristics) y la de precisión/recuperación (Precision/Recall, PRC), Precision, Recall y la Medida-F (F-Score). Estas métricas ayudan a los usuarios a seleccionar elementos de muy alta calidad entre el conjunto disponible y consideran el procedimiento de predicción como una operación binaria que distingue los elementos buenos de los que no lo son.

La precisión (*precision* en inglés) es la fracción de productos recomendados que fueron relevantes para el usuario, mientras que el recall puede definirse como la fracción de elementos relevantes que también forman parte del conjunto de elementos recomendados. Se calculan como:

$$\text{Precision} = \frac{\text{Ítems correctamente recomendados}}{\text{Total de ítems recomendados}},$$
$$\text{Recall} = \frac{\text{Ítems correctamente recomendados}}{\text{Total de ítems útiles}}.$$

Para evaluar el modelo derivado de esta investigación se elegirán las métricas anteriores adecuadas para calcular su eficiencia.

3.5. Simulaciones

Para esta sección, se exhiben una serie de simulaciones, en donde se busca trabajar con las matrices de utilidad obtenidas en la Seccion 2.2; es decir, D, D_{norm}, $D_{bayesian}$. La matriz D será utilizada para los modelos KNN con la métrica euclidiana, KNN con la métrica de Hausdorff común (que implícitamente utiliza la métrica euclidiana) y el modelo KNN que utiliza la métrica modificada de Hausdorff (hace uso también de la métrica euclidiana).

La Tabla 3.1 representa de forma gráfica los tres modelos recomendadores, las métricas que utilizaron y las matrices con las que fueron alimentados (entrenados).

Acrónimo	**Métrica**	**Matriz**
KNN E.	Euclidiana	D
KNN M. N.	Euclidiana	D_{norm}
KNN M. B.	Euclidiana	$D_{bayesian}$
KNN H.	Euclidiana	D
KNN H. M.	Euclidiana	D

Tabla 3.1: Tabla de Acrónimos, Métricas y Matriz

A continuación se explica el significado de cada acrónimo:

- **KNN E**: Modelo KNN que utiliza la métrica euclidiana y la matriz sin procesar D.
- **KNN M. N.**: Modelo KNN que utiliza la métrica euclidiana y la matriz normalizada D_{norm}.
- **KNN M. B.**: Modelo KNN que utiliza la métrica euclidiana con la matriz normalizada con la media bayesiana $D_{bayesian}$.
- **KNN H.**: Modelo KNN que utiliza la métrica de Hausdorff común y la matriz sin procesar D.
- **KNN H. M.**: Modelo KNN que utiliza la métrica de Hausdorff modificada y la matriz sin procesar D.

Para poder llevar a cabo esta prueba, se buscaron de forma aleatoria, mediante una función de Python, 30 índices de películas dentro de la matriz de utilidad; cada índice fue mapeado con los diccionarios inversos que permiten conocer el id de cada película.

El siguiente paso en el experimento fue obtener, para cada uno de los ids de las 30 películas, sus 10 vecinos más cercanos, tomando como parámetro los ratings que recibieron.

Finalmente, se muestran ejemplos de las distintas recomendaciones dadas por los distintos modelos recomendadores para el título *persona* de 1996:

El modelo $KNN\ E.$ arrojó los resultados en la tabla 3.2.

ID	**Título**
1	Fearless Vampire Killers, The (1967)
2	Hour of the Wolf (Vargtimmen) (1968)
3	Tenant, The (Locataire, Le) (1976)
4	Virgin Spring, The (Jungfrukällan) (1960)
5	Silence, The (Tystnaden) (1963)
6	Diary of a Chambermaid (Journal d'une femme de chambre, Le) (1964)
7	Contempt (Mépris, Le) (1963)
8	Woyzeck (1979)
9	Pierrot le fou (1965)
10	Short Film About Killing, A (Krótki film o zabijaniu) (1988)

Tabla 3.2: Películas recomendadas por KNN E.

El modelo $KNN\ M.N.$ arrojó las recomendaciones de la tabla 3.3.

ID	**Título**
1	Fearless Vampire Killers, The (1967)
2	Virgin Spring, The (Jungfrukällan) (1960)
3	Diary of a Chambermaid (Journal d'une femme de chambre, Le) (1964)
4	Short Film About Killing, A (Krótki film o zabijaniu) (1988)
5	Hairdresser's Husband, The (Le mari de la coiffeuse) (1990)
6	Wild Strawberries (Smultronstället) (1957)
7	Pierrot le fou (1965)
8	How to Steal a Million (1966)
9	Mon Oncle (My Uncle) (1958)
10	Play It Again, Sam (1972)

Tabla 3.3: Películas recomendadas por KNN M. N.

El modelo $KNN\ M.\ B.$ sugirió las recomendaciones de la tabla 3.4.

ID	Título
1	Fearless Vampire Killers, The (1967)
2	Tenant, The (Locataire, Le) (1976)
3	Hour of the Wolf (Vargtimmen) (1968)
4	Virgin Spring, The (Jungfrukällan) (1960)
5	Pierrot le fou (1965)
6	Diary of a Chambermaid (Journal d'une femme de chambre, Le) (1964)
7	Avventura, L' (Adventure, The) (1960)
8	Short Film About Killing, A (Krótki film o zabijaniu) (1988)
9	Silence, The (Tystnaden) (1963)
10	Hairdresser's Husband, The (Le mari de la coiffeuse) (1990)

Tabla 3.4: Películas recomendadas por KNN M. B.

El modelo *KNN H.* recomendó las películas de la tabla 3.5.

ID	Título
1	Practical Magic (1998)
2	Everything Is Illuminated (2005)
3	Penelope (2006)
4	Thing, The (2011)
5	Ruby Sparks (2012)
6	Sudden Death (1995)
7	Balto (1995)
8	Four Rooms (1995)
9	Othello (1995)
10	Now and Then (1995)

Tabla 3.5: Películas recomendadas por KNN H.

El modelo *KNN H. M.* determinó que las mejores recomendaciones son las de la tabla 3.6.

ID	Título
1	Practical Magic (1998)
2	Everything Is Illuminated (2005)
3	Penelope (2006)
4	Thing, The (2011)
5	Ruby Sparks (2012)
6	Capote (2005)
7	Porco Rosso (Crimson Pig) (Kurenai no buta) (1992)
8	Princess and the Frog, The (2009)
9	Singles (1992)
10	Loser (2000)

Tabla 3.6: Películas recomendadas por KNN H. M.

Para guardar la información resultante de este proceso, se utilizó un diccionario en Python, donde cada clave (*key*) del diccionario era el id de la película de interés, y el valor que guarda esa key es una lista con los ids de las recomendaciones correspondientes a esa película. Dicho proceso fue realizado para cada uno de los modelos recomendadores; teniendo en total 5 diccionarios con 300 películas cada uno.

Posterior a ello, se creó un diccionario que tiene como keys a los ids de las películas de interés. Los valores de esas keys son, de nuevo, un diccionario correspondiente a cada una de ellas. Ahora bien, puesto que a cada película de interés se le ha asignado un diccionario, éste se rellena con keys que guardan como valores, nuevamente en forma de diccionario, cada id de la recomendación correspondiente a la película de interés. Este último diccionario anidado tiene como valores una lista de dos listas; la primera lista guarda las predicciones para cada una de las calificaciones que recibió esa recomendación, mientras que la segunda lista guarda las calificaciones reales que obtuvo esa recomendación.

El paso anterior se realiza para cada uno de los modelos; y así, una vez que se tiene para cada modelo recomendador el diccionario con las calificaciones y predicciones de las recomendaciones, se procede a crear un DataFrame que las contenga a todas de forma organizada, para facilitar su posterior análisis. En dicho DataFrame, se han agregado también las métricas de evaluación de los modelos recomendadores vistas en la Sección 3.4, como lo son el Error Absoluto Medio (MAE), el Error Cuadrático Medio (MSE) y la Raíz del Error Cuadrático Medio (RMSE).

Es claro que la idea de organizar la información en diccionarios anidados no es la más adecuada, pues es complejo manejar cada nivel de anidación, así como el uso de recursos de la computadora y la complejidad que puedan tener los algoritmos derivados de usar estos diccionarios. Sin embargo, por la dificultad que presenta este problema, el buscar una forma de almacenamiento más óptima se plantea como un trabajo futuro; los fundamentos proporciona-

dos por el presente trabajo de tesis brindan las bases suficientes.

Con todo lo anterior, se presenta la Tabla 3.7 como una forma de visualizar el DataFrame recién creado. Dicha tabla no es una representación exacta y presenta valores y columnas faltantes; lo anterior debido a que el DataFrame creado posee demasiadas columnas y filas.

Rec	**DF**	**Método**	**ID**	**MAE**	**RMSE**	**Ratings**
		KNN E	95088	0.7046	0.8303	5
		KNN H	41	0.1751	0.1751	–
rec_1	df_128360	KNN H M	41	0.1751	0.1751	–
		KNN M N	95088	0.7046	0.8303	5
		KNN M B	95088	0.7046	0.8303	5

Tabla 3.7: Resultados de recomendaciones (5 métodos).

Una vez se ha obtenido el DataFrame, es posible empezar con el análisis del comportamiento de los modelos recomendadores mediante sus primeras y quintas películas recomendadas, principalmente si son consistentes.

Primera recomendación por método

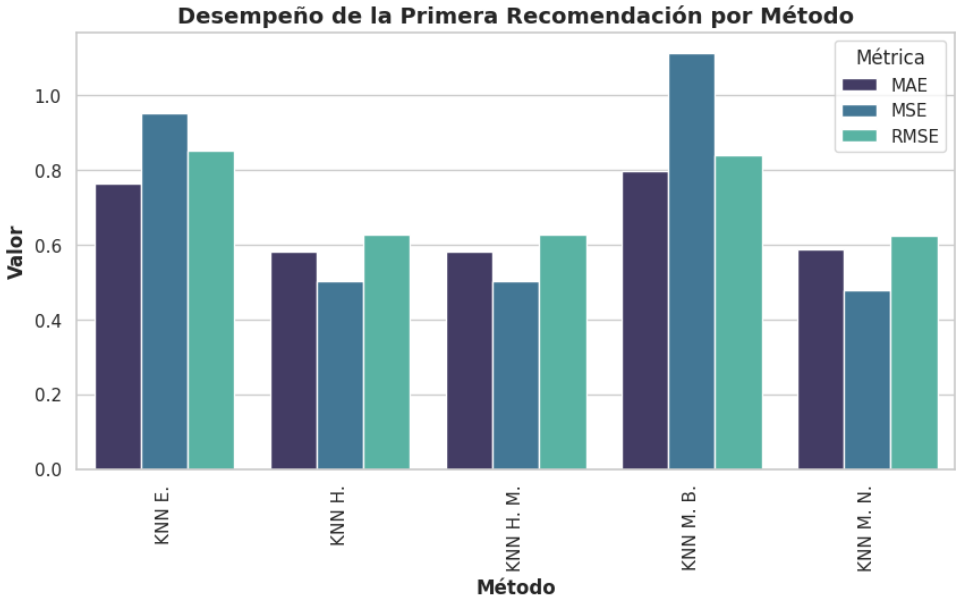

Figura 3.1: Desempeño de la primera recomendación por método.

Consideremos la primera película recomendada. La gráfica de la Figura 3.1 muestra, en primer lugar, que en todos los modelos, el MAE es siempre menor que el RMSE, lo cual es esperable porque el RMSE amplifica errores grandes debido a la elevación al cuadrado en el

MSE; sin embargo, para los modelos KNN M. B. y KNN E. la diferencia entre MSE y RMSE es más grande, lo que indica que estos modelos presentan errores con mayor dispersión.

También es posible observar que el modelo KNN M. B. presenta los valores más altos para las tres métricas (MAE, MSE y RMSE), lo que indica que tiene el peor desempeño en cuanto a la primera recomendación; mientras que los modelos KNN H., KNN H. M. y KNN M. N. tienen valores más consistentes y bajos en todas las métricas, lo que sugiere un mejor desempeño en comparación con KNN E. y KNN M. B.

Consistencia de la primera recomendación por método

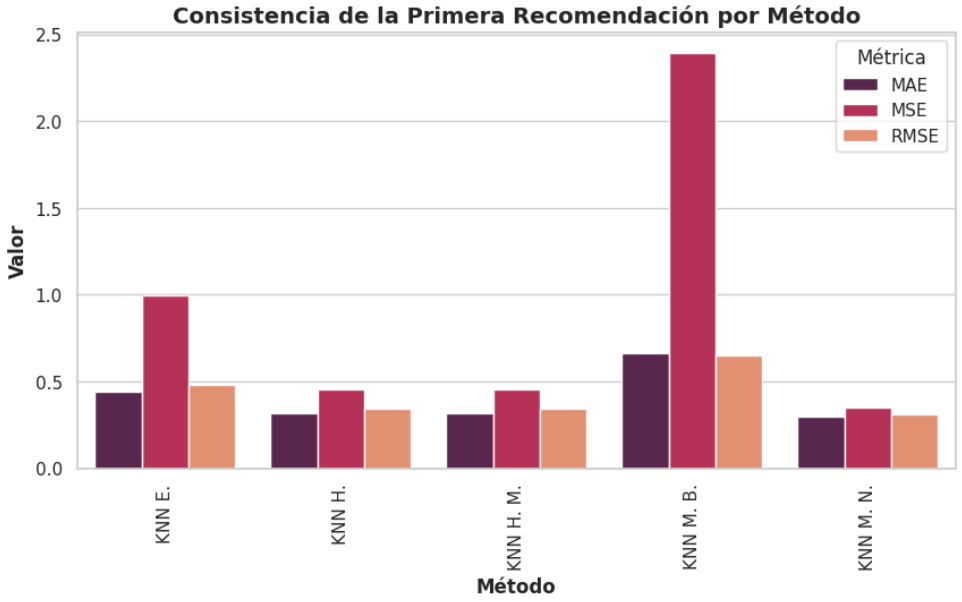

Figura 3.2: Consistencia de la primera recomendación por método.

La gráfica 3.2 muestra la consistencia de la primera recomendación por método, evaluada con las mismas métricas MAE, MSE y RMSE. A partir de ella, se observa que el método KNN M. B. posee un MSE extremadamente alto en comparación con el resto de los modelos. Esto es, que el modelo sufre de una gran dispersión en los errores y genera que algunas de sus recomendaciones sean muy inexactas; además, su RMSE y MAE son altos, lo que refuerza la idea de que este modelo es el menos confiable y menos consistente.

El modelo KNN E. tiene el segundo mayor MSE y RMSE, lo que indica que este modelo también genera recomendaciones con errores más grandes y menos consistentes en comparación con otros métodos.

Por otra parte, se observa mayor consistencia en KNN H., KNN H. M. y KNN M. N., pues todos ellos tienen valores bajos y similares en todas las métricas, lo que sugiere que son los más estables en cuanto a la calidad de sus recomendaciones, de donde KNN M. N. parece ser el más consistente de todos, ya que tiene los valores más bajos en todas las métricas.

En la gráfica 3.1, el modelo KNN M. B. ya mostraba un mal desempeño en precisión, pero aquí queda claro que también es el menos consistente, mientras que KNN H., KNN H. M. y KNN M. N. no solo eran mejores en desempeño en la gráfica 3.2, sino que también son más consistentes en esta, lo que los hace las mejores opciones.

Si un modelo tiene buen desempeño pero es inconsistente, puede no ser fiable en escenarios reales. En este caso, KNN M. B. es el peor en ambos aspectos y debería evitarse.

Quinta recomendación por método

A continuación, para poder tener un panorama más amplio, a fin de poder revisar que los resultados de los distintos modelos tengan congruencia con aquello observado en la primera recomendación, y dado que el número 5 corresponde a la mitad del número de recomendaciones por cada película, se ha decidido aplicar las mismas métricas que en la primera recomendación, con la particularidad de que se aplicarán a la quinta película recomendada.

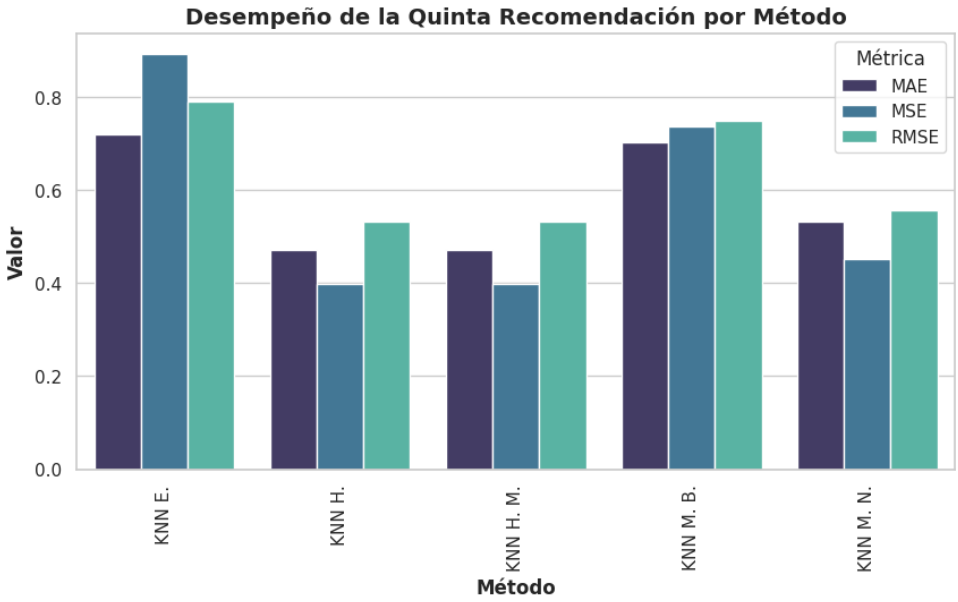

Figura 3.3: Comportamiento de la desviación estándar de los ratings.

Ahora, la gráfica de la Figura 3.3 muestra el desempeño de la quinta recomendación por método. En general, los modelos mantienen un patrón de desempeño esperado al que mostraron en la primera recomendación; es decir, el patrón de errores se mantiene similar entre la primera y la quinta recomendación en todos, por ejemplo, el MAE aumenta en todos los métodos, y es congruente, ya que la quinta recomendación está "más lejos", es decir, que todos los modelos tengan un desempeño más bajo respecto a la primera recomendación.

También se observa que KNN E. y KNN M. B. siguen teniendo los peores errores en todas las métricas, mientras que KNN H., KNN H. M. y KNN M. N. mantienen un mejor desempeño con valores más bajos en todas las métricas. Se observa una mejora en KNN M. B., que tenía

un desempeño pésimo en la primera recomendación; ahora, aunque sigue siendo malo, KNN E. tiene un error similar o incluso peor en algunos casos.

Por otra parte, KNN H., KNN H. M. y KNN M. N. siguen siendo los mejor evaluados, con KNN M. N. con los valores de error más bajos.

Consistencia de la quinta recomendación por método

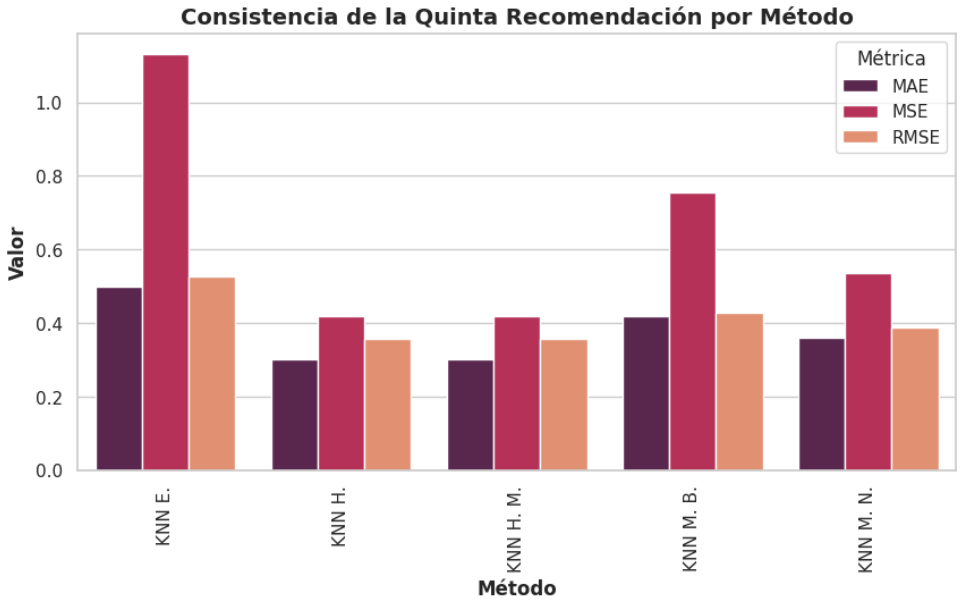

Figura 3.4: Consistencia de la quinta recomendación por metodo.

La gráfica de la Figura 3.4 muestra cómo el modelo KNN E. tiene una mayor variabilidad en el MAE y RMSE. Por otro lado, el modelo KNN H. M. y KNN H. mantienen una menor variabilidad en el comportamiento de sus recomendaciones, incluida en esta quinta, por lo que son más confiables que el resto. Se observa también cómo el modelo KNN M. N. mantiene una ligera desventaja con respecto a KNN H. y KNN H. M. La quinta recomendación es menos fiable que la primera en todos los métodos; sin embargo, el comportamiento de todos es el esperado; además, en la mayoría de los casos, el MAE y el RMSE tienen valores similares, lo que indica que los errores promedio no varían demasiado entre los métodos.

Por último, el modelo KNN E. es el menos consistente de todos ellos, por lo que lo vuelve el menos confiable para las recomendaciones en esta posición. Esto reafirma, tanto para la primera como para la quinta recomendación, que el método KNN E. es el menos apto de los cinco para emitir recomendaciones; y que los métodos KNN H., KNN H. M. y KNN M. N. son las opciones más sólidas.

Evaluación por películas mediante el MAE

Una vez se ha realizado el análisis de la primera y quinta recomendación en general para todas las películas de interés, se desea evaluar el desempeño de todas las películas a las que se les realizaron las recomendaciones. Para lograrlo, se ha creado una gráfica que compara, para cada una de las películas de los títulos aleatorios, el comportamiento del MAE respecto a cada modelo recomendador, y se muestra en la Figura 3.5.

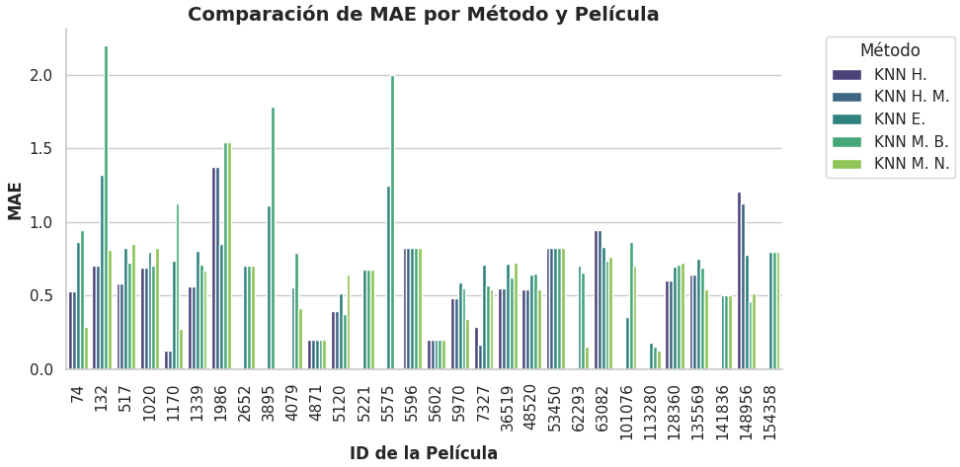

Figura 3.5: Comparación del MAE de todas las películas y todos los métodos.

Una vez más se confirma que los modelos KNN H. M. y KNN H. son los que muestran los MAEs con valores incluso cercanos a 0 en algunos casos, lo que sugiere que estos métodos calculan mejor las preferencias de los usuarios.

Por contraparte, los modelos que tuvieron un comportamiento irregular y con un MAE elevado fueron los modelos KNN E., KNN M. N. y KNN M. B. En la gráfica anterior (3.5) también se observa que algunas películas presentan valores de MAE significativamente más altos que otras, lo que sugiere que ciertos títulos son más difíciles de predecir con precisión, especialmente las películas 132, 1986, 3895, 5575, 148956 muestran picos elevados en comparación con el resto.

Evaluación por películas mediante el RMSE

Continuando con la evaluación, se procede a analizar el RMSE de todos los modelos para cada uno de los títulos de interés. La Figura 3.6 muestra que el RMSE es más alto que MAE en todos los casos, debido a la naturaleza cuadrática, esto provoca que haya un mayor impacto de los datos atípicos, lo cual es beneficioso para esta investigación. En este caso, el modelo KNN H. M. sigue siendo el mejor, pero la diferencia con otros métodos se reduce.

Una vez más, los títulos que presentan mayor MSE son los 132, 1986, 3895, 5575, 148956, mostrando consistencia con MAE. Estos títulos requieren de mayor atención debido a que cabe la posibilidad de que sean atípicos; se plantea como trabajo posterior hacer este estudio.

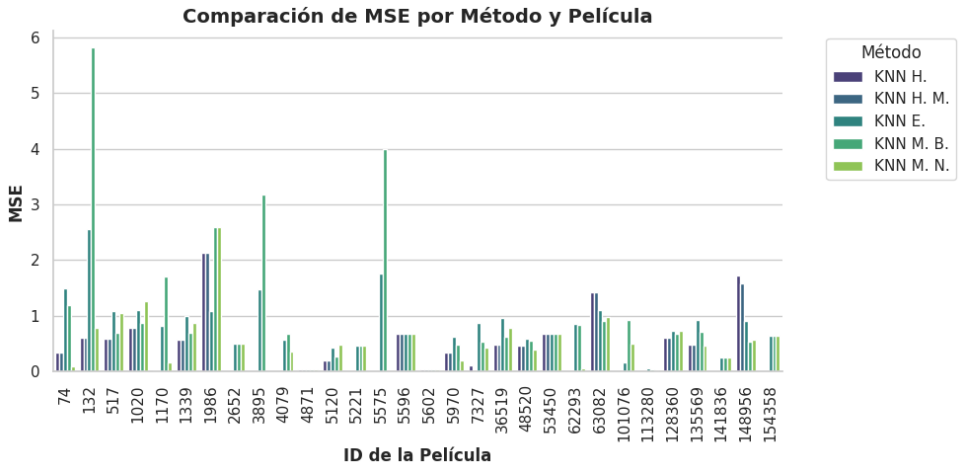

Figura 3.6: Comparación del MSE de todas las películas y todos los métodos.

Evaluación por películas mediante el RMSE

El RMSE es la raíz cuadrada del MSE, por lo que tiene una interpretación más intuitiva, ya que mantiene las mismas unidades que los valores originales. A diferencia del MSE, el RMSE suaviza los valores extremos, pero sigue penalizando errores grandes más que el MAE.

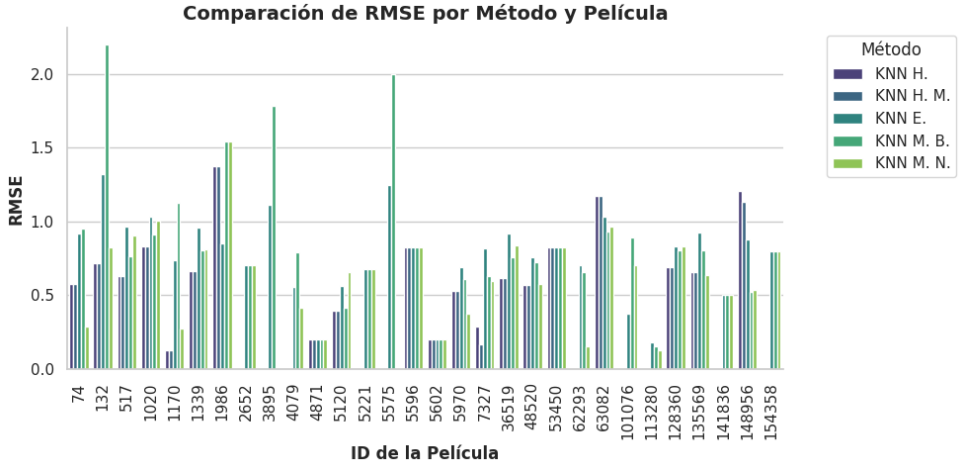

Figura 3.7: Comparación del RMSE de todas las películas y todos los métodos.

En la gráfica de la Figura 3.7 se observan patrones similares a los vistos en la Gráfica de MSE (3.6), por ejemplo, las películas con ID 132, 1986, 3895, 5575, 148956, muestran los valores de RMSE más altos, lo que indica que en estas películas los errores son más significativos.

En este caso, los métodos KNN M. B. y KNN E. siguen presentando algunos de los valores más altos de error en ciertas películas, sugiriendo mayor variabilidad en su precisión; mientras que los métodos KNN H. y KNN H. M. parecen tener RMSE más bajos en comparación con otros métodos. En particular, si una película tiene RMSE alto en todos los métodos, puede ser un caso difícil de predecir debido a características intrínsecas de los datos.

Para contar con una perspectiva más amplia sobre los detalles de las películas de interés, se muestra la Tabla 3.8, que servirá como guía para determinar qué película podría ser considerada como un dato atípico.

movieId	**n ratings**
74	8
132	6
517	9
1020	37
1170	2
1339	29
1986	2
2652	1
3895	1
4079	2
4871	1
5120	3
5221	1
5575	1
5596	1
5602	1
5970	10
7327	6
36519	16
48520	6
53450	1
62293	4
63082	71
101076	3
113280	1
128360	18
135569	15
141836	1
148956	4
154358	1

Tabla 3.8: Ids de las películas de interés y número de ratings recibidos.

Desempeño promedio por método

Una vez realizado el análisis de forma particular para cada una de las películas de interés, es prudente realizar ahora una comparación general de cada uno de los métodos.

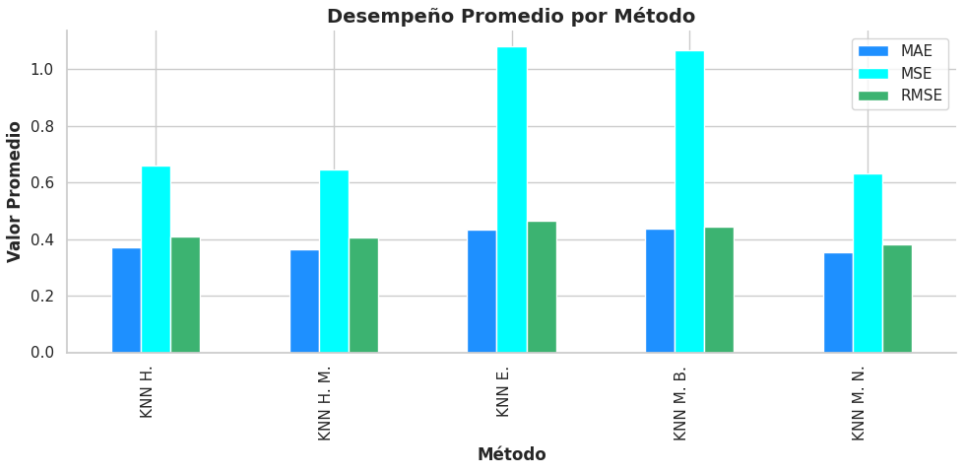

Figura 3.8: Desempeño general por método respecto a los errores.

En la Figura 3.14 se observa, en primer lugar, que el MSE es la métrica con valores más altos en cada una de las variantes de los modelos, lo que es de esperarse, ya que es el error cuadrático medio. Por otra parte, el RMSE tiene valores similares a MAE en la mayoría de los métodos, pero siempre ligeramente superiores debido a su relación matemática con el MSE. Mientras que el MAE es la métrica con valores más bajos, ya que mide el error absoluto sin penalizar errores grandes tanto como MSE.

Ahora, existen diferencias en cada uno de los métodos; por ejemplo, KNN E. y KNN M. B. tienen los valores más altos de MSE, lo que indica todo aquello que se observó anteriormente, que generan errores más grandes en la mayoría de los datos. Así también, los modelos KNN H. y KNN H. M. muestran valores más bajos en todas las métricas, lo que sugiere un mejor desempeño en términos de precisión.

Distribución MAE Promedio

Lo primero que se puede observar en la Gráfica 3.9 es que todos los modelos tienen una distribución bastante similar de MAE, y que la línea central de cada caja (la mediana) está en un nivel parecido, lo que indica que el MAE central es aproximadamente el mismo en todas las configuraciones. También todos los modelos tienen una dispersión parecida en la mayoría de los casos, sin una gran ventaja de uno sobre otro. Por otra parte, los modelos como KNN E. y KNN M. B. presentan más valores atípicos con MAE alto (> 1.5), lo que sugiere que en algunos casos pueden cometer errores considerables.

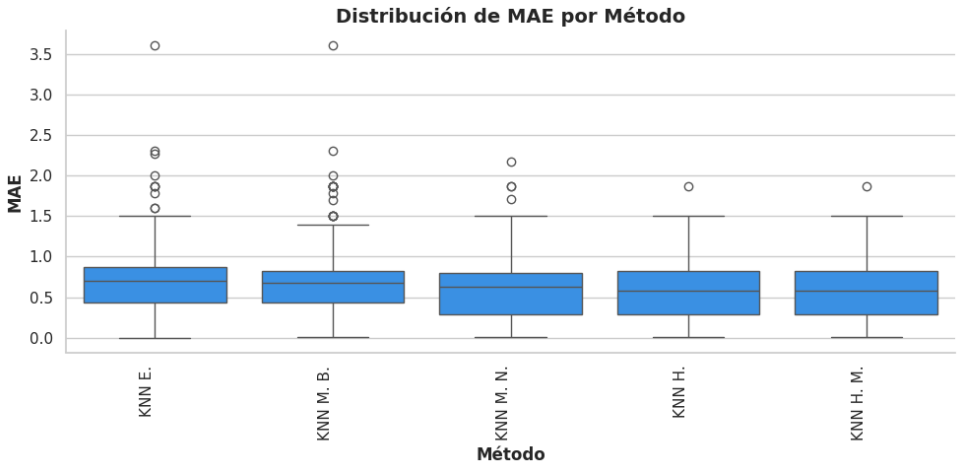

Figura 3.9: Comportamiento de la distribución del MAE por método.

Distribución MSE Promedio

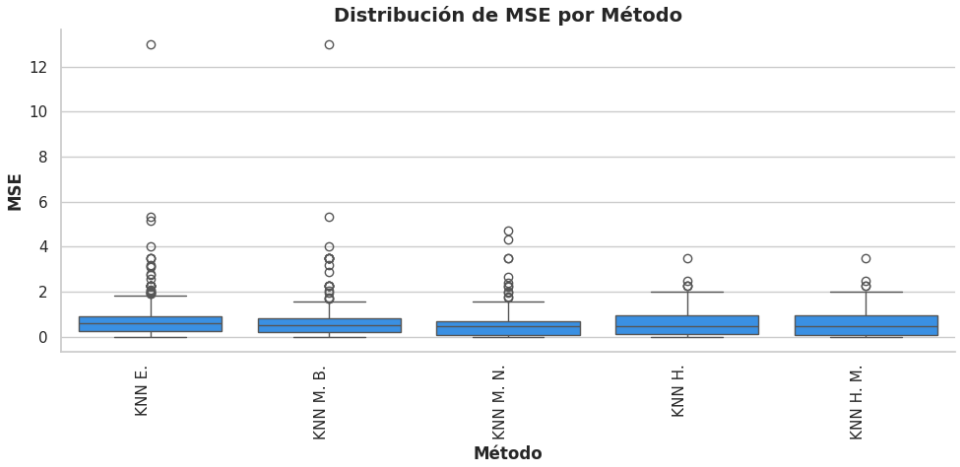

Figura 3.10: Comportamiento de la distribución del MSE por método.

En primer lugar, en la gráfica de la Figura 3.10 es posible observar que las medianas están muy cercanas entre los modelos, lo que indica que el MSE promedio es similar para todas las variantes de KNN; además de que los bigotes son relativamente cortos, esto último también sugiere que la mayoría de los valores de MSE están dentro de un rango controlado. Sin embargo, los outliers son frecuentes y algunos muy grandes, especialmente en KNN E. y KNN M. B., lo que indica que estas configuraciones pueden fallar drásticamente en ciertos casos de películas difíciles de recomendar.

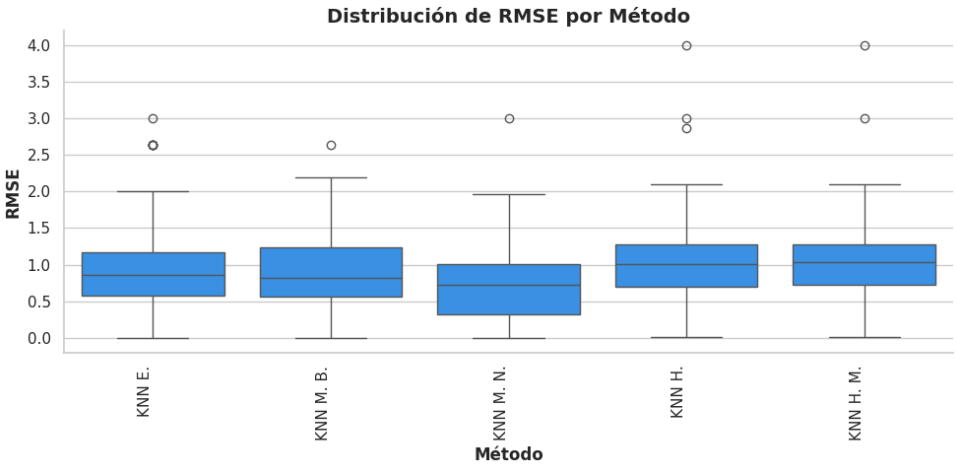

Figura 3.11: Comportamiento de la distribución del RMSE por método.

Distribución RMSE Promedio

Por último, se realizan los boxplots para el caso del RMSE que, como se observa en la Figura 3.10, las conclusiones obtenidas en todas las gráficas anteriores de esta sección son congruentes entre sí.

Además, se puede concluir que tanto en la primera y quinta recomendación en general para todas las películas, los modelos KNN H. y KNN. H. M. mostraron un mejor desempeño y una baja variabilidad en sus recomendaciones, lo que los vuelve más precisos y confiables. Se concluye también que los modelos con peor desempeño para las primeras y quintas recomendaciones fueron los modelos KNN E. y KNN M. B., siendo poco precisos y muy variables con sus recomendaciones.

Por último, en particular para cada título de interés, nuevamente los modelos KNN H. y KNN H. M. resultaron tener un mejor comportamiento en general, siendo los que menor número de malos comportamientos tuvieron para las películas, a pesar de que algunas fuesen datos atípicos. De forma contraria, nuevamente los modelos KNN E. y KNN M. B. fueron los que peor comportamiento tuvieron, siendo incluso aquellos con los que mayores errores globales.

Ahora bien, evaluando los errores de cada modelo, se notó que todos los errores, ya sea MAE, MSE, o RMSE, se comportaban de forma muy parecida para cada uno de los métodos, por lo que se puede concluir que, en efecto, las películas fueron seleccionadas al azar y que los resultados anteriores eran congruentes y justificables.

Precisión de los modelos por película

Para evitar sesgos con los resultados obtenidos anteriormente, se propone evaluar también la precisión de los modelos, y determinar, por cada película y cada modelo, el porcentaje de recomendaciones calificadas sobre todas las recomendadas.

La Gráfica 3.12 muestra algo un poco disruptivo respecto a los resultados anteriormente

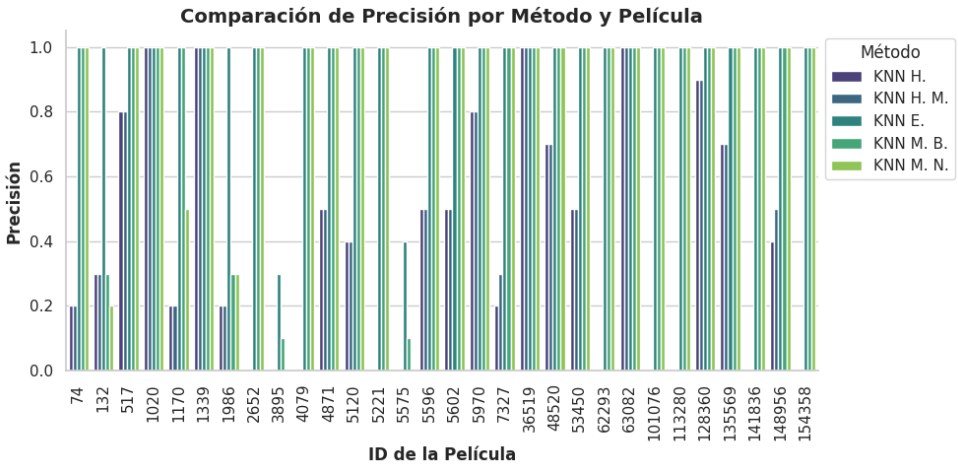

Figura 3.12: Precisión de los modelos por película.

vistos: Los modelos KNN H. y KNN H. M. no resultaron ser los más precisos; sino que, contrariamente a lo anterior, fueron los modelos KNN M. B. y KNN M.N aquellos con un mejor porcentaje de acierto.

Lo anterior sugiere que algunas películas tienen una alta variabilidad en la precisión entre métodos, lo que implicaría que ciertos modelos funcionan mejor en algunos casos que en otros.

Para otras películas, todos los métodos coinciden en una alta precisión, lo que sugiere que estas son más fáciles de predecir. El único resultado que se mantiene es que el modelo menos preciso, en términos generales, fue el modelo KNN E.

Ahora bien, descartando al modelo KNN E., el cual resultó ser consistentemente malo para cada una de las distintas métricas aplicadas, el resultado obtenido de la gráfica 3.12 se debe tomar con cuidado, pues el hecho de recomendar películas que no fueron vistas no implica siempre que fue una *mala* recomendación, sino también podría indicar que esos modelos tienen la capacidad de ofrecer recomendaciones *nuevas*, es decir, podrían ser robustos al problema de *cold start*. Por supuesto, para aseverar con mayor confianza esto último, se deben aplicar más pruebas, e incluso algunas que involucren usuarios reales.

Comparación de rendimiento entre Hausdorff

Por último, en esta investigación, debido a que intuitivamente se llegó a pensar que los modelos KNN H. y KNN H. M. llegarían a tener resultados parecidos, se ha decidido realizar un análisis que involucre exclusivamente a dichos modelos.

De forma manual se determinó que ambos modelos ofrecían en la gran mayoría de las películas de interés, las mismas recomendaciones. Determinando posteriormente, de forma algorítmica, que únicamente difieren en dos películas, aquellas con ids 7327 (con 5 recomendaciones distintas) y 148956 (con 2 recomendaciones distintas).

Por la poca cantidad de información con la que se cuenta para comparar los modelos, se ha

determinado compararlos con la película *Persona* de 1966, correspondiente al id 7327. Esta es la razón de mostrar la Gráfica 3.13.

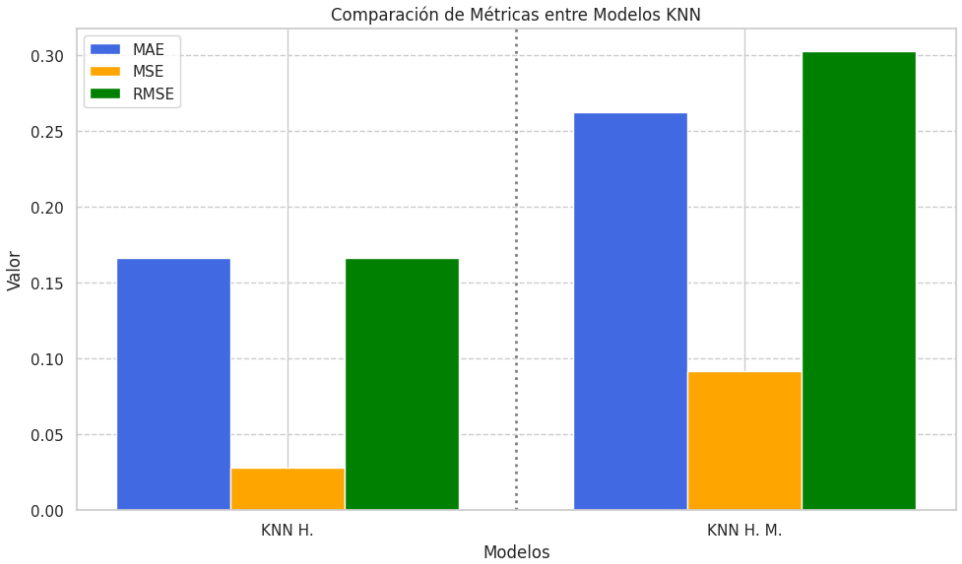

Figura 3.13: Desempeño de los dos modelos Hausdorff.

Los resultados observados en la Gráfica 3.13, junto con los obtenidos en la Figura 3.12, refuerzan que no son concluyentes los resultados para determinar cuál de los dos modelos es mejor que el otro, pues la precisión de ambos solo difiere en 0.1 puntos, poniendo en ventaja al modelo KNN H. M.; sin embargo, los errores son menores para el modelo KNN H., análogamente, como producto de tener una mejor precisión. Más aún, de todas las calificaciones recibidas por todas las recomendaciones emitidas, difieren en solo dos calificaciones (véase la Tabla 3.9).

Resultados de las calificaciones				
Recomendaciones	KNN H.		KNN H. M.	
No. Rec	Predicción	Calificación	Predicción	Calificación
1	–	–	–	–
2	4.166	4.0	4.166	4.0
3	–	–	–	–
4	–	–	–	–
5	–	–	–	–
6	–	–	4.52	4.0
7	4.166	4.0	3.692, 4.166	3.5, 4.0
8	–	–	–	–
9	–	–	–	–
10	–	–	–	–

Tabla 3.9: Resultados de predicción KNN.

Se le invita al lector a ser el primer sujeto de prueba de investigaciones futuras y comparar las Tablas 3.5 y 3.6 y decidir cuál modelo hace mejores recomendaciones.

Conclusión

El objetivo general de esta investigación fue plantear un modelo matemático de un sistema recomendador para una plataforma de streaming de películas, que lograra capturar la atipicidad observada en la base de datos de Netflix, y con ello, ser insensible a estos datos. Además, se propuso que el modelo estuviera respaldado en las teorías de las k-vecindades más cercanas y en la teoría de métricas de Hausdorff y, con todo ello, finalmente, realizar simulaciones y validar la certeza de sus recomendaciones. Durante el transcurso de la presente tesis se lograron plantear no solo uno, sino cinco distintos modelos (variantes) al cambiar la métrica de Hausdorff utilizada y al darle distintas modificaciones a la matriz de utilidad, los cuales fueron KNN E., KNN H., KNN H. M., KNN M. N. y KNN M. B, dichos modelos fueron evaluados a través de 30 películas aleatorias y un predictor de calificaciones. Posteriormente, se realizó un análisis del comportamiento de los errores en las recomendaciones que emitían cada uno para evaluar su desempeño.

Para lograr el objetivo general, en el Capítulo 1, se recopiló información referente a los sistemas recomendadores, tema principal de esta investigación, así como un recorrido por los principales conceptos trabajados en este proyecto, como son la Matriz de Utilidad, los modelos KNN, y la teoría requerida para comprender los conceptos de similitud y distancias; las métricas de Hausdorff; así como una introducción a la problemática de los datos atípicos.

En el Capítulo 2, se realiza un análisis exploratorio de la base de datos para determinar como se distribuyen los datos y, así, se determina la existencia de datos atípicos en ella, principal motivación para utilizar las métricas de Hausdorff en los modelos KNN. Se construyó también la matriz de utilidad y se realizaron modificaciones a la misma, a fin de poder evaluar más variantes de los modelos recomendadores y tener una investigación más completa.

En el Capítulo 3, se muestra el diseño del modelo y algoritmo de k-vecindades más cercanas para un sistema recomendador basado en ratings, utilizando la base de datos de Netflix y las variantes de la Matriz de utilidad construidas en el Capítulo 2. Con lo anterior, se implementaron diversas métricas modificadas de Hausdorff al modelo y se realizaron las simulaciones mediante la programación en el Notebook de Colaboratory, basado en el lenguaje Python. Por último, se validó el modelo y se discutió su robustez en la base de datos particular propuesta mediante el uso de métricas de errores.

Un resultado relevante es que los modelos que mejor comportamiento tuvieron durante las simulaciones fueron (*a priori*) el modelo KNN H. M., KNN H., KNN M.N., teniendo todos ellos

un desempeño muy similar. Estos modelos se destacaron por tener la mejor exactitud de sus recomendaciones respecto al predictor construido en el Capítulo 3.

Un punto a favor de los modelos KNN H.M. y KNN H. es que lograron igualar el desempeño del mejor de los otros tres modelos convencionales (KNN M. N.), sin la necesidad de tener que transformar los datos. No obstante, KNN H. M. y KNN H. resultaron no ser tan precisos, pues del total de recomendaciones que produjeron por cada película evaluada, solo una porción de ellas resultaban tener calificaciones.

Esta aparente incongruencia en los resultados obtenidos no puede ser tomada como una conclusión adversa a lo esperado en la hipótesis de la investigación. A partir de estos resultados se puede interpretar lo siguiente:

- En efecto, a pesar de igualar el desempeño del mejor de los modelos convencionales (KNN M. N.), los modelos KNN H.M. y KNN H. no logran tener una precisión alta en las películas que recomiendan, por lo que se descartan como una alternativa viable a los modelos convencionales.

- Al tener una precisión tan alta, el modelo KNN M. N. está sobreajustado para la base de datos con la que se trabajó en esta investigación; por otra parte, KNN H.M. y KNN H. al tener una precisión más baja, un desempeño similar a KNN M. N. y la ventaja de no requerir de una transformación previa de los datos, resultan ser más flexibles, incluso pudiendo tomar las recomendaciones no calificadas como nuevas opciones para recomendar, siendo así mejores opciones.

También, debido a que la diferencia entre las películas recomendadas entre los modelos KNN H.M y KNN H. resultó ser unicamente de siete películas de un total de trescientas, tampoco fue posible evaluar de forma contundente las diferencias en el desempeño y la precisión de estos modelos.

Por lo anterior, no es posible emitir una conclusión contundente de la presente investigación. Sin embargo, es posible sentar un precedente para futuras investigaciones y lineas de investigación respecto a este fascinante tema. Algunas líneas de investigación que darían continuidad a esta tesis son:

- Implementar *A/B testing* (pruebas A/B) entre los modelos KNN H, M. y KNN M, B. para medir impacto en usuarios reales y poder determinar con mayor veracidad y contundencia lo concluido en esta investigación.

- Realizar una combinación de métricas, matrices con tratamientos previos y métricas modificadas de Hausdorff, para poder tener un punto de referencia más amplio al momento de evaluar recomendadores.

El repositorio donde se encuentra la programación asociada a esta investigación se ubica en el siguiente código QR:

Figura 3.14: Código QR del colaboratory de esta investigación.

Con todo lo anterior, se concluye que se logra el objetivo general y se brinda un panorama dentro de la teoría de sistemas recomendadores que hacen uso de la matriz de utilidad, para posteriores referencias y como fuente de consulta para estudiantes de licenciatura y posgrado.

Bibliografía

[Aggarwal, 2015] Aggarwal, C. C. (2015). *Data Mining - The Textbook*. Springer.

[Alimohammadi and Chen, 2022] Alimohammadi, H. and Chen, S. N. (2022). Performance evaluation of outlier detection techniques in production time series: A systematic review and meta-analysis. *Expert Systems with Applications*, 191:116371.

[Boukerche et al., 2020] Boukerche, A., Zheng, L., and Alfandi, O. (2020). Outlier detection: Methods, models, and classification. *ACM Computing Surveys (CSUR)*, 53(3):1–37.

[Datalaria, 2019] Datalaria (2019). Casos de éxito: Netflix. `https://www.datalaria.com/post/casos_exito/2019-12-03-netflix/`. Consultado el 10 de octubre de 2023.

[Fernandez et al., 2005] Fernandez, J.-C., Mounier, L., and Pachon, C. (2005). A model-based approach for robustness testing. In Khendek, F. and Dssouli, R., editors, *Testing of Communicating Systems*, pages 333–348, Berlin, Heidelberg. Springer Berlin Heidelberg.

[Gorakala, 2016] Gorakala, S. K. (2016). *Building Recomendations Engines*. Packt Publishing, Birmingham, UK.

[Hodge and Austin, 2004] Hodge, V. and Austin, J. (2004). A survey of outlier detection methodologies. *Artificial intelligence review*, 22:85–126.

[Ilyas and Chu, 2019] Ilyas, I. F. and Chu, X. (2019). *Data Cleaning*. ACM Books.

[Jofra and Gómez, 2019] Jofra, X. and Gómez, A. (2019). Innovaciones tecnológicas en la cadena de suministro aplicadas al eCommerce. *Oikonomics*, pages 41–57.

[Koren, 2009] Koren, Y. (August 2009). The BellKor Solution to the Netflix Grand Prize. *Netflix Prize Document*, 81:1–10.

[Lehmann and Lösler, 2016] Lehmann, R. and Lösler, M. (2016). Multiple outlier detection: hypothesis tests versus model selection by information criteria. *Journal of surveying engineering*, 142(4):1–21.

[Olguín et al., 2019] Olguín, G. M., Jesús, Y. L. D., and de Celis Herrero, M. C. P. (2019). Métricas de similaridad y evaluación para sistemas de recomendación de filtrado colaborativo. *Revista de Investigación en Tecnologías de la Información*.

[Rawn, 2024] Rawn, E. (2024). The Work and Vision of Ubiquitous Computing at Xerox PARC. *IEEE Annals of the History of Computing*.

[Resnick et al., 1994] Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., and Riedl, J. (1994). Grouplens: An open architecture for collaborative filtering of netnews. In *Proceedings of the 1994 ACM conference on Computer supported cooperative work*, pages 175–186.

[Singh and Upadhyaya, 2012] Singh, K. and Upadhyaya, S. (2012). Outlier detection: applications and techniques. *International Journal of Computer Science Issues (IJCSI)*, 9(1):307.

[Smiti, 2020] Smiti, A. (2020). A critical overview of outlier detection methods. *Computer Science Review*, 38:100306.

[Technology, 2023] Technology, T. (2023). Estadísticas de abonados a netflix. `https://tridenstechnology.com/es/estadisticas-de-abonados-a-netflix/`. Consultado el 10 de octubre de 2023.

[Wang and Tan, 2012] Wang, J. and Tan, Y. (2012). Hausdorff distance with k-nearest neighbors. In Tan, Y., Shi, Y., and Ji, Z., editors, *Advances in Swarm Intelligence*, pages 272–281, Berlin, Heidelberg. Springer Berlin Heidelberg.

[Wikipedia, 2023] Wikipedia (2023). Netflix prize. `https://en.wikipedia.org/wiki/Netflix_Prize`. Consultado el 10 de octubre de 2023.