

# Group Recommender System for Restaurant Lunches

ERIK HALLSTRÖM



**KTH Datavetenskap  
och kommunikation**

Degree project in  
Computer Science  
Second cycle  
Stockholm, Sweden 2013



**KTH Computer Science  
and Communication**

# Sistema de recomendación de grupos para Almuerzos en restaurante

Tesis de Maestría en Ciencias de la Computación

Erik Hallström

Supervisor en Ericsson: Dr. Vincent A. Huang  
Supervisor en KTH: Profesor Anders Lansner  
Examinador en KTH: Profesor Jens Lagergren

15 de agosto de 2013

## Resumen

Se desarrolla un sistema de recomendaciones grupales para restaurantes de comida. La interfaz de usuario es una aplicación de Android que se ejecuta en un teléfono inteligente. El sistema presenta un enfoque novedoso para la recopilación de calificaciones implícitas cuando un usuario navega por una lista de descripciones de artículos. La recomendación individual se basa en extraer y comparar vectores de características tf-idf de textos de menú, así como clasificaciones individuales de los restaurantes. El sistema de recomendación de grupo funciona agregando las puntuaciones estimadas individuales de los miembros del grupo y, además de esto, se utilizan métodos de aprendizaje automático para capturar la dinámica del grupo en la decisión del grupo.

### Grupprekommenderingssystem för lunchrestauranger - Sammanfattning

Inom ramen för exámenessarbetet utvecklas ett system för grupprekommendation av lunchrestauranger. Användarapplikationen laddas ner och körs på en Android smartphone. Systemet använder sig av en ny metod för att samla implícita information från användarnas nyttjande av applikationen. Den individuella rekommendationen bygger på att extrahera och jämföra tf-idf feature-vektorer från menyernas texter samt individuell rangordning av restaurangerna. Grupprekommendationen fås genom att aggregata de individuella estimerade betygen hos användarna i gruppen, och därtill används maskininlärningsmetoder för att simulera gruppdynamiken av gruppvalet.

### Agradecimientos

Me gustaría expresar mi más sincero agradecimiento al Dr. Vincent Huang de Ericsson y al profesor Anders Lansner de KTH por ayudarme y guiarme a lo largo de esta tesis.

## Contenido

|  |           |
|--|-----------|
| I. Introducción  | 1         |
| 1. Propósito   | 1         |
| 2 Discusión sobre decisiones y elección                            | 2         |
| 3 Categorización de artículos                                      | 5         |
| 4 Antecedentes 4.1   | 7         |
| Sistemas de recomendación.   | 7         |
| 4.1.1 Colaborativo.  | 8         |
| 4.1.2 Basado en contenido.   | 14        |
| 4.2 Extensiones adicionales y técnicas de aprendizaje automático.  | dieciséis |
| 4.3 Sistemas de recomendación de grupos.                           | 17        |
| 4.4 Trabajo previo.  | 18        |
| 4.4.1 Recomendación individual.                                    | 18        |
| 4.4.2 Recomendación de grupo.                                      | 19        |
| 4.4.3 Calificaciones implícitas.                                   | 20        |
| Método II  | 21        |
| 5 Configuración del  | 21        |
| sistema 5.1 Aplicación de Android :                                | 21        |
| 5.2 Servidor.  | 24        |
| 5.3 Base de datos.   | 26        |
| 5.4 Analizador.  | 28        |
| 5.5 Sistema de recomendación.                                      | 29        |
| 6 Algoritmo de recomendación 6.1                                   | 29        |
| Discusión de la elección del método.                               | 29        |
| 6.2 Preferencia por el restaurante.                                | 32        |
| 6.3 Preferencia por la comida.                                     | 35        |
| 6.4 Popularidad mundial.   | 43        |
| 6.5 Agregación de resultados.                                      | 43        |
| 6.6 Recomendación de grupo.  | 44        |
| 6.6.1 Agregación de grupos.  | 44        |
| 6.6.2 Máquina de vectores de soporte para recomendación de grupos. | 44        |
| 6.6.3 Agregación de resultados de recomendadores de grupo.         | 47        |

|   |    |
|---|----|
| III Resultados  | 47 |
| 7 Rendimiento   | 48 |
| 7.1 Ranking de restaurantes consultados. . . . .                        | 48 |
| 7.2 Número de navegaciones en una sesión de navegación. . . . .         | 52 |
| 7.3 Rankings de elementos seleccionados y navegación de grupos. . . . . | 53 |
| 8 Datos de usuario  | 55 |
| 9 Análisis textual de los menús   | 58 |
| IV Discusión y trabajo futuro   | 63 |
| 10 Discusión  | 63 |
| 11 Trabajo futuro   |    |
| V. Conclusiones   | 66 |

## Parte I

# Introducción

## 1. Propósito

La tesis de maestría se llevó a cabo en el Departamento de Análisis de Datos de Ericsson. Su objetivo a largo plazo es "analizar información pública y personal, y usar algoritmos para tomar decisiones por el usuario". La motivación para hacer esto provino de asistentes personales en dispositivos móviles como Google Now para Android y Siri para iOS. En particular, están interesados en los sistemas de recomendación, especialmente en los sistemas de recomendación grupales. Mi objetivo era ejemplificar esto mediante el desarrollo de un sistema de recomendación para almuerzos en restaurantes en una aplicación de Android. La interfaz de usuario, la aplicación de Android, no fue desarrollada por mí, fue desarrollada por estudiantes de Ericsson Labs en Beijing, China. Mi tarea fue desarrollar el backend del sistema con el servidor, el analizador y el sistema de recomendación. Los requisitos de especificación completa de los empleadores para el diseño de la parte delantera y trasera de este sistema se explican en la implementación de la sección 5. Los problemas planteados a partir de estos requisitos fueron los siguientes:

**Información del usuario** No se pueden solicitar identificadores de usuario explícitos a los usuarios, como nombres de usuario o números de teléfono. Esto significaba que la recomendación de grupo para un grupo completo tenía que basarse en la entrada de una persona a la aplicación. Para resolver esto, el empleador sugirió el supuesto de navegación grupal como se describe en la sección 6.2. La idea era que el sistema pudiera simular una selección de grupo como un todo sin conocer las preferencias de una persona individual en el grupo.

**Preferencias implícitas** Todo el sistema estaba destinado a requerir la menor cantidad de información posible por parte del usuario. Al igual que la información del usuario, no se pueden recopilar calificaciones explícitas de los usuarios para los elementos del sistema.

**Recomendación individual** La primera tarea fue proporcionar una recomendación individual para un usuario basado en preferencias anteriores.

**Recomendación de grupo** Después de que se logró una recomendación individual, una recomendación de grupo se iba a implementar el sistema ommender.

Para abordar estos problemas se hizo lo siguiente en esta tesis de maestría:

1. Una discusión sobre en qué contexto se han desarrollado los sistemas de recomendación y cómo logran la felicidad del usuario.
2. Un estudio de literatura sobre sistemas de recomendación individual, sistemas de recomendación grupal y trabajos previos en estos campos.



3. Un estudio de la literatura sobre la agregación de resultados y la obtención de selecciones implícitas de comportamiento del usuario.
4. Desarrollo de un servidor de manejo de datos de usuarios y datos públicos.
5. Dirigir y especificar el desarrollo de aplicaciones de la aplicación front-end de Android.  
plegamiento.
6. Desarrollar un método novedoso para la recopilación de calificaciones implícitas cuando un usuario navega por una lista de descripciones de elementos.
7. Desarrollar un algoritmo para predecir calificaciones individuales para los elementos.
8. Desarrollar un algoritmo para predecir las calificaciones grupales de los elementos.

## 2 Discusión sobre decisiones y elección

¿Sobre qué bases tomamos decisiones y elecciones? Todo depende de qué se trate la elección y qué impacto tendrá la decisión en nuestras vidas. Las tácticas que las personas utilicen para abordar este problema serán muy diferentes dependiendo de si la elección se trata de elegir una universidad, a dónde ir de vacaciones, qué restaurante visitar o qué música escuchar.

Tal vez hagamos una investigación exhaustiva sobre el tema, busquemos información y consideremos todas las opciones posibles disponibles. También podemos pedirle orientación y sugerencias a un amigo en quien confiamos o incluso a un extraño. Algunas personas pueden tener en cuenta las opiniones de personas más experimentadas, ya sea preguntando o leyendo reseñas y artículos escritos por ellos.

A veces simplemente elegimos al azar o al tacto.

Este proceso de toma de decisiones puede ser engorroso y exigir trabajo. Para elegir un buen restaurante para una ocasión especial, el cliente puede buscar todos los restaurantes del vecindario, leer los menús, sopesar sus propias preferencias y las preferencias de los comensales. Para elegir un buen plan de seguro, se puede contratar a un profesional que haga este trabajo por usted y seleccione un plan adecuado adaptado a sus necesidades.

Últimamente, la transición a la era digital ha llevado a un gran aumento en la cantidad de productos que se pueden consumir instantáneamente a diario. El número de opciones se ha disparado. Por ejemplo, tome la actividad de escuchar música. Hace algunas décadas, el consumidor tenía que ir a una tienda de música para comprar un disco. En algunas tiendas de música, se permitía al cliente intentar escuchar canciones, pero sacar un disco LP o CD de la estantería y ponerlo en un reproductor lleva algo de tiempo, y el dueño de la tienda ciertamente no dejaría que el cliente siguiera escuchando canciones. indefinidamente. Hoy el usuario accede fácilmente a Internet a través de computadoras o dispositivos móviles y escucha música con servicios de streaming como Spotify y Grooveshark. Estos sitios tienen miles de títulos de canciones disponibles, muchos más que las antiguas tiendas de música. El usuario puede consumir cientos de títulos en poco tiempo;

comience a escuchar una canción, salte hacia adelante, decida si continúa escuchando la canción o pasa a la siguiente. La decisión de escuchar una canción en particular no requerirá de análisis o consultas ya que el impacto de tomar una mala decisión y elegir una canción que no se ajuste a las preferencias del usuario solo hará que el usuario salte a la siguiente canción. Sin embargo, escuchar una canción lleva algo de tiempo y es posible que el usuario tenga que escuchar varias canciones que no son de su agrado antes de que aparezca una buena canción.

El mismo fenómeno se aplica a las películas. A principios del siglo XX, solo una o un par de películas se proyectaban en un cine una noche en particular. Del mismo modo, en los primeros días de la televisión, solo había un puñado de estaciones que transmitían, pero desde entonces ha aumentado constantemente. Más tarde, la tecnología de video y las tiendas de alquiler de videos hicieron muchas opciones, hasta la era de Internet, donde sitios web como Netflix y Hulu han brindado a sus clientes transmisión de películas a pedido. Al igual que con la mayoría de los servicios de transmisión de música, las membresías son en forma de suscripciones, lo que significa que los usuarios no tienen que preocuparse por los gastos en el camino. A veces, las películas o los programas de televisión se pagan con anuncios, lo que significa que los usuarios primero tienen que mirar un clip de anuncio antes de poder ver el contenido. Sin embargo, la diferencia entre la música y las películas es que la duración promedio de una canción es de solo un par de minutos, mientras que para una película o un programa de televisión, el lapso de tiempo suele ser de horas. El usuario puede tardar un tiempo en determinar si la película que está viendo actualmente es de su agrado o no. Hacer que el usuario mire a través de una serie de películas, de manera similar a como se describió anteriormente con la música, llevará tiempo y tal vez requiera que el usuario mire a través de varios anuncios. Se podría argumentar que el observador de películas podría ver los tráileres de las películas, pero la aprobación de un tráiler no coincide necesariamente con la aprobación de la película real y también lleva tiempo mirar los tráileres.

Otra revolución innovadora que llegó con la llegada de Internet y la World Wide Web fueron las compras en línea y el comercio electrónico para las empresas minoristas. Un cliente puede navegar a través de una gran cantidad de productos de los sitios web de los vendedores o usar motores de búsqueda de compras para encontrar ciertos productos o marcas disponibles. Cuando se toma la decisión de comprar un determinado artículo, el cliente paga el producto a través de un banco de Internet, después de lo cual el producto se envía a la dirección del cliente. Una diferencia entre comprar productos en línea y transmitir música o películas es que el tiempo de consumo es de un par de días para la compra, mientras que para la transmisión el consumo comienza inmediatamente después de la selección. La clave para comprar en línea es que el cliente basa principalmente la decisión de qué artículo comprar leyendo las descripciones y mirando las imágenes de los productos. Se asemeja a la actividad de ver avances de películas como se describe anteriormente; una aprobación de una imagen o una descripción de un producto puede no parecerse a la aprobación del producto real después de un tiempo de uso, aunque comúnmente están correlacionadas. Sin embargo, un usuario a veces puede probar una cierta cantidad de productos con una compra abierta y devolverlo con la devolución del dinero si el cliente no está satisfecho. Este procedimiento es bastante agotador.

y requiere mucho tiempo, y coincide con la actividad de ver películas o escuchar canciones como se describió anteriormente. Además, el tiempo de consumo para usar los productos puede diferir mucho según el artículo que se compre. Después de un sorbo de vino, uno sabrá inmediatamente si es bueno o malo según los gustos, mientras que para un automóvil el ciclo de vida es de años.

Internet hizo eficiente la distribución de libros electrónicos y permitió que se introdujeran en un mercado masivo. Un libro electrónico es un libro en formato digital legible en dispositivos como computadoras, tabletas, lectores electrónicos o dispositivos móviles. Al principio, se publicaron en medios de almacenamiento en disco como CD o disquetes. Para entonces, las únicas ventajas de los libros electrónicos frente a los libros tradicionales eran que eran fáciles de llevar y más baratos de producir, pero las desventajas aún son varias. Actualmente, los libros y las revistas se publican en Internet y se descargan directamente a lectores especiales de libros electrónicos.

Los minoristas de libros en línea como Amazon ofrecen libros electrónicos y libros en el formato tradicional. La diferencia con los servicios de transmisión es que el contenido digital de los libros generalmente se vende para cada uso. El tiempo de consumo de un libro es de varias horas, más que el de una película media. Por lo tanto, los clientes leen reseñas para tener una idea del contenido del libro. El vendedor de libros electrónicos también puede proporcionar al consumidor una muestra del libro, un par de páginas que se pueden leer de forma gratuita. Pero lo que realmente ha cambiado es el tiempo hasta el consumo, el usuario puede descargar el libro en un instante y empezar a leer, no hay necesidad de ir a la librería o biblioteca y empezar a buscar en los estantes.

No importa si se trata de música, películas, libros electrónicos u otros artículos, el denominador común es, como se dijo antes, que la cantidad de opciones es enorme con Internet. También los artículos que se asemejan al consumo; trailers, muestras, reseñas o descripciones son muy abundantes y de fácil acceso. El tiempo hasta el consumo tanto del contenido en sí como de las reseñas y muestras también ha disminuido considerablemente. Antes uno tenía que tomar un periódico o una revista que hiciera una reseña de un producto, o un catálogo que contenía el artículo, mirar la tabla de contenido y desplazarse por las páginas, donde hoy simplemente ingresamos un término de búsqueda en el motor de búsqueda. . Con tantas opciones, es imposible que el usuario las considere todas, y uno puede sentirse abrumado fácilmente.

Dado que el tema de esta tesis de maestría es sobre recomendaciones de restaurantes, es adecuado mencionar la actividad de elegir un buen restaurante cuando sale a cenar. La World Wide Web realmente no ha traído un gran cambio a esta búsqueda en particular. Lo que se consume en un restaurante no es solo la comida, sino también el servicio y el entorno del comedor, lo que se suma a la experiencia del usuario y la aprobación de la comida real. El tiempo de consumo de una comida puede ser desde un par de minutos hasta horas, y el tiempo desde la decisión hasta el consumo es en el mismo orden. En realidad, se están tomando dos decisiones; a qué restaurante ir y qué plato en particular pedir. La primera de estas decisiones suele tomarla un grupo de personas, ya que muchas veces solemos salir a comer juntos.

La segunda es más una elección individual. Después de seleccionar un restaurante, la fiesta

deben transportarse allí. Cuando el grupo llega al restaurante, pueden tener una idea de si les gusta el restaurante o no, pueden mirar el menú y decidir si quedarse o buscar otro. Como antes, este procedimiento requiere mucho tiempo y aumenta el tiempo hasta que se haya hecho una selección definitiva de un restaurante. La selección de un plato se realiza mirando las descripciones o imágenes de los elementos reales que están disponibles para ordenar en el menú. Se asemejan y describen los artículos como avances o reseñas, y el tiempo para considerar el menú es mucho menor que para probar todos los platos. Uno podría imaginar otras formas para que los comensales del restaurante se formaran una opinión sobre los platos, por ejemplo, podrían probar muestras de los platos, pero eso sería un poco engorroso y posiblemente costoso para el dueño del restaurante.

### 3 Categorización de artículos

Dados estos ejemplos, y con el fin de categorizar la selección y el consumo de diferentes artículos, sugiero estos ocho factores que son esenciales para comprender la naturaleza de la toma de decisiones. A pesar de un extenso estudio de literatura, no he podido encontrar nada equivalente en la literatura. Que yo sepa, esta clasificación es única.

1. Coste del consumo
2. Tiempo hasta el consumo
3. Momento de consumo
4. Número de opciones
5. Impacto de la decisión
6. Afluencia de nuevos elementos y singularidad de los elementos.
7. Personas involucradas
8. Tipo de consumo

El costo de consumo es cuánto esfuerzo paga el usuario por usar el producto. Por lo general, es dinero, ya sea un pago único o una suscripción, pero también puede ser una caminata de un kilómetro por la ciudad hasta un buen restaurante, con el estómago vacío.

El tiempo hasta el consumo es el tiempo desde que se ha hecho la selección de un artículo hasta que es consumido por un usuario. Esta caminata de un kilómetro mencionada anteriormente se asemeja a un aumento en el costo del consumo y el tiempo hasta el consumo. Para los servicios de transmisión y

El tiempo de navegación de video en línea hasta que el consumo es cero, mientras que para las compras en línea un par de días.

Como se señaló anteriormente, diferentes productos toman diferentes tiempos para consumir. Este hecho no ha cambiado después de la revolución de Internet. El tiempo que se tarda en escuchar una canción es del orden de minutos, una película varios minutos hasta horas. El tiempo que se tarda en terminar una comida en un restaurante es el mismo que en las películas. Leer libros puede llevar horas y días, dependiendo del lector. Una cosa clave aquí es, como veremos más adelante, el tiempo que tarda un usuario en formarse una opinión sobre el artículo que está consumiendo, si le gusta o no.

El número de opciones es simplemente el número de elementos entre los que puede elegir un consumidor. Tenga en cuenta que esto puede ser decenas de miles en el caso de la música, pero solo un puñado si se trata de platos en el menú de un restaurante. La cantidad de artículos disponibles también cambia con el tiempo, en el caso de música o películas, casi todos los títulos anteriores siguen siendo accesibles, pero para productos en una tienda en línea o platos de restaurante, esto será muy diferente en diferentes ocasiones.

A veces, el impacto de la decisión puede ser muy significativo para el individuo que elige entre las opciones. Imagine a un estudiante de secundaria eligiendo entre universidades o un director ejecutivo invitando a cenar a clientes potenciales. Este tipo de decisiones son muy importantes y el que elige seguramente se esforzará más en explorar las diferentes posibilidades.

Algunos restaurantes pueden servir platos nuevos todos los días, que nunca antes habían tenido en su menú. En ese caso, cada plato es único y solo se puede consumir en una ocasión. Para los libros, por otro lado, los proveedores obtienen nuevos libros escritos de vez en cuando, pero cada libro puede ser leído por un grupo más grande de personas. Eso afectará la forma en que podemos recopilar calificaciones sobre los elementos, como veremos más adelante. Todas estas consideraciones están cubiertas por la afluencia de nuevos artículos y la singularidad de los artículos.

Las personas involucradas se asemejan a la cantidad de personas que están tomando la decisión en conjunto, su relación respectiva y si hay otras personas afectadas por la decisión. En el ejemplo del restaurante mencionado anteriormente, este factor es claramente de gran importancia. Pero también podría ser un grupo que decida ver una película juntos, o padres afectados por la selección de collage de sus hijos.

La apreciación de una película después de verla es muy diferente de la opinión después de ver el tráiler o leer una reseña. Es por eso que uno tiene que dejar claro si es la descripción que se parece al artículo real que se está consumiendo o el artículo mismo. El tiempo de consumo y el tiempo hasta el consumo es muy diferente dependiendo de si estamos leyendo el menú describiendo el plato real o comiéndolo. Estas dos medidas son del orden de segundos para abrir el menú de un restaurante y leer sobre un plato, pero

minutos para el plato real que se entrega y horas para el consumo. Por lo tanto, estas consideraciones están cubiertas en el tipo Consumo.

## 4 Antecedentes

### 4.1 Sistemas de recomendación

Un sistema de recomendación es un sistema que da sugerencias a un número de usuarios sobre qué artículo les gustaría dado lo que les ha gustado en el pasado, qué preferencias han especificado manualmente en sus perfiles u otros datos que están disponibles sobre el usuario [1, 2, 3, 4, 5]. El objetivo de un sistema de recomendación es ahorrar tiempo y aumentar la satisfacción del usuario. Es especialmente útil en el caso de tener que navegar entre miles de títulos. Esta tarea de recomendación es de naturaleza inteligente, pudiendo ser realizada previamente por un humano que tenga en cuenta la personalidad y las preferencias de la persona que elige.

En caso de que el costo del consumo y el impacto de la decisión sean altos, algunos quizás contratarían personas para considerar todas las opciones y brindar una recomendación.

De lo contrario, el usuario podría preguntarle a un amigo que esté bien familiarizado con todos los elementos y la persona que elige. Por el contrario, con Internet los proveedores están ofreciendo al consumidor una elección mucho más amplia de modo que es casi imposible que un ser humano esté familiarizado con todos los artículos.

Dado que este problema es algorítmico hasta cierto punto, requiere que se recolecten y analicen datos, sería deseable tener un programa que haga la sugerencia. También sería adecuado usarlo cuando el usuario busca artículos en Internet, es decir, recomendar una canción, un movimiento, un producto en una tienda en línea o un libro electrónico, ya que la abundancia de los artículos es tan alta y el consumidor puede perderse fácilmente. Los datos de las preferencias del usuario pueden recopilarse explícitamente, es decir, el usuario proporciona una calificación justo después del consumo, o implícitamente, donde se analiza el comportamiento de navegación y las calificaciones se estiman automáticamente. [3, 6]

Independientemente de si la actividad es ver películas, escuchar música o comprar productos en línea, una forma simple de obtener una recomendación para escuchar una canción en particular es elegir de las listas principales. La posición de los elementos en la lista superior se puede basar en las ventas o las calificaciones promedio recopiladas de las encuestas. Obviamente, este tipo de recomendación también podría hacerse antes de la era de Internet. De hecho, esta es una forma simple de sistema de recomendación, pero el principal problema con este enfoque es que no tiene en cuenta el hecho de que los gustos de todos son diferentes y puede producir recomendaciones inútiles. Sin embargo, teniendo en cuenta la popularidad de un artículo, se puede impulsar en gran medida una recomendación existente, [7] un hecho que se utiliza en este sistema de recomendación.

Ahora es apropiado introducir algunas notaciones, tenemos un conjunto de  $m$  usuarios,  $U = \{u_1, u_2, \dots, u_m\}$  y un conjunto de  $n$  elementos que los usuarios pueden consumir  $I = \{i_1, i_2, \dots, i_n\}$ . Para

estos elementos se ha proporcionado un conjunto de calificaciones de los usuarios,  $R = \{r_{ij}\}$ . Aquí tenemos que  $1 \leq i \leq n$  y  $1 \leq j \leq m$ . Estas calificaciones se pueden organizar en una matriz  $T \times m$ , pero algunos componentes estarán vacíos ya que no es seguro que todos los usuarios hayan calificado todos los elementos.

$$T = \begin{matrix} & u_1, \dots, u_m \\ \begin{matrix} i \\ 1 \\ \vdots \\ n \end{matrix} & \begin{matrix} r_{i1} & \dots & r_{im} \\ \ddots & & \ddots \end{matrix} \end{matrix}$$

Generalmente, el usuario proporciona las calificaciones en una escala discreta,  $r_{ij} \in \{1, 2, 3, 4, 5\}$ , y todas estas calificaciones se almacenan en una base de datos en la memoria de la computadora para su posterior análisis. La tarea de un sistema de recomendación es ahora estimar la calificación de un producto  $i$  para un usuario  $j$  donde  $r_{ji} \notin R$ , es decir,  $j$  no ha calificado el artículo antes. Para simplificar los cálculos, denotamos el conjunto de ítems que el usuario  $a$  ya calificó como  $Q(a)$ , donde  $Q(a) \subseteq I$ .

Desde que surgieron los primeros sistemas de recomendación a principios de los 90, ha habido dos enfoques principales para calcular esta estimación, estos son sistemas de recomendación colaborativos y basados en contenido [2, 4, 8, 9]. Ahora hablaré de cada uno de ellos y describiré brevemente cómo funcionan.

#### 4.1.1 Colaborativo

Un sistema de recomendación colaborativo solo tiene en cuenta las calificaciones anteriores del usuario para los elementos, no intenta analizar los elementos reales en sí. El objetivo es encontrar elementos que hayan gustado a usuarios similares en el pasado. [4, 8, 9] Si hay bastantes personas a las que les gusta beber té y comer bollos, y parece que a usted le gusta el té, entonces un sistema de recomendación colaborativo le recomendará que compre té. Encuentra grupos de usuarios que tienen gustos similares. El filtrado colaborativo a su vez se puede dividir en dos subclases, basado en memoria o basado en modelo [3, 2].

**Filtrado colaborativo basado en memoria** En el filtrado colaborativo basado en memoria, la base de datos de calificaciones se mantiene en la memoria y se usa directamente para generar recomendaciones durante el tiempo de ejecución, no hay preprocesamiento. Dos ejemplos de recomendación basada en la memoria son la recomendación del vecino más cercano basada en el usuario y la recomendación del vecino más cercano basada en elementos [3, 2].

Recomendación de vecino más cercano basada en el usuario Aquí el objetivo es encontrar usuarios similares, que previamente han clasificado los elementos similares entre sí. [10, 11, 12, 9] La tarea es estimar una calificación  $\tilde{r}_{ia}$  para un artículo  $i$  que un usuario  $a$  no ha calificado antes y, por lo tanto, es de esperar que no lo haya consumido previamente. Comenzamos calculando la calificación promedio que un usuario  $u$  le ha dado a los productos calificados  $Q(u)$ .

$$\bar{r}_u = \frac{\sum_{i \in Q(u)} r_{iu}}{|Q(u)|} \quad (1)$$

Ahora trataremos de calcular la similitud,  $\text{sim}(a, b)$ , entre un par de usuarios  $a$  y  $b$ . Hay varias formas de hacer esto, una es el coeficiente de correlación de Pearson que se muestra en la ecuación 2. [2, 4, 3, 9] Primero tenemos que encontrar el conjunto de ítems que ha sido calificado tanto por  $a$  como por  $b$ , que es igual a la intersección entre los conjuntos de elementos calificados de los dos usuarios,  $Q(a)$  y  $Q(b)$ .

$$\text{sim}(a, b) = \frac{\sum_{i \in Q(a) \cap Q(b)} (r_{ia} - \bar{r}_a)(r_{ib} - \bar{r}_b)}{\sqrt{\sum_{i \in Q(a) \cap Q(b)} (r_{ia} - \bar{r}_a)^2 + \sum_{i \in Q(a) \cap Q(b)} (r_{ib} - \bar{r}_b)^2}} \quad (2)$$

El coeficiente de correlación de Pearson tomará valores de  $-1$  (usuarios muy diferentes) a  $1$  (usuarios muy similares). Como se ve en la ecuación 2, tenemos en cuenta el hecho de que el comportamiento de calificación del usuario es diferente. Algunas personas pesimistas casi nunca le dan una buena calificación a un artículo, mientras que otros le dan un cinco a cada artículo que encuentran. Es la calificación relativa de un producto que nos interesa.

A continuación procederemos a realizar la estimación. El concepto clave es seleccionar un umbral para el coeficiente de correlación y, por lo tanto, seleccionar un subconjunto de usuarios que sean más similares a  $a$ . Estos también se denominan vecinos más cercanos. Los seleccionamos y los ponemos en un subconjunto  $N$ , donde  $N \subseteq U$ . Luego se puede calcular la estimación:

$$\tilde{r}_{ia} = \bar{r}_a + \frac{\sum_{b \in N} \text{sim}(a, b)(\bar{r}_b - \bar{r}_a)}{\sum_{b \in N} \text{sim}(a, b)} \quad (3)$$

La ecuación 3 considera a cada usuario vecino y calcula su aprobación relativa del ítem  $i$ . Esta aprobación se pondera por la proximidad de cada vecino al usuario para el que estamos haciendo la estimación y, finalmente, la expresión se normaliza.



Recomendación del vecino más cercano basada en artículos El problema con la recomendación basada en el usuario es que tenemos que considerar una gran cantidad de usuarios del vecino más cercano, que en los grandes sitios de comercio electrónico pueden ascender a millones. Por lo tanto, el sistema es muy ineficiente y no se puede realizar en tiempo de ejecución en una aplicación a gran escala. En cambio, el sistema de recomendación basado en elementos analiza los elementos anteriores que los usuarios calificaron y calcula la similitud entre pares de elementos. [11, 2, 3, 4, 9] Considere los ítems  $c$  y  $d$ , donde  $c, d \in I$ . La similitud entre estos ítems se puede calcular como el producto escalar entre dos vectores fila en la matriz de calificaciones  $T$ . Dados estos dos los vectores se denotan  $rc_{1:m}$  y  $rd_{1:m}$ , podemos escribir la similitud entre los elementos  $c$  y  $d$  de la siguiente manera:

$$\text{sim}(c, d) = \frac{rc_{1:m} \cdot rd_{1:m}}{|rc_{1:m}| \cdot |rd_{1:m}|} \quad (4)$$

$$T = \begin{matrix} & \begin{matrix} u_1 & \dots & u_m \\ \vdots & & \vdots \end{matrix} & \\ \begin{matrix} i_1 \\ \vdots \\ i_n \end{matrix} & \begin{matrix} r_{c1} & \dots & r_{cm} \\ \vdots & & \vdots \\ r_{d1} & \dots & r_{dm} \end{matrix} & \end{matrix}$$

Tenga en cuenta que la similitud entre los elementos no tiene nada que ver con lo que contienen los elementos en sí, o lo que los caracteriza. Es solo si han sido calificados de manera similar por los usuarios.

Lo que calculamos en la ecuación 4 es el producto escalar normalizado de dos vectores, y que a su vez es igual al coseno del ángulo entre los vectores. Por lo tanto, esta métrica se etiqueta como la similitud del coseno. [2, 3, 4, 6] Esta métrica se utiliza en el aprendizaje automático, los sistemas de recuperación de información y los sistemas de recomendación, y ha demostrado que produce resultados precisos. Dado que solo hay calificaciones positivas en nuestro ejemplo, esta medida de similitud solo toma valores entre 0 y 1.

Un problema con este enfoque es que no tiene en cuenta que diferentes usuarios tienen diferentes comportamientos de calificación. Como se describió anteriormente, esto se tiene en cuenta al calcular el coeficiente de la ecuación de correlación de Pearson en la ecuación 2. Por lo tanto, definimos

el coseno ajustado mide como:

$$\text{sim}(c, d) = \frac{\sum_{u \in U} \frac{r_{cu} r_{du}}{\sqrt{r_{cu}^2 + r_{du}^2}}}{\sqrt{\sum_{u \in U} r_{cu}^2} \sqrt{\sum_{u \in U} r_{du}^2}} \quad (5)$$

Las sumas de la ecuación 5 se toman sobre el conjunto de usuarios que cumplen la relación  $u \in U \mid c \in Q(u) \text{ y } d \in Q(u)$ . El usuario debe haber calificado los elementos  $c$  y  $d$  para contribuir a la similitud. En realidad, esto también es cierto para la similitud del coseno tradicional en la ecuación 4, pero en ese caso simplemente omitimos los términos en el producto escalar para los que no existe una calificación. Al igual que con el coeficiente de correlación de Pearson en la ecuación 2, esta similitud toma valores entre  $-1$  y  $1$ .

Ahora intentamos predecir una calificación para un elemento  $c$  no calificado por un usuario  $u$  usando similitudes entre  $i$  y los elementos  $u$  calificados previamente como  $Q(u)$ . Cuanto más similar sea uno de estos artículos al artículo real estimado  $i$ , más influirá la calificación de  $u$  en la calificación final.

$$\tilde{r}_{cu} = \frac{\sum_{i \in Q(u)} \text{sim}(c, i) \cdot r_{iu}}{\sum_{i \in Q(u)} \text{sim}(c, i)} \quad (6)$$

Aunque este enfoque basado en elementos es más adecuado para el preprocesamiento fuera de línea que la recomendación basada en el usuario [3], aún requiere mucha memoria y potencia informática para realizar los cálculos, especialmente cuando hay millones de usuarios y elementos.

**Filtrado colaborativo basado en modelos** El objetivo del filtrado colaborativo basado en modelos es preprocesar los datos en un modelo que se utiliza para obtener las recomendaciones durante el tiempo de ejecución. [2, 3, 4] Si en el ejemplo anterior precalculamos una matriz (simétrica) de similitudes de elementos por pares para acceso instantáneo, el sistema pasaría a estar basado en modelos. Por supuesto, hay varias formas de obtener un modelo, ahora describiré brevemente dos de ellas; recomendación colaborativa bayesiana ingenua y recomendación colaborativa basada en SVD.

**Naive Bayes** En nuestros ejemplos, las calificaciones se califican en una escala discreta de 5 puntos. Por lo tanto, podemos tratar las calificaciones como clases y utilizar métodos estadísticos para clasificar la calificación de un elemento no visto. [2, 3]

El clasificador Naive Bayes se basa en el teorema de Bayes, que establece la relación entre probabilidades de dos eventos,  $A$  y  $B$  y sus respectivas probabilidades condicionales.

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \quad (7)$$

Dado que falta una calificación para un ítem  $i$  por parte de un usuario  $u$ , se utiliza el teorema de Bayes para aproximar la probabilidad de que la calificación sea igual a cada una de las puntuaciones de la escala. Luego, elegimos el puntaje con la probabilidad más alta y se lo asignamos a  $\tilde{r}_{iu}$ .

$$\arg \max_x P(r_{iu} = x | r_{qu} : q \in Q(u)) = \tilde{r}_{iu} = \frac{P(r_{qu} : q \in Q(u) | r_{iu} = x) \cdot P(r_{iu} = x)}{P(r_{qu} : q \in Q(u))} \quad (8)$$

Ahora hay dos cosas que podemos usar, la primera es que el denominador en la ecuación 8 es independiente de  $x$ . Esto significa que no necesitamos lidiar con eso al calcular las probabilidades, ya que solo estamos buscando la probabilidad más alta. El segundo es el supuesto de Bayes ingenuo: todas las probabilidades son condicionalmente independientes. Esto significa que damos por sentado que  $P(r_{qu} : q \in Q(u) | r_{iu} = x) = \prod_{q \in Q(u)} P(r_{qu} | r_{iu} = x)$ , que en realidad es una suposición ingenua, pero Naive Bayes ha demostrado dar un rendimiento decente en el pasado. [2, 5]. Con estos dos supuestos concluimos que

$$\tilde{r}_{iu} = \arg \max_x \prod_{q \in Q(u)} P(r_{qu} | r_{iu} = x) \cdot P(r_{iu} = x) \quad (9)$$

La siguiente tarea es averiguar cómo interpretar estas probabilidades a partir de la matriz de calificaciones  $T$ . Es importante tener en cuenta que el ítem  $i$  no está calificado por  $u$ , por lo que tenemos que usar las calificaciones de otros usuarios  $u'$  cuando aproximamos la probabilidad  $P(r_{iu} = x)$  en la ecuación 9. Simplemente contamos el número de calificaciones  $x$  otorgadas al elemento  $i$  dividido por el número total de calificaciones otorgadas a  $i$ .

$$P(r_{iu} = x) = \frac{\sum_{u' \in U : i \in Q(u')} \mathbb{1}_{r_{iu'} = x}}{\sum_{u' \in U : i \in Q(u')} 1} \quad (10)$$

De manera similar, cuando la expresión  $P(r_{qu} | r_{iu} = x)$  se calcula usando solo las filas en el  $T$ -matriz donde un usuario  $u'$  otorgó el puntaje  $x$  al elemento  $i$ . Para cada elemento  $q$  y cada usuario  $u'$  contamos el número de calificaciones  $x$ , dividido por el número total de calificaciones a  $q$  por  $u'$  y asignamos el valor a  $P(r_{qu} | r_{iu} = x)$ .

$$Y = P(r|u) = Y \quad \frac{P(u|Y)}{P(u)} = \frac{P(u|Y)}{P(u)} \quad (11)$$

A estas alturas, todas las probabilidades están explícitamente definidas y podemos calcular la más probable calificación con la ecuación 8.

Recomendación basada en SVD Hay varios tipos de técnicas de reducción de dimensionalidad que podemos emplear para reducir la cantidad de datos que necesitamos manejar en la matriz de calificaciones para capturar mejor el modelo. [13] Como se indicó anteriormente, si los elementos y los usuarios suman millones, en el peor de los casos podemos tener hasta un billón de entradas. Por supuesto, esto no es cierto en la realidad, ya que cada usuario solo califica un pequeño subconjunto de los elementos.

En la factorización SVD, la matriz de calificaciones  $T$  se puede descomponer en el producto de tres matrices  $U$ ,  $\Sigma$  y  $V$  de la siguiente manera. [3]

$$T = U \Sigma V^T \quad (12)$$

Donde  $T$  es una matriz  $n \times m$ ,  $U$  es una matriz ortogonal  $n \times n$ ,  $\Sigma$  es una diagonal rectangular es una matriz  $n \times m$  y  $V$  es una matriz ortogonal  $m \times m$ .

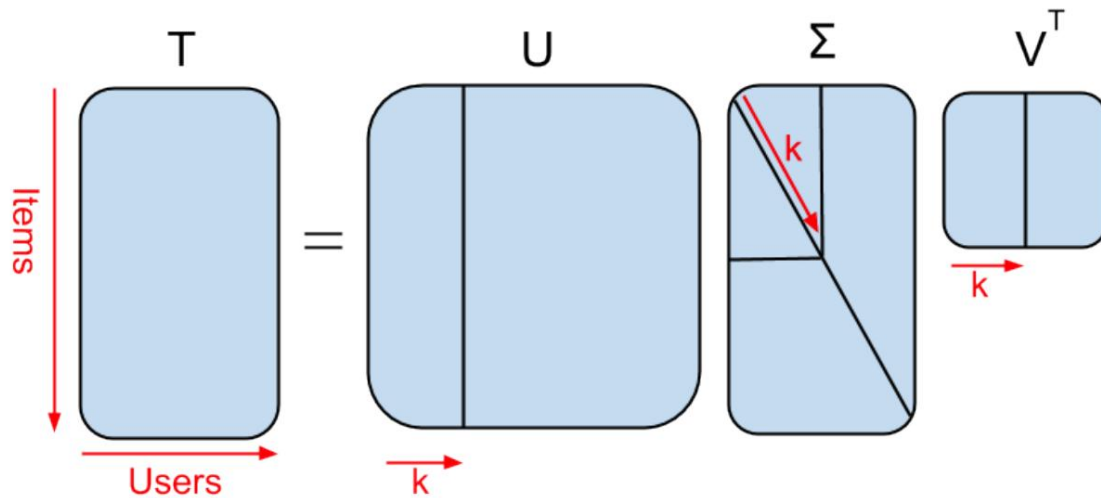


Figura 1: Forma de las matrices en SVD-descomposición y truncamiento

La matriz  $\tilde{Y}$  es una matriz diagonal, y solo tiene valores en su diagonal, mientras que el resto es cero. Estos valores se denominan valores singulares y resulta que están ordenados de forma decreciente, con el mayor en la esquina superior izquierda. Al elegir solo los valores singulares top-k y establecer el resto en cero, y multiplicando  $U$ ,  $\tilde{Y}$  y  $V^T$  de nuevo a una nueva matriz  $T$ , hemos creado una aproximación de rango k de la matriz de calificaciones original  $T$ . Este procedimiento se llama truncamiento y, como se muestra en la figura 1, también podríamos descartar todas menos las primeras k columnas de  $U$  y  $V$  ya que las otras son iguales a cero. Denotamos estos nuevos que solo matrices con columnas descartadas como  $U_k$ ,  $\tilde{Y}_k$  y  $V_k^T$ .

También parece que las filas de  $U$  corresponden a vectores de características de artículos, y las filas de  $V$  corresponden a vectores de características de usuarios. Estos se pueden colocar en un espacio de características de elemento de usuario común, donde se pueden observar usuarios y elementos similares. En realidad, solo necesitamos considerar solo

extraer las k primeras componentes de los vectores fila de  $U$  y  $V$ ,  $V_k^T$  y Reino Unido.

Ahora pasamos al objetivo real de nuestro sistema de recomendación; recomendar un artículo a un usuario. Hay muchas maneras inteligentes de hacer esto. Previamente, cuando realizamos la factorización de la matriz de calificaciones  $T$ , simplemente establecimos todas las calificaciones faltantes en cero. Cuando queremos obtener una recomendación para el usuario  $u$ , podemos mirar todos los vectores de características de los elementos en las cercanías de  $u$  en el espacio de características del elemento del usuario. Una medida directa para calcular esto es la similitud del coseno. Luego, podemos verificar el elemento más cercano para el que no existe una calificación y recomendarlo al usuario.

Si obtenemos un nuevo vector de calificación de un usuario,  $r_{1:n,u}$ , y queremos colocarlo en el espacio de características del elemento del usuario, podemos proyectarlo usando esta fórmula.

$$\hat{r}_k = r_{1:n,u} \cdot \text{Reino Unido} \cdot \tilde{Y}_k^T \quad (13)$$

Donde  $\tilde{Y}_k$  es la matriz que resulta al mantener solo los k primeros valores singulares.

#### 4.1.2 Basado en contenido

Una desventaja de un sistema de recomendación colaborativo es que nunca sabe nada sobre el contenido de los elementos que recomienda. Un sistema de recomendación basado en contenido recomienda un elemento a un usuario en función de una descripción o el contenido real de un elemento [2, 3, 4, 5, 8, 9]. En el caso de películas, las descripciones pueden ser datos textuales como género, director, actores, tiempo de reproducción y edad recomendada. Para las canciones, podría ser artista, álbum, estilo musical y fecha de lanzamiento. Obviamente, los artículos de noticias y los libros en línea podrían tener el mismo tipo de descripciones, pero dado que el contenido de texto de estos elementos ya está en el sistema, todo el cuerpo del texto podría servir como descriptor del contenido. Los archivos de música y películas también se almacenan, por lo que los datos visuales y de sonido de estos podrían parecerse a ellos en

un sistema de recomendación basado en contenido, pero el problema con esto es que el sonido y el video ocupan mucho espacio, varios megabytes en el caso de la música y varios cientos de megabytes hasta gigabytes para el video. Se requeriría una inmensa cantidad de poder de procesamiento para hacer un sistema de recomendación basado en esto, razón por la cual la gran mayoría de los sistemas de recomendación basados en contenido hasta la fecha se han basado en el análisis textual.

**Frecuencia de términos: frecuencia inversa del documento** Una forma estándar de caracterizar documentos de texto en sistemas de recomendación basados en el contenido es la frecuencia de términos: medida de frecuencia inversa del documento [2, 3, 5]. Se basa en la idea simple de que si una palabra clave aparece con frecuencia en un documento, se asemejará más al contenido del documento. Pero si la palabra clave en cuestión aparece con frecuencia en todos los documentos de la colección de documentos, la importancia relativa de la palabra se reduce.

Introduzcamos algo de terminología, tenemos un conjunto de documentos  $D$ , al que llamamos corpus de documentos, donde  $j \in D$  y hay  $N$  documentos en el corpus,  $|D| = N$ . En estos documentos tenemos un conjunto de palabras,  $I$  donde  $i \in I$ . Una función de frecuencia,  $\text{freq}(i, j)$ ,  $\text{freq}(z, j)$  dará cuenta del número un documento  $j$ . La expresión  $\max_z \text{freq}(z, j)$  la frecuencia máxima de una palabra en el documento  $j$ . La continuación tenemos la función  $n(i)$  que cuenta el número de documentos en los que aparece la palabra clave  $i$ . A partir de esto se definen dos medidas:

$$\text{TF}_{ij} = \frac{\text{frecuencia}(i, j)}{\max_z \text{frecuencia}(z, j)} \quad (14)$$

$$\text{IDF}_i = \frac{1}{n(i)} \quad (15)$$

Donde la ecuación 14 es la frecuencia del término y la ecuación 15 es la frecuencia inversa del documento.[2] El término frecuencia varía de cero a uno, mientras que la frecuencia inversa del documento toma valores entre cero y  $\log(N)$ . El término frecuencia frecuencia de documento inversa, o abreviadamente TF-IDF para una palabra clave  $i$  en un documento  $j$  son estas dos expresiones multiplicadas juntas:

$$\text{TF-IDF}_{ij} = \text{TF}_{ij} \cdot \text{IDF}_i \quad (\text{dieciséis})$$

Ahora hemos calculado qué tan bien una determinada palabra clave caracteriza un documento. Pero nuestro objetivo es comparar documentos entre sí, y eso se puede hacer si cada documento se describe con un vector de características,  $\vec{d}_j$ . Construimos este vector para que cada componente sea una puntuación TF-IDF de un término  $i$ , por lo que hay en total  $|I|$  componentes

$$\vec{d}_j = (\text{TF-IDF}_{1j}, \text{TF-IDF}_{2j}, \dots, \text{TF-IDF}_{|I|j}) \quad (17)$$

Hay algunos preprocesamientos que se pueden realizar en el corpus, lo que mejorará los resultados. Dos de estos son la derivación y la elección de palabras vacías. Stemming es modificar palabras gramaticalmente cambiadas en sus formas de raíz y agrupar sinónimos. Por ejemplo, "papas" y "papas" se agruparán como la misma palabra clave. Elegir palabras vacías es eliminar palabras frecuentes como preposiciones, "y" y "yo" que no comunican nada sobre el contenido.

Para obtener una calificación predicha para un documento  $d$  sin calificar, la similitud del coseno se puede usar para encontrar los  $k$  documentos más similares a  $d$  para los que existe una calificación, y permitirles votar por la clasificación en el documento actual.

Algoritmo de Roccio Sin embargo, predecir la puntuación de un documento no puntuado no es la única forma de recomendar un artículo. El algoritmo de Roccio es un método de retroalimentación de relevancia originado en la investigación de recuperación de información en los años 70. También se ha utilizado en sistemas de recomendación [5]. A un usuario se le muestra una serie de documentos y especifica cuáles de ellos son relevantes y no relevantes. Cada uno de los vectores de características de estos documentos se colocan en los conjuntos  $D_r$  y  $D_{nr}$  respectivamente. Un vector de características que se asemeja al usuario,  $q$ , se deriva de estas entradas.

$$\bar{q} = \gamma q_0 + \tilde{\gamma} \frac{1}{|D_r|} \sum_{d \in D_r} \bar{r}_d - \tilde{\gamma} \frac{1}{|D_{nr}|} \sum_{d \in D_{nr}} \bar{d} \quad (18)$$

Donde  $\bar{q}$  es el vector de características tf-idf del usuario,  $q_0$  es el vector de características tf-idf del usuario anterior,  $\gamma$  es el grado en que disminuye la preferencia por los documentos antiguos,  $\tilde{\gamma}$  es la cantidad de comentarios positivos y  $\tilde{\gamma}$  es la cantidad de comentarios negativos. El algoritmo actualiza el vector de usuario  $q$  de forma recursiva, donde se repite el procedimiento y el antiguo vector de usuario se establece en  $q_0$ . Generalmente  $\gamma + \tilde{\gamma} + \tilde{\gamma} = 1$ . La recomendación real de un documento no leído se realiza calculando la similitud del coseno entre  $q$  y los vectores de características tf-idf del documento no considerado. Se recomienda al usuario el documento que mejor coincida con el interés del usuario en el que se trata. Se utiliza un enfoque similar al algoritmo de Roccio al recomendar platos a los usuarios en este sistema de recomendación.

## 4.2 Otras extensiones y técnicas de aprendizaje automático

La principal desventaja de los sistemas de recomendación colaborativos es que requieren muchas calificaciones existentes para hacer recomendaciones precisas. Esto hace que los sistemas funcionen mal al principio; después de implementarlos, simplemente no hay suficientes usuarios similares en estas primeras etapas. Este problema se denomina problema de arranque en frío [4, 3, 2].

Un problema con el sistema de recomendación basado en contenido es que solo recomendará artículos dentro de la misma categoría. No hay diversificación de los artículos recomendados. Después de un tiempo, esto puede volverse escalonado para el usuario, que tal vez quiera ver algo completamente nuevo.

Hay varias formas de hacer frente a estos inconvenientes, una es combinar los sistemas, y estos se denominan sistemas de recomendación híbridos. [2, 3, 4] La investigación ha demostrado que funcionan mejor que cuando solo se usa uno de los dos enfoques estándar. En cierto modo, la predicción de una calificación para un elemento no visto puede verse como una tarea de aprendizaje automático. En el pasado se han empleado algoritmos tales como máquinas de vectores de soporte, redes neuronales artificiales y árboles de decisión, realizando la regresión de calificación con gran éxito [5, 3].

### 4.3 Sistemas de recomendación de grupos

Muchas actividades, como ver películas o comer, a menudo no se realizan solos. Un sistema de recomendación de grupo tiene como objetivo recomendar un artículo a un grupo de personas. A menudo es difícil medir la eficacia de una recomendación de este tipo, ya que todos los miembros del grupo tendrán opiniones diferentes sobre el elemento recomendado. Se puede incorporar una gran cantidad de psicología y filosofía al problema, cómo determinar la satisfacción del grupo de manera confiable. [14] Si, por ejemplo, es el cumpleaños de uno de los miembros del grupo, es razonable que no se sienta decepcionado con el elemento seleccionado y, por lo tanto, su calificación debe tener un mayor peso. [15]

Hay dos enfoques principales de recomendación de grupos que se han utilizado en el pasado: modelos de preferencia de grupo [16, 17, 7, 15] y agregación de calificación [10, 12, 1, 9, 18, 19, 20, 15]. Esta tesis utiliza ambos métodos en el algoritmo de recomendación. Un modelo de preferencia de grupo utiliza selecciones y calificaciones anteriores de los usuarios, crea un modelo y estima una calificación para cada elemento del grupo. La agregación de calificaciones primero estima una calificación para cada usuario y luego las agrega en una calificación de grupo. La forma en que se construye un modelo de preferencia de grupo es muy específica del problema. Uno podría combinar palabras clave de los intereses de los miembros individuales del grupo y encontrar un elemento que coincida con la mayoría de los sinónimos de estas palabras clave. Esto ya lo han hecho anteriormente MS Pera y Y.-K. Ng [7]. En esta tesis, el algoritmo de aprendizaje automático que captura la dinámica de grupo está construyendo un modelo de preferencia de grupo. Para la agregación de rangos hay varios enfoques, un subconjunto de estos son promedio, menos miseria, desacuerdo promedio y más placer. Todos estos se basan en evaluaciones psicológicas y, a menudo, se agregan en una calificación final. En las fórmulas a continuación,  $G$  indica el conjunto de usuarios que realizan la actividad juntos, es decir, el grupo, y tenemos que  $|G| > 1$ .

Promedio Este es el enfoque utilitario estándar, que maximiza la felicidad en el grupo. Se calcula la calificación promedio para un artículo. [10, 11, 12, 1, 16, 19, 20] Uno



El problema con este enfoque es que si un usuario tiene una gran preferencia por un elemento y a los otros usuarios no les gusta este elemento, aún puede ser favorecido sobre un elemento donde las calificaciones son mucho más uniformes.

$$\text{avr}(G, i) = \frac{\sum_{u \in G} r_{iu}}{|G|} \quad (19)$$

**Mínima miseria** La recomendación grupal de mínima miseria minimiza la infelicidad individual en el grupo. [10, 12, 16, 19, 15, 20] La puntuación de grupo de un elemento  $i$  se establece igual a la calificación estimada más pequeña para el elemento entre los usuarios. Una desventaja de esto es que si un artículo tiene una calificación alta para todos los miembros del grupo excepto uno, y otro artículo tiene una calificación mucho más baja para todos los usuarios excepto para el que no le gusta el último artículo, que tiene una calificación ligeramente más baja, el primero el elemento seguirá estando seleccionado. Por ejemplo, las valoraciones de tres usuarios pueden ser en tal caso [10, 10, 3] para el primer elemento y [4, 4, 2] para el segundo elemento. Obviamente, es mejor elegir el primer elemento, ya que al último usuario parece que no le gustan ambos elementos.

$$\text{lms}(G, i) = \min_{u \in G} r_{iu} \quad (20)$$

**Desacuerdo promedio** Es justo para todos los miembros del grupo si se selecciona un ítem donde la mayoría de los usuarios tienen la misma preferencia por el ítem. Minimizar el desacuerdo sobre los artículos es una forma justa y diplomática de elegir. El desacuerdo promedio por pares, introducido por primera vez por S. Amer-Yahia, SB Roy, A. Chawla, G. Das y C. Yu [10] se puede calcular de la siguiente manera:

$$\text{dis}(G, i) = \frac{2}{|G|(|G| - 1)} \sum_{u, v \in G, u \neq v} |r_{iu} - r_{iv}| \quad (21)$$

**Mayor placer** Esta medida es lo contrario de menor miseria: trata de maximizar la felicidad individual en el grupo. [12, 16] Este método no es realmente una forma justa de seleccionar, pero puede tener algunas ventajas si un usuario del grupo es particularmente importante.

$$\text{mps}(G, i) = \max_{u \in G} r_{iu} \quad (22)$$

#### 4.4 Trabajo previo

Esta tesis de maestría constaba de varias subpartes que debían investigarse antes de seleccionar un enfoque para resolver el problema.

#### 4.4.1 Recomendación individual

Los primeros sistemas de recomendación surgieron a principios de los 90, una de las primeras aplicaciones donde filtrar mensajes de noticias interesantes de Usenet para particulares. Para entonces, los sistemas utilizaban enfoques colaborativos. Luego, a mediados de los 90, se desarrollaron sistemas de recomendación para álbumes de música y videos. Debido a la utilidad de los sistemas de recomendación, una gran parte de las empresas de Internet mostraron interés en la tecnología, y el camino desde la investigación hasta la comercialización fue bastante corto. Después de una caída durante la burbuja de Internet a principios de la década de 2000, el interés en la tecnología se encendió con el anuncio de 2006 del Premio Netflix, que era una competencia para los mejores sistemas de recomendación colaborativos que podían predecir las calificaciones de los usuarios para las películas. [3] Un conjunto de datos estaba disponible públicamente y el mejor equipo recibió un premio de un millón de dólares estadounidenses. A menudo, con los sistemas de recomendación en línea, el rendimiento es un problema real y se ha invertido mucho esfuerzo en desarrollar algoritmos efectivos. Dado que este no será un problema real en esta tesis, el enfoque principal fue investigar los conceptos básicos de cómo funcionan los sistemas de recomendación.

Dos documentos que resumieron el trabajo previo realizado en el campo, así como también propusieron posibles extensiones futuras donde "Hacia la próxima generación de sistemas de recomendación: una encuesta del estado del arte y posibles extensiones" por G. Adomavicius y A. Tuzhilin [2], y "Sistemas de recomendación" de P. Melville y V. Sindhvani [4]. En estos artículos, se explican la similitud de coseno, el filtrado colaborativo basado en elementos y basado en modelos, así como los conceptos básicos para la recomendación basada en contenido y la ponderación  $tf-idf$ . La forma en que la clasificación bayesiana se puede usar para predecir calificaciones también se describe brevemente en [2]. "Recomendación colaborativa basada en contenido" de M. Balabanovic e Y. Shoham brindaron una revisión más profunda de los algoritmos de recomendación basados en contenido, discutiendo la representación de elementos con valores  $tf-idf$ , métodos de vecino más cercano, retroalimentación de relevancia y el algoritmo de Rocchio, el aprendizaje de modelos de usuario, así como bayesianos ingenuos basados en contenido. En "Técnicas de factorización de matrices para sistemas de recomendación" de Y. Koren, R. Bell y C. Volinsky [13] se investigó el filtrado colaborativo basado en modelos, que fue la base de cómo el ganador del Premio Netflix resolvió el problema. La "Recomendación colaborativa basada en contenido" de M. Balabanovic e Y. Shoham también ofreció una breve introducción a los sistemas de recomendación colaborativos y basados en contenido y explicó la implementación de un sistema de recomendación real que recomendaba páginas web a los usuarios de Internet. [8] Un buen libro sobre el tema con varios ejemplos fue "Recommender Systems: An Introduction" de D. Jannach, M. Zanker, A. Felfernig y G. Friedrich. [3]

#### 4.4.2 Recomendación de grupo

El trabajo que más se parece a esta tesis es el "Pocket RestaurantFinder: Un sistema de recomendación situado para grupos" de JF McCarthy. Recomienda restaurantes a un grupo de personas en función de sus preferencias predefinidas. El sistema podría ejecutarse en dos interfaces de usuario diferentes, un quiosco o una computadora de mano. A diferencia del sistema de esta tesis, el

distancia, precio y cocina general del restaurante se tuvo en cuenta en el Pocket RestaurantFinder. Sin embargo, el menú del día de los restaurantes no se consideró y la recomendación del grupo simplemente consistió en agregar puntajes con solo la estrategia promedio.

En el pasado se han desarrollado varias aplicaciones de recomendación de grupos específicos de problemas. "Estado del arte en recomendación de grupos y nuevos enfoques para la identificación automática de grupos" por L. Boratto y S. Carta [11] y "Recomendación a grupos" Jameson y B. Smyth [15] dieron un resumen del trabajo anterior, que eran sistemas de recomendación grupal para música, libros, páginas web, destinos turísticos y de viajes, programas de televisión y noticias. Debido a la naturaleza ad hoc de los problemas, y que ninguno de los trabajos anteriores tenía suficientes similitudes con esta tesis, se decidió investigar solo algunos de los sistemas mencionados. Los dos artículos también incluyeron discusiones sobre los sistemas de recomendación grupal, sus beneficios y limitaciones. Además de este grupo básico, se explicaron técnicas de agregación de puntajes como promedio y mínima miseria.

Dos ejemplos de sistemas implementados fueron "PolyLens: un sistema de recomendación para grupos de usuarios" de M. O'Connor, D. Cosley, JA Konstan y J. Riedl [16], y "Recomendación de programas de televisión para múltiples espectadores basada en el usuario". fusión de perfiles" por Z. Yu, X. Zhou y JG Yanbin Hao [17]. La primera de estas películas recomendadas a un grupo mediante el uso de un sistema de recomendación colaborativo para obtener calificaciones y luego agregarlas con la medida de menor miseria. El segundo recomendaba programas de TV a un grupo e implementaba un sistema de recomendación de grupos basado en un modelo mediante la fusión de perfiles de usuario. Otro sistema de recomendación de grupo que utilizó un enfoque basado en modelos fue "Un recomendador de grupo para películas basado en la similitud y popularidad del contenido" de MS Pera y Y.-K. Ng. [7] Fusionó palabras clave de los intereses de los usuarios individuales en un perfil de grupo. Además de esto, se utilizó una fórmula combinada para aumentar la recomendación con la popularidad general de un artículo.

Los sistemas de recomendación de grupos a menudo se usan en entornos en línea donde se manejan millones de usuarios y calificaciones. Es por eso que la eficiencia es un problema real, y "Recomendación grupal: semántica y eficiencia" de S. Amer-Yahia, SB Roy, A. Chawla, G. Das [10] y C. Yu, y "Recomendaciones de grupos rápidos aplicando la agrupación de usuarios" por E. Ntoutsi, K. Stefanidis y K. Norvag [12] examinaron este problema. Además, se establecieron técnicas de agregación estándar, y especialmente el desacuerdo promedio por pares que fue propuesto por primera vez por [10]. No se predijo que la eficiencia causara ninguna complicación en esta tesis, por lo que no se tuvo en cuenta.

#### 4.4.3 Calificaciones implícitas

La búsqueda de recopilar calificaciones implícitas para un elemento también es un problema muy específico, pero se han realizado estudios en diferentes contextos. En el artículo "Indicadores de interés implícito" de M.

Claypool, P. Le, M. Wased y D. Brown [21] recopilaron calificaciones implícitas para las páginas web, en función de las actividades de la barra de desplazamiento, el teclado y el mouse en la página. También se registró el tiempo total dedicado a una página. Se pidió a los usuarios que proporcionaran calificaciones explícitas para medir el rendimiento. Mostró que el tiempo pasado en la página estaba claramente correlacionado con la calificación explícita. En "Un recomendador personalizado adaptativo basado en el aprendizaje del comportamiento de navegación web" de K. Takano y KF Li [6], se desarrolló otro sistema de recomendación para páginas web mediante el análisis del comportamiento de navegación y la recopilación de calificaciones implícitas. Las páginas web consultadas, los términos copiados en el portapapeles, las páginas marcadas, las páginas impresas y las palabras clave de búsqueda introducidas en los motores de búsqueda, etc., se registraron y utilizaron como datos de entrada para el sistema de recomendación. Dado que había pocas similitudes entre los trabajos anteriores y esta tesis, era necesario desarrollar un método completamente nuevo.

## Parte II

# Método

### 5 Configuración del sistema

El objetivo de esta tesis fue configurar un sistema donde un usuario pudiera ver los menús de los restaurantes cercanos en una aplicación de Android. Luego, se recomendó al usuario un restaurante en función de las preferencias recopiladas anteriormente. El sistema constaba de un cliente que se ejecutaba en un teléfono inteligente, un servidor, una base de datos, un analizador que recopilaba datos de un sitio web público que contenía los menús y un sistema de recomendación. La estructura general se muestra en la figura 2. Desarrollé el backend del sistema, todas las partes en la figura excepto el cliente.

#### 5.1 Aplicación Android

La aplicación fue desarrollada en los laboratorios de Ericsson en Beijing por estudiantes de la Universidad de Beihang. Presentaba una lista de los restaurantes en el municipio de Kista, un suburbio de Estocolmo donde se encuentra la sede de Ericsson, véase la figura 3. Al señalar uno de los restaurantes, se mostraba el menú del día al operador. No había una recomendación directa de un restaurante para el día, la recomendación era más como un sistema de clasificación donde la lista de restaurantes se ordenaba según las preferencias de los usuarios.

Había dos funcionalidades en la aplicación, la primera era la capacidad de obtener una recomendación grupal para un grupo de personas que almorzaban juntas agregando contactos de la lista de contactos, figura 5. Después de que se agregaron los amigos almorzando, el pedido de los restaurantes en se suponía que la lista cambiaría en consecuencia. La segunda funcionalidad era la capacidad de decirle a uno de los compañeros de almuerzo, supuestamente no presente en la fiesta, a dónde iba a almorzar el grupo, figura 6. De esa forma, la decisión del grupo se le comunicaba implícitamente a los

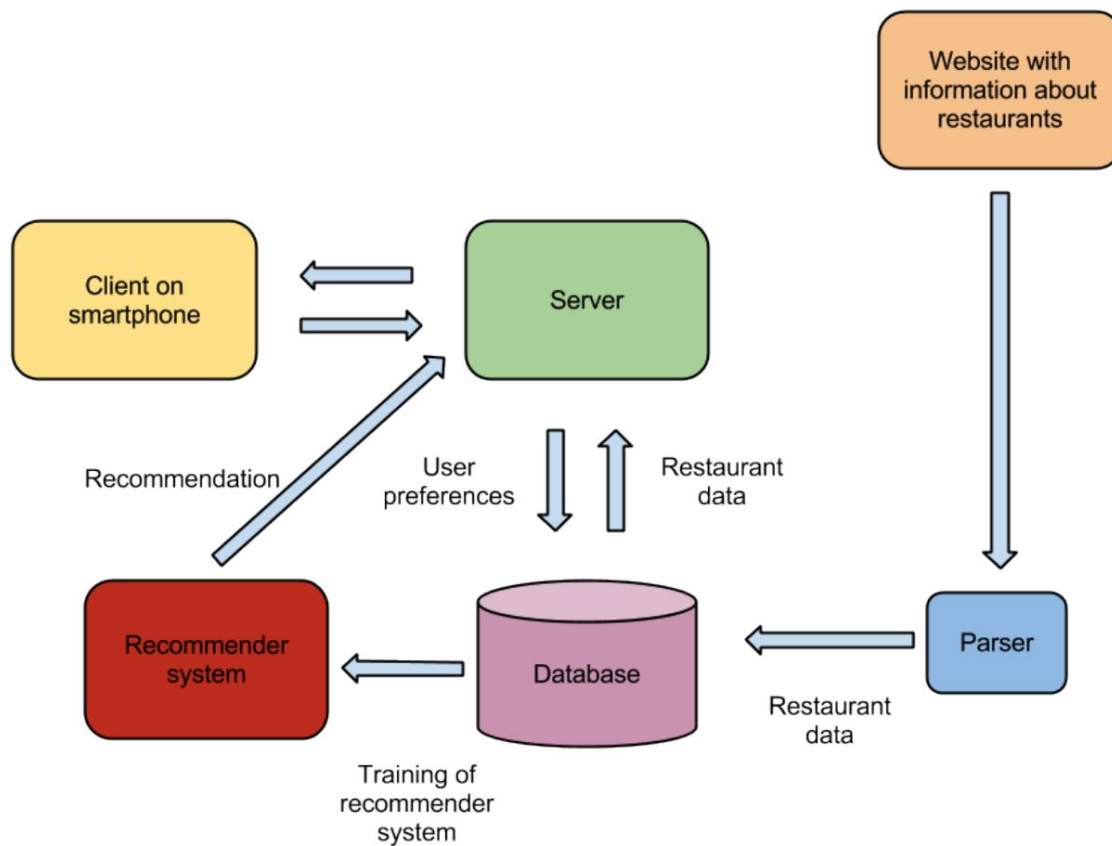


Figura 2: Configuración del sistema para el experimento de recomendación

servidor, y podría ser utilizado en el sistema de recomendación. Las capturas de pantalla de la aplicación se muestran en las figuras 3 a 8.

Los usuarios previstos de esta aplicación eran quizás colegas que almorzaban juntos, personas que ya se conocían. En el departamento de análisis de datos de Ericsson, varios de los compañeros de trabajo almorzaban juntos todos los días en uno de los restaurantes cercanos, y siempre había una discusión sobre qué restaurante visitar. El objetivo de esta aplicación era que una de estas personas en el grupo fuera el operador, luego agregaría a los compañeros de trabajo de la lista de contactos y comenzaría a buscar restaurantes. La suposición era que los compañeros de trabajo cercanos también mirarían el dispositivo móvil que ejecuta la aplicación e influirían en el comportamiento de navegación del operador. Este procedimiento se explicará con más detalle en la sección posterior que cubre el algoritmo de recomendación. Muchas veces sucede que uno o varios de los compañeros de trabajo están atrapados en reuniones u otras citas, por lo que el operador puede decirle a esa persona a dónde va el grupo, como se mencionó anteriormente.

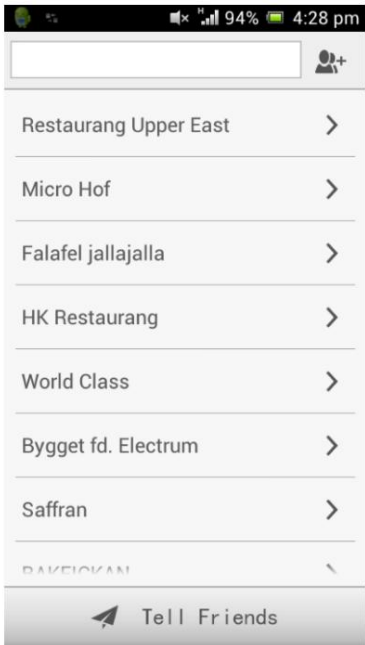


Figura 3: Captura de pantalla de la pantalla de inicio de la aplicación, una lista de restaurantes. El sistema de recomendación ordenó la lista en función de las preferencias del usuario.



Figura 4: Platos en el menú de un restaurante. Después de que el usuario señalara uno de los restaurantes en la figura 3, aparece una lista de los platos disponibles en el restaurante que día fue proporcionado al usuario.



Figura 7: Se podría notificar a varias personas de la lista de compañeros de almuerzo sobre el restaurante seleccionado

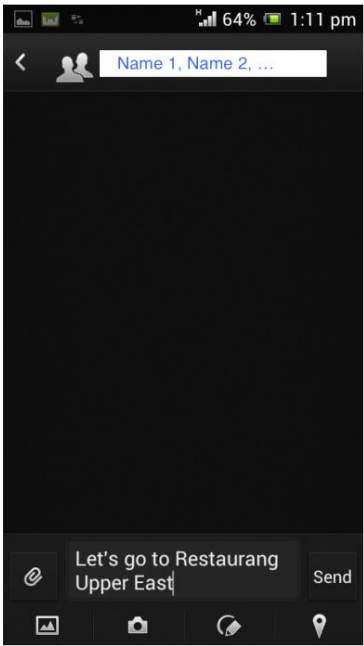


Figura 8: La notificación se envía por sms a los contactos seleccionados.

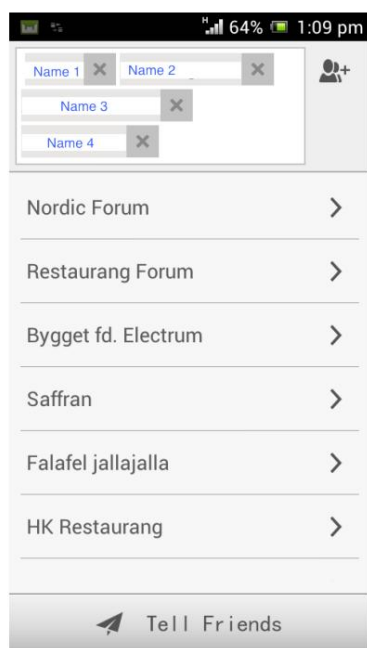


Figura 5: Los amigos que almorzarán juntos se agregan desde la libreta de contactos. Si hay alguna información sobre las preferencias de los amigos, la clasificación de los restaurantes cambiará.

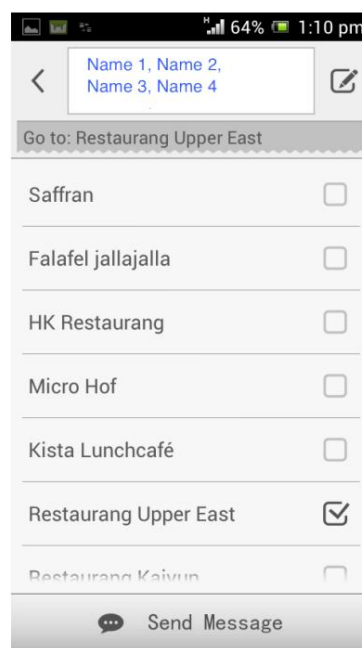


Figura 6: Después de que el partido haya elegido un restaurante, el operador de la aplicación puede optar por decirle a uno o varios de los contactos de la lista a dónde va el grupo.

## 5.2 Servidor

El servidor es responsable de la comunicación con el cliente y la base de datos, y del manejo de la información de identificación de los usuarios, como número de teléfono, número de IMEI<sup>1</sup> y correo electrónico. Hay tres funciones básicas en el servidor, estas son `getRestaurants`, `getDishes` y `writeSelection`. La funcionalidad del servidor se muestra en la figura 9.

**getRestaurants** La función requiere una lista específica de socios para el almuerzo, así como el operador que está utilizando la aplicación en el teléfono inteligente. El servidor llama al programa de recomendación con los usuarios y restaurantes especificados, y el algoritmo de recomendación ordenará los restaurantes de acuerdo con las preferencias del usuario y devolverá la lista ordenada. Los componentes de esta función se muestran en rojo en la figura 9.

**getDishes** Esta función no solo requiere la lista de socios de almuerzo y la identificación del operador, sino también los restaurantes especificados. Esta función se llama cuando el usuario apunta a un

<sup>1</sup>Un número IMEI es un identificador de hardware único para un teléfono móvil, consulte la sección para obtener más detalles.

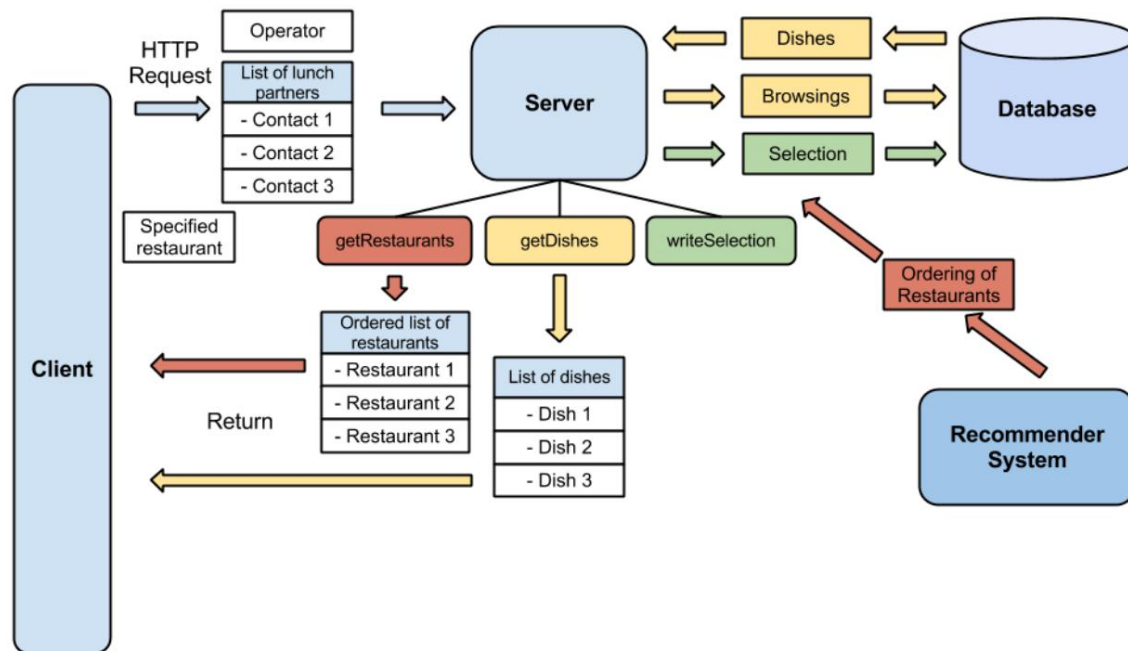


Figura 9: Funcionalidad básica del servidor

restaurante en la aplicación. El servidor leerá los platos del día de la base de datos, que previamente se han guardado allí por la mañana. A continuación, la función devuelve la lista de platos (es decir, el menú) del restaurante mencionado y también escribe una entrada de esta ocasión de navegación en la base de datos. De esa manera se pueden rastrear los intereses de los usuarios. Las funcionalidades de esta función se muestran en amarillo en la figura 9.

**writeSelection** La función se llama cuando el grupo ha tomado una decisión sobre qué restaurante visitar y quiere usar la función "Cuéntale a los amigos", como se describió anteriormente, para decirle a una persona que no está presente en el grupo. La función requiere información sobre el grupo, así como el restaurante seleccionado. El servidor escribe la selección en la base de datos y esta información se utiliza en el algoritmo para impulsar aún más la recomendación. Los componentes de esta función se muestran en verde en la figura 9.

Todas las comunicaciones con el servidor se realizan a través de solicitudes HTTP y solo se devuelven mensajes de texto sin formato. El servidor está escrito como una aplicación web Java que se ejecuta en un



Contenedor de servlets de Apache Tomcat. El acceso desde el servidor a la base de datos se obtiene utilizando el controlador Java DataBase Connectivity, JDBC.

### 5.3 Base de datos

Hay un sistema de gestión de base de datos MySQL ejecutándose en el servidor, con una base de datos que contiene una serie de tablas en las que el servidor escribe y lee. Las tablas se denominan tabla de consumo de id, historial de navegación, platos del restaurante, selección y selección estimada. Además de estas tablas, hay algunas otras que se utilizan con fines de depuración y análisis de datos.

id tabla de consumo Existen varias formas de identificar a un usuario en el sistema. Se decidió usar tres de esos, que son comunes para el uso de identificación en teléfonos inteligentes.

Los identificadores seleccionados fueron número IMEI, número de teléfono y correo electrónico. El IMEI, abreviatura de International Mobile Equipment Identity, es un identificador de hardware único para teléfonos móviles. Es un número de 15 dígitos que va ligado al teléfono móvil, por lo que no se puede cambiar, aunque el móvil cambie de propietario. Si se le da acceso, el número IMEI puede ser leído por una aplicación en el teléfono inteligente. Si el usuario ha especificado su número de teléfono o correo electrónico en el perfil de Google, esta información también puede ser leída por la aplicación de Android. La aplicación también puede acceder a información sobre los socios de almuerzo agregados, su número de teléfono o correo electrónico si están especificados en la libreta de contactos.

Toda esta información se reúne en la tabla de consumo de id. Cada vez que un identificador de usuario ingresa al sistema, el servidor busca en la tabla y ve si el identificador se ha registrado anteriormente. Si esto es cierto, se le asignará al usuario el número de identificación interno correspondiente, como se muestra en la tabla 1. Si la búsqueda no arroja nada, se crea una nueva entrada en la tabla para el usuario con los identificadores especificados.

La razón para usar tantos identificadores es que para hacer una buena recomendación para un grupo de personas, necesitamos conocer las preferencias de la mayor cantidad posible de personas en el grupo. Dado que no se recopilan calificaciones explícitas, lo único que podemos estudiar es el historial de navegación. Cuando un usuario navega por los restaurantes, seguramente se recopilará el número IMEI y, con suerte, se especificará el número de teléfono o el correo electrónico en el perfil de Google. Si esta misma persona se agrega posteriormente como contacto con otra persona que opera la aplicación en otro dispositivo, en el mejor de los casos, uno de los números de teléfono de la libreta de contactos coincide con el número de teléfono especificado en el perfil de Google. A continuación, podemos cargar el historial de navegación del usuario. Si se produce este evento, se denomina coincidencia de operador de contacto. Obviamente, los números IMEI no se pueden leer de la libreta de contactos.

Puede haber varios números de teléfono en la libreta de contactos, por lo que varias columnas se asignan para guardar los de un usuario. Solo se permite un correo electrónico y un número IMEI para identificación de un usuario. Se muestra la estructura de la tabla y ejemplos de las entradas. en la tabla 1.

Tabla 1: Ejemplo de una tabla de consumo de id. Hay en total 10 identificadores donde el primero ha sido asignado al número IMEI, el segundo ha sido asignado a la dirección de correo electrónico, y las columnas tres a diez están guardando números de teléfono. Una coincidencia de operador de contacto tiene sucedió en la primera fila, ya que hay varios números de teléfono vinculados a un número IMEI. En la segunda fila un operador no ha especificado el número de teléfono ni el correo electrónico, por lo que no puede continuar desde allí. En la tercera fila se ha agregado un contacto con solo uno especificado número de teléfono, que aún no ha sido emparejado.

| Identificación interna | Identificador 1                | Identificador 2 | Identificador 3 | Identificador 4... |     |
|------------------------|--------------------------------|-----------------|-----------------|--------------------|-----|
| 1                      | 358506046839050 yo@ejemplo.com | +46733443222    | +4623423423 ... |                    |     |
| 2                      | 142342352324234                | NULO            | NULO            | NULO               | ... |
| 3                      | NULO                           | NULO            | +46790977096    | NULO               | ... |

historial de navegación Esta tabla se escribe tan pronto como un usuario solicita ver el menú de un restaurante. Mientras que el servidor devuelve el menú de hoy para un restaurante específico, el interno número de identificación para el operador y contactos agregados, el nombre del restaurante y una marca de tiempo para la navegación se almacenan en la tabla.

Además de esto, cuando el menú de un restaurante designado se solicita al servidor, el servidor envía una solicitud interna para la clasificación de los restaurantes, utilizando el identificadores de usuario y socio de almuerzo como parámetros. De esa manera, el servidor se da cuenta de lo que posición que ocupaba el restaurante actual en la lista. Se utiliza para potenciar y evaluar la rendimiento del algoritmo de recomendación, como veremos más adelante. Un ejemplo de cómo el la tabla se vería como se muestra en la tabla 2

platos restaurante Esta tabla se utiliza para almacenar todos los platos para cada fecha y restaurante en la base de datos Solo hay tres columnas; fecha, nombre del restaurante y platos. Para más información sobre esta tabla vea la siguiente sección sobre el analizador.

selección y selección estimada Estas dos tablas tienen exactamente el mismo formato como la tabla de historial de navegación en la figura 2. La tabla de selección se escribe después del "Diga

Tabla 2: Ejemplo de cómo se vería la tabla del historial de navegación. Las marcas de tiempo están en el formato día.mes.año.hora.minuto.segundo. El ranking del restaurante consultado es almacenado en la última columna. El operador y las personas en la lista de socios del almuerzo están todos especificados con identificaciones internas.

| Contactos de usuario | marca de tiempo | Restaurante  | Ranking de seleccionados restaurante |
|----------------------|-----------------|--|--------------------------------------|
| 1                    |                 | 13.6.2013.10.33.43 Receta secreta de clase mundial del | 3                                    |
| 1                    | 2,3             | 13.6.2013.10.34.2 restaurante Upper East               | 2                                    |
| 3                    | 4               | 14.6.2013.11.10.19                                     | 10                                   |

Mesa 3: Ejemplo de cómo se verían los platos de un restaurante. La fecha está en el formato día mes año. Los diferentes platos se separan con signos de tres libras.

| Fecha     | Nombre del restaurante          | Platos  |
|-----------|---------------------------------|---|
| 18.6.2013 | Scandic Victoria Tower Raggmunk | med stekt fl"ask och r"yar"orda lingon<br>### Rimmad lax med dillstuvad potatis ###<br>Varm bulgursallad med fetaost, berenjena y cit rondressing 135:-<br>..   |
| 18.6.2013 | Alto Oriente                    | Ort -& citronhoki med f"ank"yal och kokt potatis<br>### Kalvf"arsbiff med chevr"e och grynallad<br>### Veg Indisk lins -& okragryta con korian derkr"am ###<br>Caesarsallad con kyckling eller<br>skaldjur 89:- |

Se ha presionado el botón de amigos y el usuario ha especificado un restaurante al que ir. Interno Los números de identificación de las personas del grupo que comen juntas se guardan en la tabla, así como el nombre del restaurante y una marca de tiempo. El ranking del restaurante seleccionado tenido en la lista también es relevante, por lo que también se guarda.

Si no hay información sobre adónde fue el grupo, se debe seleccionar un restaurante. ser adivinado Esto se hace en el algoritmo de recomendación, que describiré más adelante. Este la selección simulada se guarda en la tabla de selección estimada.

## 5.4 Analizador

La información sobre los menús de los restaurantes están disponibles públicamente en un sitio web. Cuando se solicita un menú de un restaurante al servidor, la información no se recopila

desde este sitio web, llevaría demasiado tiempo, ya que el código HTML debe analizarse en tiempo de ejecución. En cambio, la información sobre los menús se almacena cada mañana en la base de datos, es decir, en la tabla de platos del restaurante. El servidor lee directamente de esta tabla cuando proporciona los menús.

El analizador es responsable de descargar y almacenar los menús en la base de datos. El programa está escrito en Java y utiliza la biblioteca Jericho HTML Parser. Se conecta a la base de datos con el marco JDBC. El proceso se realiza cada mañana con el programador de trabajos cron que ejecuta el archivo .jar. El análisis HTML de los platos a veces no es muy preciso, ya que no existe una forma estándar de separar los diferentes platos para un restaurante determinado en el sitio web. A veces eso solo se hace con una etiqueta <BR>. Pero los diferentes restaurantes y sus respectivos menús están bien estructurados en tablas en el sitio, por lo que se pueden extraer fácilmente.

## 5.5 Sistema de recomendación

El pedido de los restaurantes en la aplicación es manejado por el sistema de recomendación. Está escrito en Python y utiliza la biblioteca de aprendizaje automático Scikit-learn [25]. Hay dos componentes en este sistema, Train and save y Recommender. La estructura general del programa se muestra en la figura 10.

Entrene y ahorre Todas las mañanas, después de descargar los menús del día en la tabla de platos del restaurante, el sistema de recomendación estima las preferencias de cada usuario y restaurante y las guarda en una estructura de datos. Luego, el programa entrena una máquina de vectores de soporte, que ayuda a la recomendación del grupo. Estos dos objetos se guardan con pickle, que es el mecanismo estándar de serialización de objetos en Python. Entrenar y guardar se coloca como un trabajo cron y se realiza por la mañana justo después de que finaliza el análisis.

Recomendador El servidor ejecuta el programa de recomendación cuando recibe una llamada a la función getRestaurants. El servidor especifica las personas del grupo y el recomendador devuelve una lista de restaurantes ordenados. Tanto la máquina de vector de soporte como los objetos de preferencia se deserializan del archivo y el programa los utiliza en tiempo de ejecución.

## 6 Algoritmo de recomendación

### 6.1 Discusión de la elección del método

El trabajo previo que se ha realizado en los sistemas de recomendación, como se ha explicado en el capítulo anterior, ha girado principalmente en torno a la predicción de una puntuación para un ítem aún no visto. Las propiedades que caracterizan a estos sistemas es que el número de opciones es enorme y la

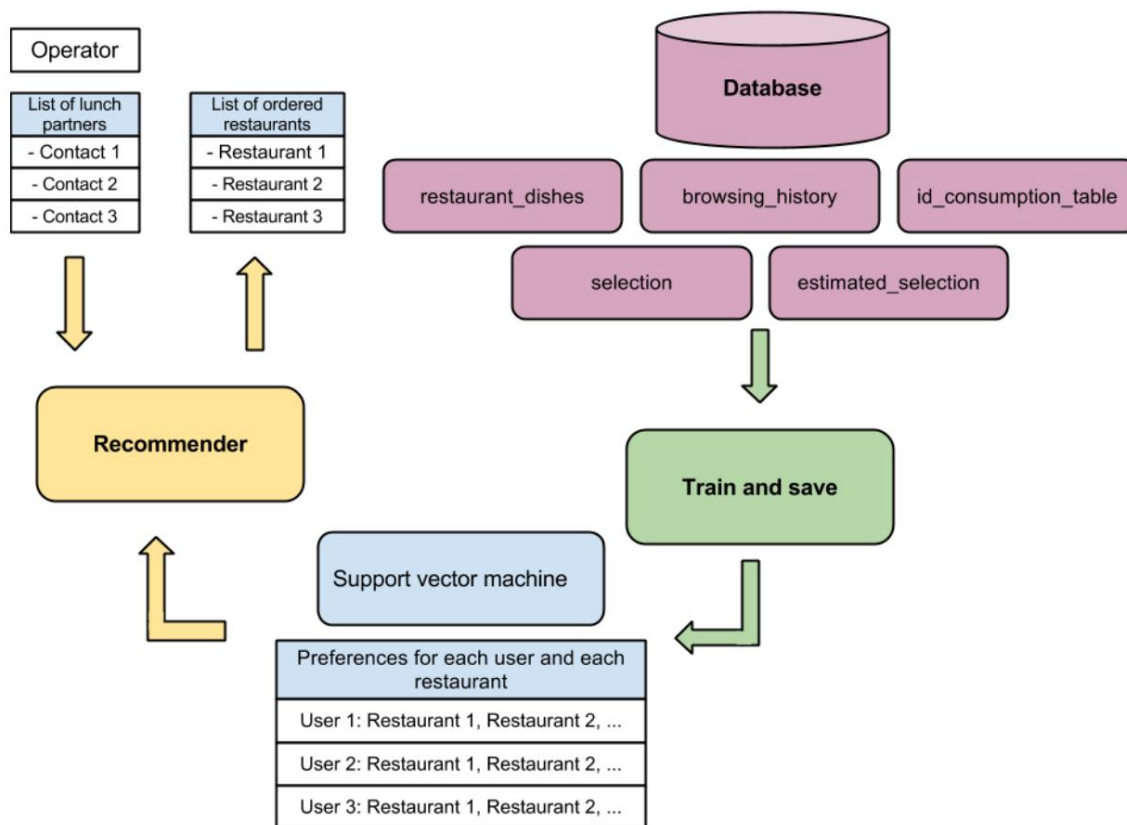


Figura 10: Funcionalidad del sistema de recomendación

el tipo de consumo es tal que se consume el artículo real, no una descripción. En la mayoría de los casos, se proporciona una calificación explícita al sistema justo después de que se haya consumido el artículo real, o al menos después de un tiempo cuando el usuario ha obtenido una opinión sobre el artículo. Tanto en los sistemas de recomendación de música como de películas, que son las dos áreas más destacadas en las que se utiliza la tecnología, hay una afluencia relativamente baja de elementos nuevos. La mayoría de los títulos están disponibles de forma indefinida y son explorados por muchos usuarios. Para estos dos servicios mencionados el impacto de la decisión y el costo del consumo es bajo. Comenzar a ver una mala película o leer un artículo de noticias poco interesante solo hace perder un poco de tiempo al usuario.

Este sistema de recomendación de restaurantes tiene varias diferencias importantes con los otros sistemas mencionados. La más importante es quizás que las calificaciones se recopilan implícitamente a partir del comportamiento de navegación cuando se miran las descripciones de los artículos, antes de que el usuario haya consumido ningún artículo. Otra característica es que un usuario puede tener preferencias por un plato o por un restaurante en particular, o por ambos. El número de opciones es bajo,

incluso existe la posibilidad de que el usuario pueda explorar todas las descripciones de los artículos, es decir, navegar por todos los restaurantes. Para muchos de los restaurantes, constantemente llegan nuevos platos al menú, por lo que hay una gran afluencia de nuevos artículos. El impacto de la decisión puede ser bastante alto, comer bien es importante para la eficiencia del trabajo, por lo que el usuario puede pensar dos veces en la selección del restaurante, en lugar de simplemente mirar la clasificación del restaurante en una aplicación móvil. Se supone que los artículos, es decir, los restaurantes y los mismos platos se consumen varias veces, mientras que los sistemas de recomendación de películas solo recomiendan artículos que no se ven. Sin embargo, hay algunas similitudes con este sistema de recomendación de restaurantes y los sistemas mencionados anteriormente. En este escenario, el coste de consumo y el tiempo de consumo son muy reducidos para la lectura de los menús, mientras que el tiempo hasta el consumo es instantáneo.

Debido a la naturaleza ad hoc de este problema, se tuvo que desarrollar un enfoque completamente nuevo. Después de todo, los métodos que se presentaron anteriormente, especialmente en los sistemas de recomendación grupal, se basan en argumentos psicológicos.

Hay dos beneficios esperados de este sistema, el primero es ahorrar tiempo para los usuarios, proporcionar ejemplos de restaurantes y menús relevantes para que el grupo los discuta y, en un tiempo relativamente corto, tomar una decisión. El segundo beneficio es notificar a los usuarios sobre los restaurantes y especialmente sobre los platos que tienen buenas preferencias y que no sabían que estaban disponibles ese día en particular. Dadas estas premisas, los algoritmos colaborativos estándar, como se explicó en la sección anterior, no son adecuados. Sufrirán el problema del arranque en frío, ya que la escala de tiempo de este experimento es demasiado corta. Además, mirando las búsquedas, el sistema obtendrá relativamente rápido una opinión sobre la preferencia de cada usuario por cada restaurante, ya que hay muy pocos. También es apropiado para analizar el texto del menú, y obtener el gusto de un plato en particular. Finalmente, todas estas calificaciones deben agregarse de manera inteligente, en una sola calificación para un individuo o un grupo. Además de esto, sería bueno tener un modelo que capture la dinámica de selección de grupos.

**Definiciones** Antes de pasar a los algoritmos, introduzcamos algo de terminología. Suponemos que un usuario usa la aplicación solo una vez al día antes de almorzar. Esto se denomina sesión de navegación y el número de restaurantes consultados se denota como  $i_{\max}$ . Al observar todas las sesiones de navegación en diferentes días, se puede calcular el número medio de restaurantes consultados en una sesión, y esto se denota como  $i_{\text{mean}}$ . El orden que tuvo un restaurante en particular en una sesión de navegación, en su navegación más reciente se denota  $i$ . Tenga en cuenta que si se explora un restaurante dos veces,  $i_{\max}$  no aumenta y  $i$  simplemente se actualiza a la última posición del restaurante en la sesión de exploración.

De forma similar al orden de navegación en el restaurante definido anteriormente, se introduce una variable  $j$  como el orden de navegación real. Con el orden de navegación del restaurante varias ocasiones de navegación.

en un restaurante contará como uno. Si un usuario visita el restaurante A, B y C en secuencia una vez cada uno, y luego mira el restaurante A varias veces,  $i_{\max}$  seguirá siendo tres y el orden de navegación para el restaurante A será  $i = 3$ . Esta variable  $i$  está ligada a un restaurante en particular en una sesión de navegación, donde la variable  $j$  está vinculada a una navegación (apuntando a un restaurante y mirando el menú) en una sesión de navegación. De manera similar  $j_{\max}$  es el número total de navegaciones en la sesión. En el ejemplo  $j_{\max} = 5$  y  $j$  es 1, 4 y 5 para cada una de las visitas al restaurante A.

La hora y la fecha en que se realizó una búsqueda o selección de un restaurante deben influir en la calificación. La hora del primer uso de la aplicación por parte del usuario es  $t_0$ , la hora de la fecha actual se denota  $t_{\text{now}}$  y la hora en que se produjo la navegación es  $t_{\text{browse}}$ .

La clasificación que tenía un restaurante en particular en la lista cuando el usuario navegó o seleccionó se denota  $n$ , y el número total de restaurantes disponibles en esa fecha es  $n_{\text{tot}}$ .

El tiempo desde que un usuario solicita el menú de un restaurante hasta que solicita un nuevo menú se extrae de la tabla de navegación en la base de datos. Luego, este número se divide por la longitud del texto (número de caracteres) en el menú navegado para obtener la norma de tiempo de navegación normalizada. El tiempo medio de navegación normalizado para un usuario se calcula sobre todas las ocasiones de navegación y se denota como  $t_{\text{mean}}$ . Hay un umbral de 60 segundos, si el tiempo de navegación es mayor, el tiempo no se extrae de la base de datos. Este es el caso del último restaurante consultado, ya que solo podemos medir el tiempo entre solicitudes.

## 6.2 Preferencia por el restaurante

Para calcular la calificación implícita de un restaurante por parte de un usuario, se recopilan datos del historial de navegación y la tabla de selección. Hay tres cosas que se tienen en cuenta: el orden de navegación, el tiempo transcurrido desde que se realizó la navegación y la clasificación del restaurante consultado.

Orden de navegación Es razonable pensar que el orden de navegación refleja algo sobre la preferencia de los usuarios. Si un usuario siempre comienza a mirar un restaurante en particular, quizás sea más interesante que el último examinado. Cuanto antes el usuario consulte un restaurante en una ocasión de navegación, más importancia merece. El peso que captura esto,  $w_1$ , se define como

$$w_1 = \frac{i_{\max} - i}{i_{\max}} \quad (23)$$

donde  $i$  e  $i_{\max}$  están previamente definidas y  $w_1 \in [0, 1]$

Tiempo desde la navegación o selección Cuanto más atrás en el tiempo se produjo una navegación, menor importancia debe tener a la hora de calcular la preferencia de restaurante. Se supone que las opiniones cambian con el tiempo, y las selecciones más recientes deberían tener mayor relevancia. El peso  $w_2$  se define como

$$w_2 = \frac{t_{\text{buscar}} - t_0}{t_{\text{ahora}} - t_0}, \quad \text{ahora } t = t_0 \quad (24)$$

donde  $w_2 \in [0, 1]$ . Las diferencias de tiempo se pueden traducir a cualquier unidad, preferiblemente segundos, ya que se trata de una fracción que se está calculando. Se garantiza que  $t_{\text{now}} = t_0$ , ya que el cálculo de las calificaciones se realiza por la mañana antes de que ingresen nuevos datos al sistema.

Ranking de restaurantes consultados Si un usuario se desplaza hacia el final de la lista, solo para ver un restaurante en particular, debería significar que es especialmente interesante para el usuario. Las partes superiores de una página desplazable reciben mucha más atención y son mucho más propensas a la selección. Defina el peso  $w_3$  como

$$w_3 = \frac{n_{\text{note}}}{n_0} \quad (25)$$

Donde  $n$  y  $n_{\text{tot}}$  se definen antes y  $w_3 \in [0, 1]$

Interacción grupal Si aumenta el número de personas en el grupo, existe una mayor probabilidad de que uno de varios de ellos haya tenido que comprometerse al seleccionar un restaurante. Para capturar esto, definimos un peso de interacción:

$$w_{\text{accióninvernal}} = \frac{1}{|G|} \quad (26)$$

donde  $|G|$  es el número de personas en el grupo (el número de personas en la lista de compañeros de almuerzo, con el operador incluido) y  $w_{\text{interaction}} \in [0, 1]$ .

Cálculo de puntajes Las fuentes para calcular los puntajes de los restaurantes se toman de la selección y exploración de las mesas de los restaurantes en la base de datos. Si el conjunto de selecciones de todos los usuarios en la base de datos se denota  $S$ , donde  $s \in S$  y el conjunto de búsquedas en restaurantes de todos los usuarios se denota  $B$ , donde  $b \in B$ , podemos definir la puntuación  $r$  para un usuario  $u$  y un restaurante  $yo$  como



$$\begin{aligned}
 \tilde{r}_{iu} = & \sum_{b \in B: \text{restaurante}(b)=i \wedge \text{operador}(b)=u} \frac{1}{\text{acción de invierno}(b) + 2} \frac{w_1(b) + w_2(b) + w_3(b)}{3} + \\
 & \sum_{b \in B: \text{restaurante}(b)=i \wedge \text{amigos}(b)} \frac{\text{acción invernal}(b)}{10} \frac{w_1(b) + w_2(b) + w_3(b)}{3} + \\
 & \sum_{s \in S: \text{restaurante}(s)=i \wedge \text{operador}(s)=u} \frac{10 \text{invierno} \text{interacción}(es)}{2} \frac{w_2(s) + w_3(s)}{2} + \\
 & \sum_{s \in S: \text{restaurante}(s)=i \wedge \text{amigos}(s)} \frac{10 \text{interacción}(es) \text{ de invierno}}{2} \frac{w_2(s) + w_3(s)}{2} \quad (27)
 \end{aligned}$$

donde restaurante(x), usuario(x) y amigos(x) devuelve el restaurante, el usuario o el conjunto de personas en la lista de socios de almuerzo para una búsqueda o selección específica x. El orden de la medida de navegación,  $w_1$ , no dará más información para la selección, normalmente solo hay una selección para un grupo en un día en particular. Sin embargo, la clasificación que tenía un restaurante seleccionado en la lista y el tiempo transcurrido desde que se seleccionó deberían afectar las preferencias para el restaurante seleccionado. Es importante mencionar que los pesos en la fórmula anterior dependen de los datos de la navegación real, la selección, los usuarios que participan y la sesión de navegación. La motivación de esta fórmula se explica por el supuesto de navegación grupal que se describe a continuación.

El valor no normalizado que se obtiene en la ecuación 27 se normaliza dividiendo cada calificación por la norma del vector de calificaciones totales  $r_u$  para un usuario, en el que cada componente es una calificación para un restaurante. El guión superior en la siguiente ecuación se refiere al índice, no al exponente.

$$r_{iu} = \frac{\tilde{r}_{iu}}{|r_u|} \quad (28)$$

**Suposición de navegación en grupo** Se supone que hay tres formas principales de uso de esta aplicación. El primero es un usuario individual que navega por los restaurantes sin personas agregadas en la lista de socios para almorzar, el segundo es un operador y un grupo de socios para almorzar que miran el mismo dispositivo mientras navegan y el tercero es un solo operador con socios para almorzar agregados en la lista. lista, navegando solo. En los últimos dos casos, el operador aún debe tener la mayor influencia en el comportamiento de navegación, pero si hay varios compañeros de almuerzo mirando el dispositivo, o si el operador está planeando un almuerzo con otros, sus opiniones afectarán la decisión de los operadores. Si, por ejemplo, el operador sabe que a la persona A realmente le gusta el restaurante B, el operador podría comprobarlo. Cuantas más personas haya en la lista de compañeros de almuerzo, más compromiso tendrá cada uno en el grupo. Todo esto está cubierto

factores de interacción en la ecuación 27. Siempre queremos aumentar el peso de un restaurante si un operador lo visitó, independientemente de cuántas personas haya en el grupo, pero la puntuación agregada aún debería disminuir con la cantidad de personas. Es por eso que se suma un medio antes de la suma de un medio por el peso de interacción en el primer término de la ecuación 27.

Es razonable pensar que uno de los socios del almuerzo solo tendrá una pequeña influencia en la navegación del restaurante, después de todo, el operador sostiene el dispositivo. Por lo tanto, el segundo término de la ecuación 27 se divide por diez, si tenemos información sobre las búsquedas cuando el compañero de almuerzo era el operador, estas deberían contar mucho más al calcular las preferencias por los restaurantes.

Además, se supone que todas las personas del grupo tienen aproximadamente la misma influencia en la selección del restaurante, por lo que si existe tal selección, se agrega la misma puntuación al restaurante en particular tanto para el operador como para los socios del almuerzo. Estos puntajes se multiplican por diez, como se puede ver en el tercer y cuarto término de la ecuación 27. Una selección dice algo sobre las preferencias reales, por lo que deberían contar mucho más que las búsquedas.

La media se toma sobre los pesos en la ecuación 27, que es una forma decente de agregar cuando los pesos están en un rango común, ya que  $w_i \in [0, 1]$ .

La fórmula anterior es muy ad hoc y solo se basa en argumentos psicológicos. Dados los requisitos de formulación del problema y la falta de datos, ningún otro enfoque parecía posible. No se había hecho nada similar en el pasado y no se podía hacer ninguna optimización para evaluar qué impacto tenía cada uno de los pesos en la decisión, por eso se calculó un promedio simple. También se adivina el tamaño de los factores mencionados: es casi seguro que una selección debería contar más que una navegación, pero no sabemos cuánto más importante es.

### 6.3 Preferencia por la comida

La preferencia por un plato de restaurante se calcula usando la medida tf-idf como se define en la ecuación 16. El algoritmo funciona de manera similar al algoritmo de Roccio, considerando menús relevantes y no relevantes, y calculando un vector de consulta de menú. Dado que la separación de los platos a veces puede ser ambigua, como se explica en la sección 5.4, se utiliza todo el cuerpo del texto del menú de un restaurante al calcular los descriptores tf-idf.

Al tratar de averiguar las preferencias de platos, tenemos en cuenta cuánto tiempo el usuario mira un menú antes de continuar con el siguiente, cuántas búsquedas hizo el usuario en una sesión en comparación con el número medio de búsquedas (cuántos restaurantes que tuvo que visitar). ser examinado antes de que el grupo tomara una decisión) y cuánto tiempo atrás en el tiempo se produjo una búsqueda o selección de un restaurante. También se tiene en cuenta si el operador vuelve varias veces al menú de un restaurante.

Tiempo de navegación Si un usuario solicita inmediatamente un nuevo menú después de mirar los platos de un restaurante, uno puede imaginar que los platos quizás no sean realmente interesantes para el usuario. Si por el contrario el usuario permanece más tiempo en la carta de un restaurante, debería significar que los platos que incluye podrían ser más relevantes. El peso del primer menú se define como

$$v1 = \left( \frac{t_{media}}{t_{norm}} \right)^2 \quad (29)$$

donde  $v1 \in [0, 1]$  y  $t_{norm}$  es el tiempo de navegación normalizado de longitud de texto de un menú, y  $t_{media}$  es el tiempo medio de navegación normalizado de longitud de texto de los menús para un usuario, como se explica en las definiciones anteriores. El motivo de la normalización es que, obviamente, un menú largo tardará más tiempo en leerse.

Número de restaurantes en la sesión de navegación Si el usuario, en promedio, consulta una cantidad media de restaurantes cada día, es razonable suponer que algo ha sucedido si la cantidad de restaurantes en una sesión de navegación,  $i_{max}$ , es drásticamente diferente a esta media en un determinado ocasión. Si el operador deja de mirar los restaurantes después de un corto período de tiempo, existe una alta probabilidad de que el usuario o el grupo encuentren platos que les gusten a todos. Si por el contrario el usuario sigue navegando por los menús, mucho tiempo después de haber alcanzado la media, es quizás porque el usuario no encuentra nada bueno. El peso que captura esto es  $w2$  y se define como

$$v2 = \frac{\left( 1 - \left( \frac{i_{max} - i}{i_{max}} \right)^2 \right) \left( 1 - \frac{i_{max}}{i_{media}} \right)}{\left( 1 - \left( \frac{i_{max} - i}{i_{max}} \right)^2 \right) \left( 1 - \frac{i_{max}}{i_{media}} \right)} \quad (30)$$

donde aproximadamente  $v2 \in [0, 1]$ . Cabe recordar que  $i$  e  $i_{max}$  son variables discretas, donde  $i \in \{1, 2, 3, \dots, i_{max}\}$ . La variable  $i$  es el pedido que tuvo un restaurante en la sesión de navegación donde solo se cuenta su última navegación. Cuanto más cerca esté el número de restaurantes consultados  $i_{max}$  de la media  $i_{media}$ , menor debería ser  $w1$ . Los pesos de los menús en la sesión de navegación en función de  $i$ ,  $i_{max}$  e  $i_{media}$  se representan en las figuras 11, 12, 13, 14, 15, 16, 17 y 18. Cada punto rojo simboliza un restaurante y su menú. Se puede observar que los pesos son fuertemente positivos si solo se han leído unos pocos menús, y empiezan a ser negativos si el usuario sigue mirando restaurantes más allá de la media. En todas las cifras de ejemplo  $i_{media} = 10$ .

Dado que solo se cuenta la última navegación por el menú de un restaurante,  $i_{max}$  nunca puede ser mayor que el número total de restaurantes disponibles en la lista,  $i_{max} \leq n_{tot}$ . La razón para usar ecuaciones de segundo grado es que asumimos que uno de los últimos menús navegados es la selección real para el usuario, que quizás esté satisfecho con el menú. Porque

esto los últimos menús navegados deben tener pesos similares que serán el caso en torno a un máximo cuadrático. Otro argumento es que un usuario debe estar bien informado sobre los contenidos del restaurante antes de seleccionarlo. Si la navegación se produjo demasiado tiempo atrás en la sesión de navegación, es razonable pensar que el usuario vuelve a consultar por última vez el restaurante seleccionado, de lo contrario, los contenidos podrían haber sido olvidados.

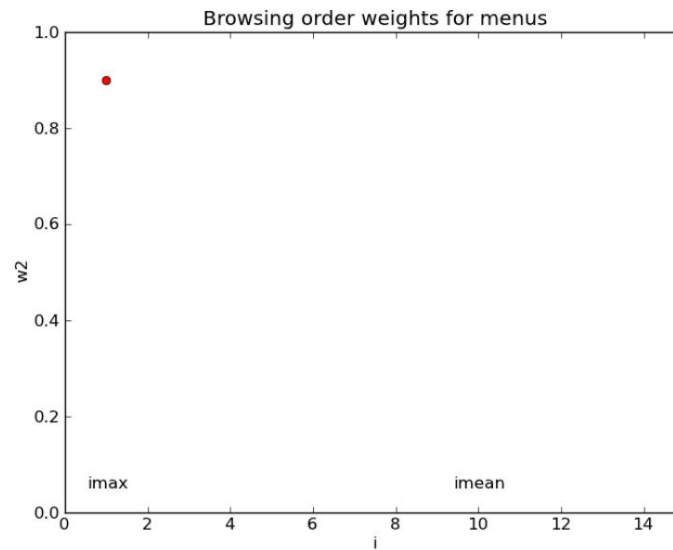


Figura 11:  $imax = 1$ ,  $imean = 10$ , el único menú navegado tiene un peso alto.

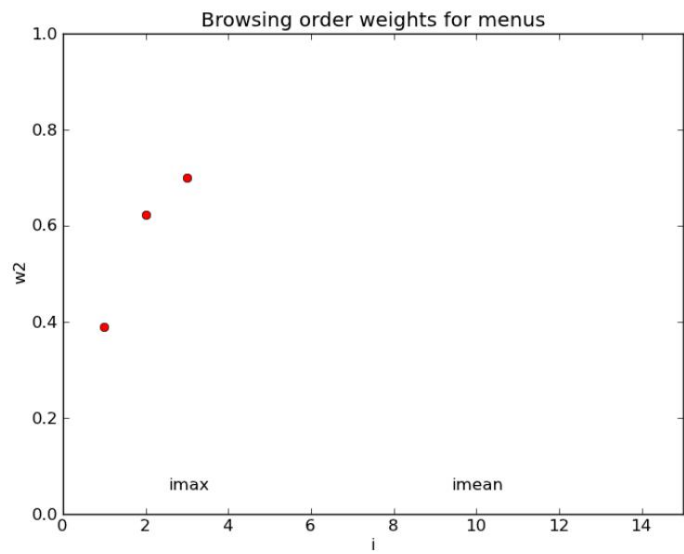


Figura 12:  $\gamma = 1$ ,  $\text{imax} = 3$ , hay varios menús que obtienen pesos altos

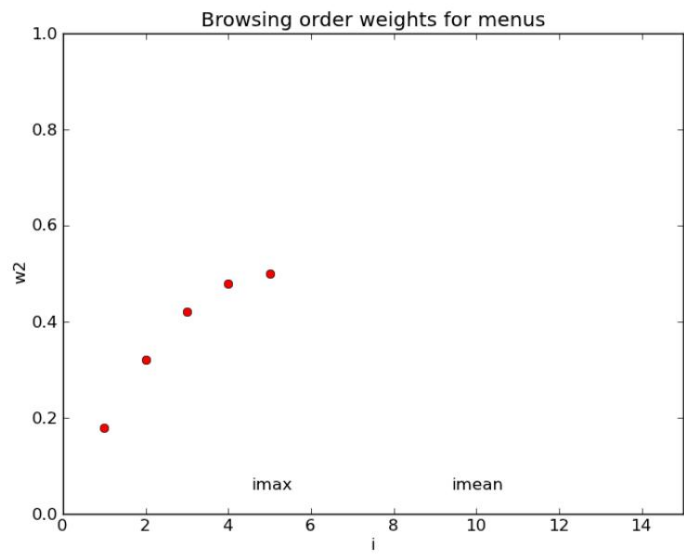


Figura 13:  $\gamma = 1$ ,  $\text{imax} = 5$ , cuanto más nos acercamos a la media, menores son los pesos positivos.

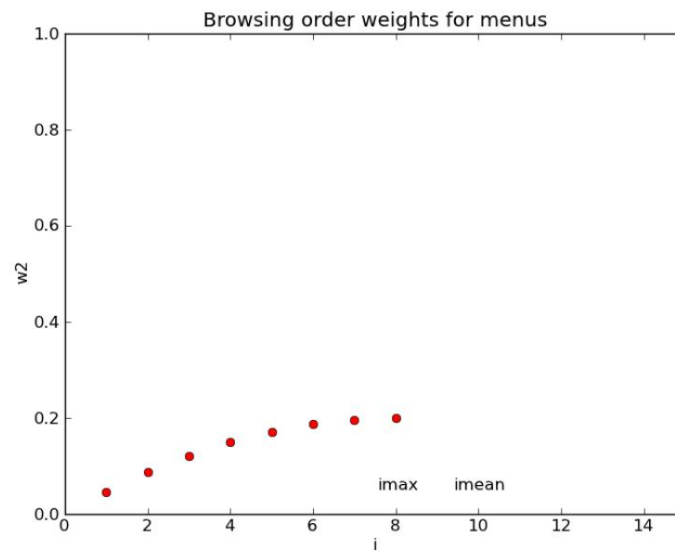


Figura 14:  $\frac{imax}{significa} \rightarrow 1$ ,  $imax = 8$  los pesos se aproximan a cero. En la media, todos se ponen a cero.

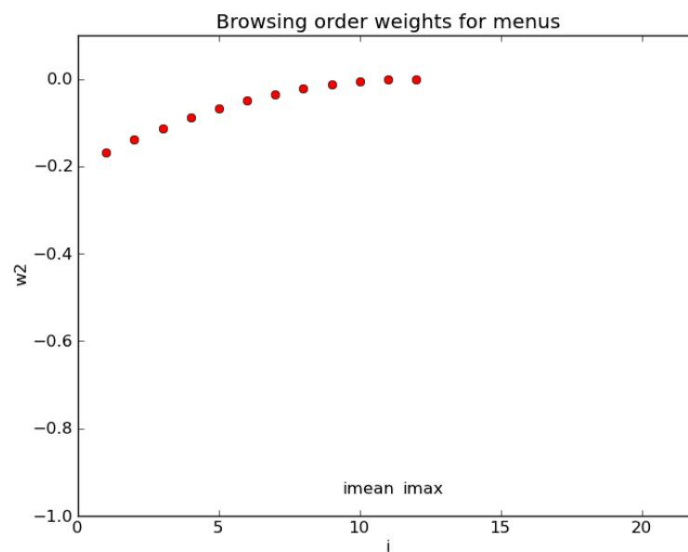


Figura 15:  $1 < \frac{imax}{significa} \rightarrow 2$ ,  $imax = 12$ , los pesos comienzan a ser negativos.

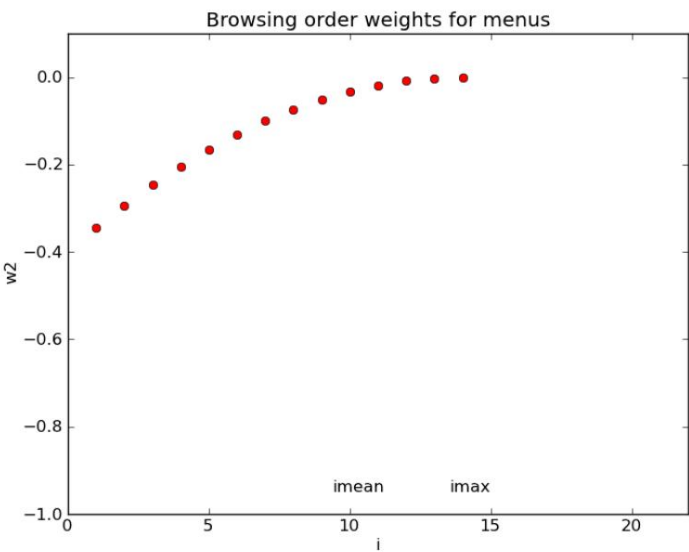


Figura 16:  $1 < \text{media} \leq \text{imax}$  tiene un peso pequeño. Los últimos restaurantes consultados son los

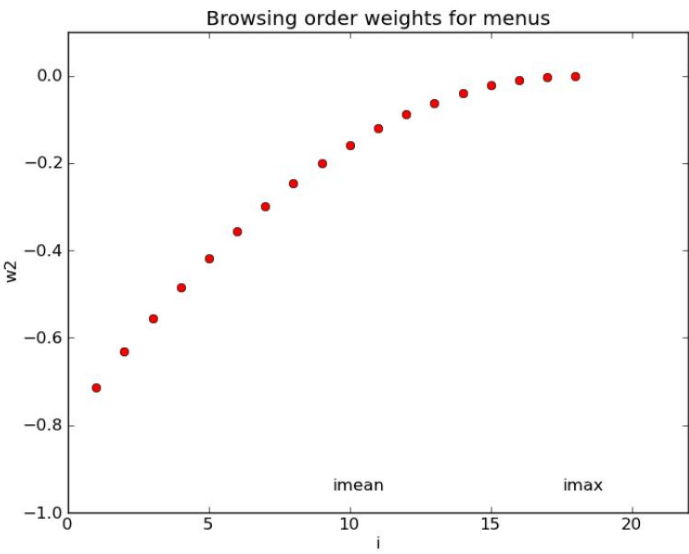


Figura 17:  $1 < \frac{\text{imax}}{\text{media}} \leq 2$ ,  $\text{imax} = 18$ , los pesos disminuyen aún más.

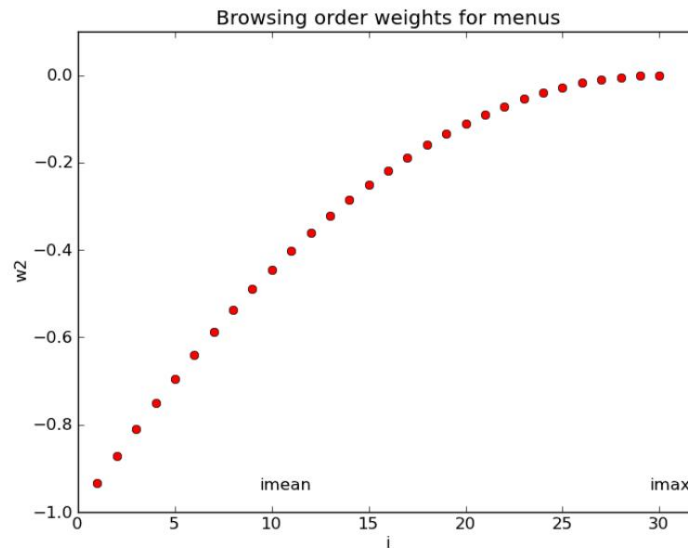


Figura 18: 2 < en  $\frac{imax}{imax} imax = 30$ , queremos  $v2$  y  $y1$ , por eso tenemos la última condición imean , la ecuación 30. En esta figura se cumple esta condición.

Tiempo desde la navegación o la selección Al igual que con las preferencias de restaurante explicadas anteriormente, se supone que las calificaciones cambian con el tiempo. Cuanto más atrás en el tiempo se produjo la navegación por un menú, menor importancia debería tener.  $v3$  se define exactamente como en la ecuación 24.

Volver a navegar Si un usuario vuelve a consultar un restaurante varias veces durante una sesión de navegación, se debe considerar claramente el menú para esa fecha. En parte, este comportamiento de reexploración ya aumenta el peso de la comida, ya que el orden de exploración del restaurante  $i$  se vuelve más bajo, lo que significa que  $v2$  aumenta. Además, varias ocasiones de navegación en un restaurante aumentan aún más las preferencias por su menú, ya que el vector  $tf-idf$  se suma varias veces al vector de preferencia de comida del usuario, como veremos más adelante. Aún así, sería bueno tener un peso que por sí solo capture este patrón de reexploración.

$$v4 = \frac{\sum_{b \in U: \text{restaurant}(b)=i} j \cdot P_{jmax}}{jmax \cdot Una} \quad (31)$$

navegación es como antes denotamos  $b$ , y  $U$  es el conjunto de navegaciones que estaban en la sesión de navegación de un usuario en una fecha particular en la que  $b$  ocurrió, y tenemos que  $|T| = jmax$ . Este peso se calcula para el menú del restaurante  $i$ , por lo que solo queremos considerar las búsquedas de este restaurante,  $\text{restaurant}(b) = i$ . Las variables discretas  $j$  y  $jmax$



se definen previamente, y ambos dependen de la sesión de navegación, navegación, usuario y restaurante seleccionado. Si un usuario selecciona un determinado restaurante muchas veces hacia el final de la sesión de navegación, este peso aumentará. También tenemos que  $v_4 \in [0, 1]$ .

**Interacción de grupo** Se puede argumentar que ocurre exactamente la misma dinámica de grupo cuando se calculan las preferencias de comida que con las preferencias de restaurante. Por lo tanto,  $v_{interacción}$  nuevamente se define como en la ecuación 26.

**Cálculo del vector de perfil tf-idf** Para cada usuario  $u$ , se calcula un vector de preferencia de alimentos tf-idf  $q_u$ .

$$q_u = \sum_{b \in B: \text{operador}(b)=u} \left( \frac{1}{2} + \frac{1}{2} \cdot \frac{v_{interacción}(b)}{v_1(b) + v_2(b) + v_3(b) + v_4(b)} \right) \cdot \frac{v_{interacción}(b)}{10} \cdot \frac{v_1(b) + v_2(b) + v_3(b) + v_4(b)}{4} +$$

$$\sum_{b \in B: u \text{ amigos}(b)} \left( \frac{1}{2} + \frac{1}{2} \cdot \frac{v_{interacción}(b)}{v_1(b) + v_2(b) + v_3(b) + v_4(b)} \right) \cdot \frac{v_{interacción}(b)}{10} \cdot \frac{v_1(b) + v_2(b) + v_3(b) + v_4(b)}{4} +$$

$$\sum_{s \in S: \text{operador}(es)=u} \left( \frac{1}{2} + \frac{1}{2} \cdot \frac{v_{interacción}(es)}{v_1(es) + v_2(es) + v_3(es) + v_4(es)} \right) \cdot \frac{v_{interacción}(es)}{10} \cdot \frac{v_1(es) + v_2(es) + v_3(es) + v_4(es)}{4} +$$

$$\sum_{s \in S: u \text{ amigos}(s)} \left( \frac{1}{2} + \frac{1}{2} \cdot \frac{v_{interacción}(es)}{v_1(es) + v_2(es) + v_3(es) + v_4(es)} \right) \cdot \frac{v_{interacción}(es)}{10} \cdot \frac{v_1(es) + v_2(es) + v_3(es) + v_4(es)}{4} \quad (32)$$

Como en la ecuación 27, todos los pesos en la expresión anterior dependen de la navegación, la selección, los usuarios y la sesión de navegación reales. El tiempo de navegación, el orden en la sesión de navegación y la re-navegación no tienen significado cuando se calcula la preferencia por el menú en una selección, por eso solo se puede usar  $v_3$  en los dos últimos términos de la ecuación 32. El vector tf-idf para el menú de un restaurante se denota  $d_{restaurante}$  (restaurante de la navegación o selección). Exactamente los mismos argumentos sobre el papel de los operadores en la sesión de navegación y la importancia relativa de una selección en comparación con una navegación, como se establece en la ecuación 27 y se explica por el supuesto de navegación grupal, se asumen en la ecuación 32.

Los diferentes pesos en la ecuación 27 tienen diferentes rangos y los dos primeros pueden volverse negativos. Por lo tanto, la media de estos cuatro pesos, tal como se usan en la ecuación, puede tomar valores entre  $-0,5$  y  $1$ . Esto no es un problema, ya que los dos primeros pesos pueden, según los argumentos psicológicos descritos anteriormente, decir algo negativo sobre el menú de un restaurante. Por otro lado, si el menú se vuelve a navegar solo unas pocas veces o se realizó una navegación hace mucho tiempo, esto nunca le dirá nada negativo sobre el menú.

El siguiente paso es calcular un puntaje para cada restaurante  $i$  y usuario  $u$  en función de la comida servido en esa fecha. Esta puntuación se denota  $r_{i,u}$  para evitar mezclarlo con el puntaje en la ecuación.

27. Todas las mañanas se calcula el puntaje de preferencia de alimentos para cada usuario y restaurante, y luego se guarda el resultado para facilitar el acceso.

$$r_{iu}^2 = \frac{\overline{q_u} \cdot \overline{d_i}}{\overline{q_u} \|\overline{d_i}\|} \quad (33)$$

La puntuación se obtiene utilizando la medida de similitud del coseno, mencionada por primera vez en la sección 4.1.1. Los menús que son similares al vector de preferencia de alimentos de los usuarios obtendrán puntajes altos.

## 6.4 Popularidad mundial

Además del restaurante y la preferencia de comida, se pondera una medida final: el gusto global de un restaurante. Simplemente captura cuán popular es un restaurante entre todos los usuarios. Si  $g$  es un vector donde cada componente  $g_i$  es el número de veces que todos los usuarios han solicitado el menú del restaurante  $i$ , entonces podemos calcular una puntuación popular para el restaurante  $i$  normalizando el vector.

$$r_{iu}^3 = \frac{g_i}{\|g\|} \quad (34)$$

Como se ve en la ecuación anterior, la calificación es independiente del usuario, por lo que  $r_{iu}^3 = r_{yo}^3 = r_{iu}^3$ .

## 6.5 Agregación de resultados

Finalmente, las clasificaciones de los restaurantes se agregan calculando puntajes individuales para los elementos. Hay muchas maneras de hacer esto, para un estudio extenso ver el artículo de D. He y D. Wu, "Toward a robust data fusion for document retrieval". [22] El método seleccionado es CombMNZ indicado en la ecuación 36. Este método se ha utilizado en el pasado y ha demostrado dar buenos resultados. [23, 22]

$$r_{iu}^c = \frac{\max_{y \in \mathcal{O}} \min_{i \in \mathcal{U}} (r_{yi}^c)}{\max_{y \in \mathcal{O}} (r_{yi}^c) \text{ y } \min_{i \in \mathcal{U}} (r_{yi}^c)} \quad (35)$$

$$r_{iu}^c = \frac{\max_{y \in \mathcal{O}} \min_{i \in \mathcal{U}} (r_{yi}^c)}{\max_{y \in \mathcal{O}} (r_{yi}^c) \text{ y } \min_{i \in \mathcal{U}} (r_{yi}^c)} \quad (36)$$

$$\text{umbral}(x) = \begin{cases} 1 & x > 0 \\ 0 & x \leq 0 \end{cases} \quad (37)$$

En las expresiones anteriores,  $c \in \{1, 2, 3\}$  se refiere a los tipos de puntuación: preferencia de restaurante, preferencia de comida y popularidad del restaurante. En la ecuación 35 primero se normalizan las puntuaciones y en la ecuación 36 se juntan en una puntuación agregada. El segundo término en la ecuación

36 está ahí para poner más énfasis en los elementos donde existen calificaciones distintas de cero para todos los tipos de calificación. La función de umbral en la ecuación 37 ayuda a lograrlo.

Si no se agregan personas a la lista de socios para el almuerzo, todos los restaurantes se ordenan de acuerdo con sus puntajes respectivos de la recomendación individual. Ahora podemos calcular una recomendación individual para un usuario.

## 6.6 Recomendación de grupo

Luego viene la tarea principal de esta tesis de maestría: utilizar los datos recopilados para obtener una recomendación de grupo para un grupo determinado de personas. Aquí se utilizan dos métodos para obtener puntajes para un elemento presentado a un grupo, y estos puntajes luego se suman.

### 6.6.1 Agregación de grupos

En esta parte, los puntajes agregados individuales se agregan una vez más en puntajes grupales para los ítems que utilizan la medida promedio, menor desdicha y desacuerdo mutuo, explicada anteriormente en la sección 4.4.2, ecuaciones 19, 20 y 21. Se obtiene un puntaje final mediante una ecuación lineal. combinación de estos valores.

$$\text{puntuación1}(i, G) = w_1 \times \text{avr}(i, G) + w_2 \times \text{lms}(i, G) + w_3 \times (1 - \text{dis}(i, G)) \quad (38)$$

El grupo de personas que van juntas se denota  $G$ . En la fórmula anterior, todos los pesos se establecen en números iguales,  $w_i = 1/3$ , y todas las funciones se establecieron previamente en la sección 4.4.2.

### 6.6.2 Máquina de vectores de soporte para recomendación de grupo

Si existe una selección para un grupo en la tabla de selección, esta información se usa para entrenar una máquina de vectores de soporte para capturar la dinámica del grupo. La entrada al algoritmo es el conjunto de usuarios que estaban en el grupo y la salida es el restaurante. Implica que todos los restaurantes son de diferentes clases, dado un grupo de usuarios es una tarea de clasificación saber a dónde irán. De hecho no solo queremos obtener una clasificación, el objetivo principal es calcular una probabilidad para cada clase. La biblioteca de aprendizaje automático utilizada, scikit learn, tiene soporte incorporado para SVM multiclase, así como un método para la estimación de probabilidad de clases propuesto por primera vez por John C. Platt, conocido como escalado de Platt.[24][25] Para obtener información más detallada, considere un libro de texto sobre el tema y el artículo de Platt.

**Extracción de características** La tarea es descubrir cómo representar un grupo como un vector, y debe basarse en una lista dada de grupos de almuerzo y sus respectivos miembros en la tabla de selección. Suponga que todos los usuarios son nodos y los bordes entre dos usuarios simbolizan si ambos han estado en un grupo almorzando juntos. Podría ser

razonable pensar que todos los usuarios de un gráfico conectado podrían comer juntos. Dado que hay 18 usuarios, con los ID internos del 1 al 18 que se muestran en la figura 19, suponga que los siguientes grupos han almorzado juntos: {1, 6, 18}, {18, 4}, {18, 17}, {17, 14}, {17, 8}, {7, 13, 9}, {13, 3}, {9, 15, 12}, {15, 16}, {10, 2, 11}, {5, 2, 11} y {5, 4, 2}. Dos cualesquiera de estos conjuntos que tengan una intersección, es decir, al menos un número en común, se combinan en el mismo grupo.

Esto se hace de manera iterativa hasta que solo sale una cantidad de conjuntos disjuntos, llamados grupos de usuarios. Este método se conoce como consolidación de conjuntos. Los grupos mencionados anteriormente se combinan en tres grupos de usuarios separados, {1, 6, 18, 14, 17, 8}, {3, 13, 7, 9, 12, 15, 16}, {10, 2, 11, 5, 4} que se muestra en la figura 19 como azul, verde y rojo. Para cada uno de estos gráficos en la figura, se entrena un predictor de probabilidad multiclase SVM. La entrada es un vector binario de las personas en el gráfico que estaban en el grupo. La posición de cada persona en este vector binario se fija después de que se haya realizado la combinación del grupo. En la ecuación 39 se muestra un ejemplo de cómo se vería la creación de este vector de características para cada uno de los tres grupos de usuarios diferentes.

$$\begin{aligned} \{1, 6, 18\} &\ddot{y} [1, 1, 1, 0, 0, 0] \{9, \\ 5, 12\} &\ddot{y} [0, 0, 0, 1, 1, 1, 0] \{5, 4, \\ 2\} &\ddot{y} [0, 1, 0, 1, 1] \end{aligned}$$

(39)

**Selección simulada** Si un grupo ha realizado una búsqueda, pero no se ha enviado ningún mensaje y, por lo tanto, no existe una selección de restaurante, se simula una selección. Esto se basa en los mismos pesos que estiman el gusto por un menú. Dado un conjunto de restaurantes en la lista,  $i \ddot{y} I$ , se supone que el restaurante seleccionado es seleccionado

$$i_{\text{selected}} = \arg \max_i w_1 + w_2 + w_3 \quad (40)$$

La selección estimada se escribe en una tabla en la base de datos llamada selección simulada.

Tanto los datos de la selección simulada como la tabla de selección se utilizan al realizar la consolidación de conjuntos y extraer clústeres de usuarios, así como al entrenar sus respectivas máquinas de vectores de soporte.

**Estimación de probabilidad** La entrada al sistema es un grupo de personas,  $G$ , que consiste en el operador y los amigos en la lista de compañeros de almuerzo. Se ha extraído un conjunto de clústeres de usuarios  $G_1, G_2, \dots, G_y, \dots, G_N$  al realizar la consolidación del conjunto. Es posible que algunos de los usuarios de  $G$  no estén en ninguno de los grupos de usuarios, ya que la consolidación de conjuntos solo se realiza en la sesión de preparación y guardado cada mañana. sin embargo, todavía tenemos que averiguar cuál de los grupos es más similar a  $G$ . El grupo seleccionado y seleccionado se decide de la siguiente manera

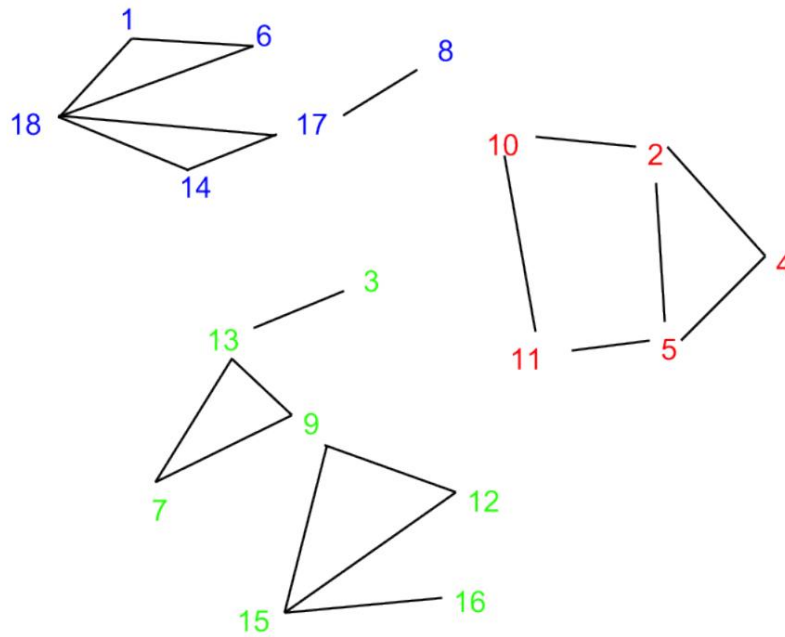


Figura 19: Red de compañeros de almuerzo. Cada número simboliza a una persona, y una conexión entre dos personas significa que han almorzado juntas, ya sea en un grupo más grande o solo las dos. Diferentes colores representan diferentes grupos de usuarios.

$$y_{\text{seleccionado}} = \arg \max_y |G \cap G_y| \quad (41)$$

Después de seleccionar un conglomerado, el grupo se traduce a un vector de características binarias de manera similar a la ecuación 39. Los miembros del grupo que no forman parte del conglomerado simplemente se omiten. Finalmente la probabilidad de que el grupo  $G$  vaya al restaurante  $i$  se calcula con el SVM,  $\text{prob}(i, G)$ . Esto también se establece como la puntuación.

$$\text{puntaje2}(i, G) = \text{prob}(i, G) \quad (42)$$

Si solo hay una persona en el grupo, obviamente podría no haber una recomendación grupal y, por lo tanto, esos ejemplos no se consideran tanto durante el entrenamiento como durante la estimación del puntaje del grupo. Además, si todavía hay una intersección demasiado pequeña entre el grupo y el grupo de usuarios seleccionado, la predicción de probabilidad no será muy precisa. Por eso  $|G \cap G_y|$  razón por la que se establece un umbral  $> 0.5$ . Si no se alcanza esta condición, no se devuelven probabilidades  $G_y$  del método de máquina de vectores de soporte.

### 6.6.3 Agregación de resultados de recomendadores de grupo

Si existen calificaciones tanto para la ecuación 38 como para la 42, las puntuaciones se agregan exactamente de la misma manera que en la ecuación 36. Si faltan las puntuaciones de una de ellas, si por ejemplo hay una intersección demasiado pequeña entre los grupos como se explicó anteriormente, solo la puntuación que existirá servirá como puntuación de grupo para el elemento, sin necesidad de agregar. Si no se pudo extraer ninguna información sobre los contactos en la lista de socios del almuerzo, solo la calificación individual de los elementos del operador servirá como puntaje grupal. Finalmente se ordenarán los restaurantes según la puntuación del grupo y tendremos una recomendación de grupo.

## Parte III

## Resultados

Después de algunas demoras, el equipo de Beijing finalmente desarrolló la aplicación de Android a fines de mayo. Luego se llevó a cabo una amplia campaña de marketing, donde se repartieron mil volantes en el área con instrucciones para descargar y utilizar la aplicación. Además de esto, se visitaron todos los restaurantes disponibles en la aplicación, y la mayoría accedió a colocar folletos en su restaurante.

A pesar de este esfuerzo, la recepción de la aplicación fue modesta en el mejor de los casos. Solo un total de 47 usuarios descargaron la aplicación, y realizaron en total 596 navegaciones y 42 selecciones, pero solo 473 de las navegaciones y 37 de las selecciones fueron utilizadas en el sistema, ya que un requisito era que se debían realizar sesiones de navegación o selecciones. entre las 10 de la mañana a las 15 de la tarde, ya que estos horarios solo son relevantes para almorzar. Del número total de navegaciones, a 51 se les agregaron compañeros de almuerzo, pero solo 28 de estas navegaciones grupales se realizaron en horarios razonables. Además de esto, se simulaban 16 selecciones. Esto significa que se realizaron bastantes selecciones sin navegar, los usuarios preferían enviar los mensajes directamente. Pero, por otro lado, es difícil sacar conclusiones de esto, ya que todas las selecciones fueron realizadas por solo ocho personas.

La tabla de consumo de identificación contenía 73 usuarios, pero solo 5 de ellos tenían información sobre el número de IMEI y el número de teléfono, lo que significa que se produjeron como máximo cinco coincidencias de operador de contacto. Pero lo más probable es que los usuarios tuvieran su número de teléfono escrito en sus perfiles de Google, y que esta información fuera registrada cuando el usuario abrió la aplicación. El número total de restaurantes disponibles era de alrededor de 25, pero eso cambió durante el transcurso del proyecto. Sin embargo, dado que el código del analizador se escribió de manera robusta, esto no causó ningún problema.

## 7 Rendimiento

Dado que las calificaciones no se recopilan explícitamente para la comida y los restaurantes, es difícil encontrar una manera de medir el rendimiento del sistema de recomendación. En el desarrollo normal de un sistema de recomendación, los ingenieros pueden acceder a un gran conjunto de datos que contiene calificaciones de los usuarios para diferentes elementos. Este conjunto de datos se divide en un conjunto de entrenamiento y un conjunto de prueba, y el algoritmo entrenado intenta estimar las calificaciones de los elementos en el conjunto de prueba. Al medir la diferencia entre las calificaciones estimadas y reales, se puede determinar el desempeño. Esto no se puede hacer en nuestro caso, ya que no tenemos calificaciones definidas. Tampoco se puede contactar a los usuarios para hacer algún tipo de encuesta donde evaluarían el sistema. Sin embargo, hay otra forma de obtener algún tipo de indicación de qué tan bien el algoritmo hace la recomendación. El objetivo del sistema es ordenar los artículos con la mayor precisión posible, de modo que el usuario no tenga que desplazarse hacia abajo para encontrar un buen artículo. Proponemos que cuanto más alta sea la posición en la lista de los elementos consultados o seleccionados, mejor será el rendimiento del algoritmo de recomendación. Además, si el número medio de navegaciones en una sesión de navegación tiende a disminuir, esto debería significar que los usuarios encuentran lo que buscan más rápido y, por lo tanto, aumenta la satisfacción del usuario.

### 7.1 Ranking de restaurantes consultados

Lo primero que se investigó es cómo cambian con el tiempo las clasificaciones de los restaurantes consultados. Los resultados se muestran en las figuras 20, 21 y 22. Se analizó el conjunto total de navegaciones de todos los usuarios, y este conjunto se dividió en tres partes según la marca de tiempo de cada navegación: del 15 de mayo al 5 de junio, del 5 de junio al 5 de junio 25 y finalmente del 25 de junio al 7 de julio. Se registró el ranking que tenía el restaurante consultado en la lista, y esto se muestra en los diagramas de bloques. Nuestra hipótesis fue que después de que pasa el tiempo y el algoritmo aprende las preferencias del usuario, las búsquedas se concentrarán más en los primeros elementos de la lista. Sin embargo, parecía que sucedía lo contrario, la tasa de navegación de los elementos principales disminuyó con el tiempo. Aún así, es interesante ver que la suposición de que es más probable que un usuario navegue por uno de los elementos principales de una lista, independientemente de sus preferencias personales, como se describe en la ecuación 25, sección 6.2, parece ser válida.

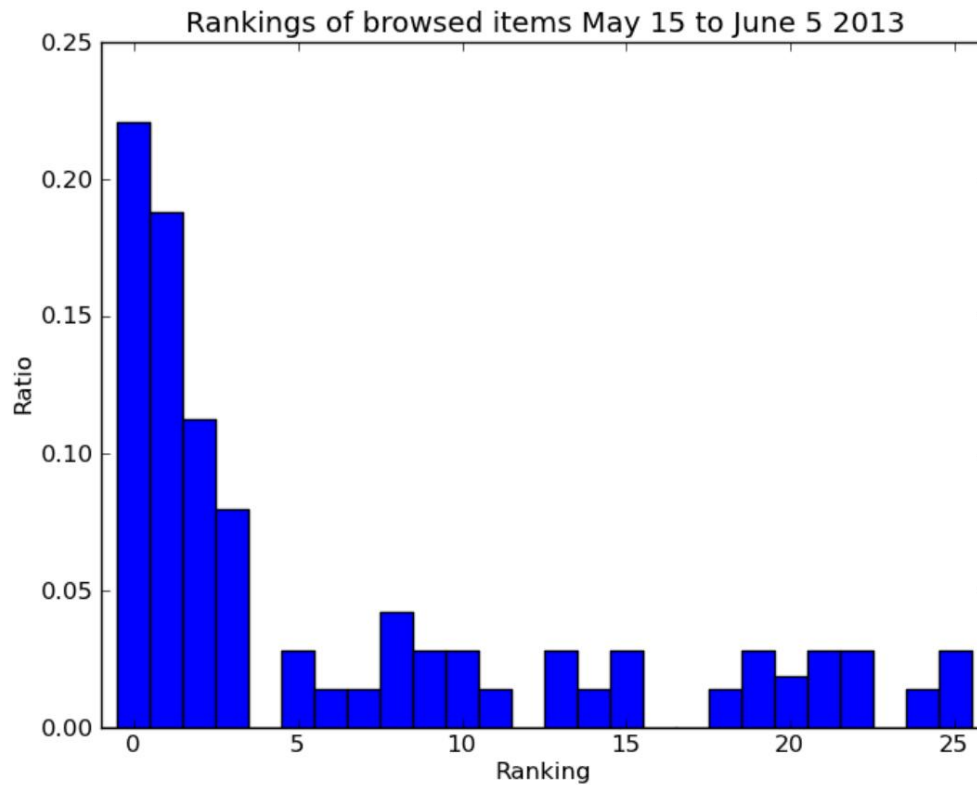


Figura 20: Clasificación de los restaurantes consultados por todos los usuarios entre las fechas especificadas. El restaurante con clasificación cero es el más alto en la lista, y el restaurante con 25 en la clasificación está al final. Las barras están normalizadas, por lo que todas las proporciones sumarán uno. Durante este tiempo, se enviaron al sistema un total de 213 solicitudes de navegación.



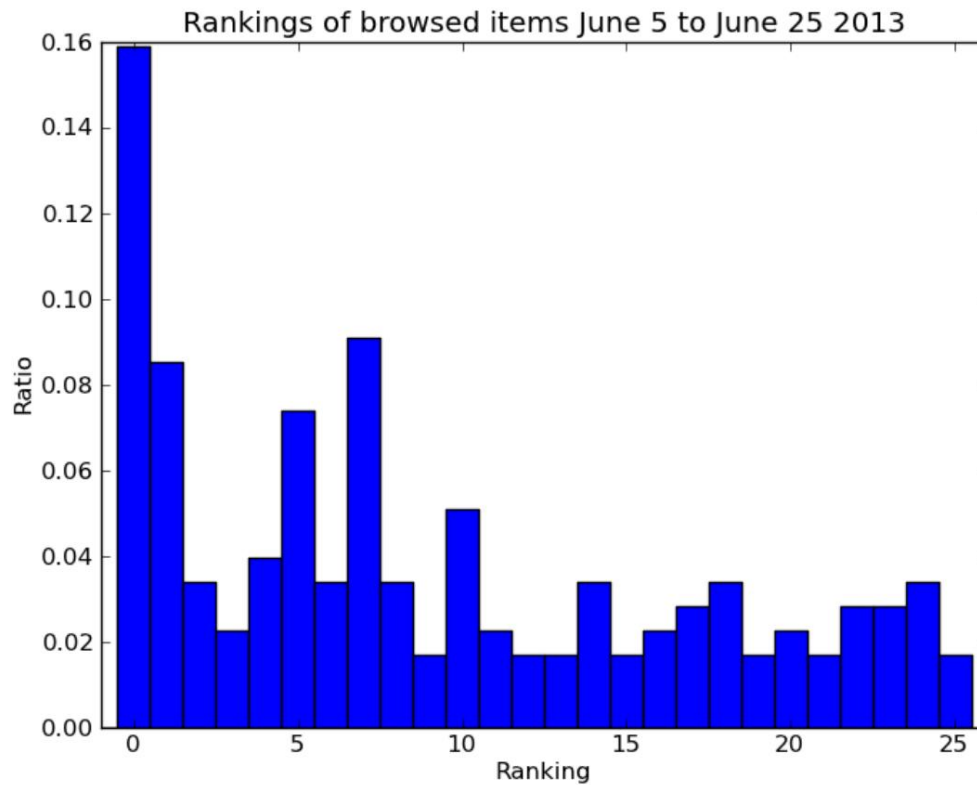


Figura 21: Clasificación de los restaurantes consultados por todos los usuarios entre las fechas especificadas. El restaurante con clasificación cero es el más alto en la lista, y el restaurante con 25 en la clasificación está al final. Las barras están normalizadas, por lo que todas las proporciones sumarán uno. Durante este tiempo, se enviaron al sistema un total de 176 solicitudes de navegación.

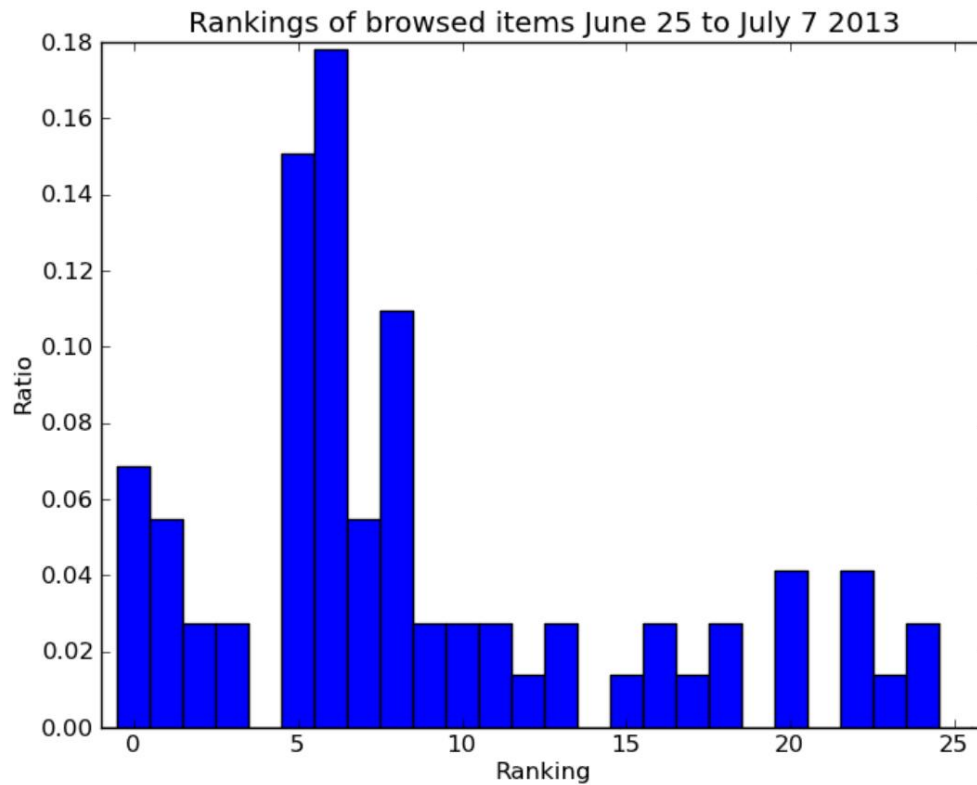


Figura 22: Clasificación de los restaurantes consultados por todos los usuarios entre las fechas especificadas. El restaurante con clasificación cero es el más alto en la lista, y el restaurante con 25 en la clasificación está al final. Las barras están normalizadas, por lo que todas las proporciones sumarán uno. Durante este tiempo, se enviaron al sistema un total de 84 solicitudes de navegación.

En estos diagramas de bloques se incluyen todos los tipos de navegación, tanto aquellas en las que se agregaron compañeros de almuerzo como aquellas en las que el operador navegaba solo. Dado que muy pocos utilizaron la recomendación grupal, los diagramas son principalmente una indicación de cómo se desempeñó el sistema de recomendación individual. Desde mediados de mayo, un pequeño grupo de Ericsson usó una versión beta de la aplicación, pero a fines de mayo se puso en línea la aplicación completa y la gran mayoría de los nuevos usuarios descargaron y comenzaron a usar el servicio a principios de mayo. Junio. Como veremos más adelante, debido al comportamiento de uso de la aplicación, no se pueden sacar conclusiones reales de la clasificación de los elementos consultados.

## 7.2 Número de navegaciones en una sesión de navegación

Si el sistema funciona satisfactoriamente, debería disminuir la cantidad de búsquedas que un usuario o un grupo deben realizar antes de tomar una decisión. Una vez finalizada la sesión de navegación, se supone que el grupo o el usuario individual han tomado una decisión. El número medio de navegaciones en una sesión de navegación se muestra en la figura 23. Esto se basa en todas las sesiones de navegación de todos los usuarios que ocurrieron en cada fecha. Si solo un operador realiza una sesión de navegación en un día, se toma como media el número de restaurantes consultados en esta sesión. Del diagrama es difícil ver mejoras con el tiempo, pero como también veremos más adelante, es difícil sacar conclusiones de los diagramas dados los datos del usuario. Un fenómeno interesante es que los fines de semana crean espacios en el diagrama, donde no se realizan sesiones de navegación y la media es cero.

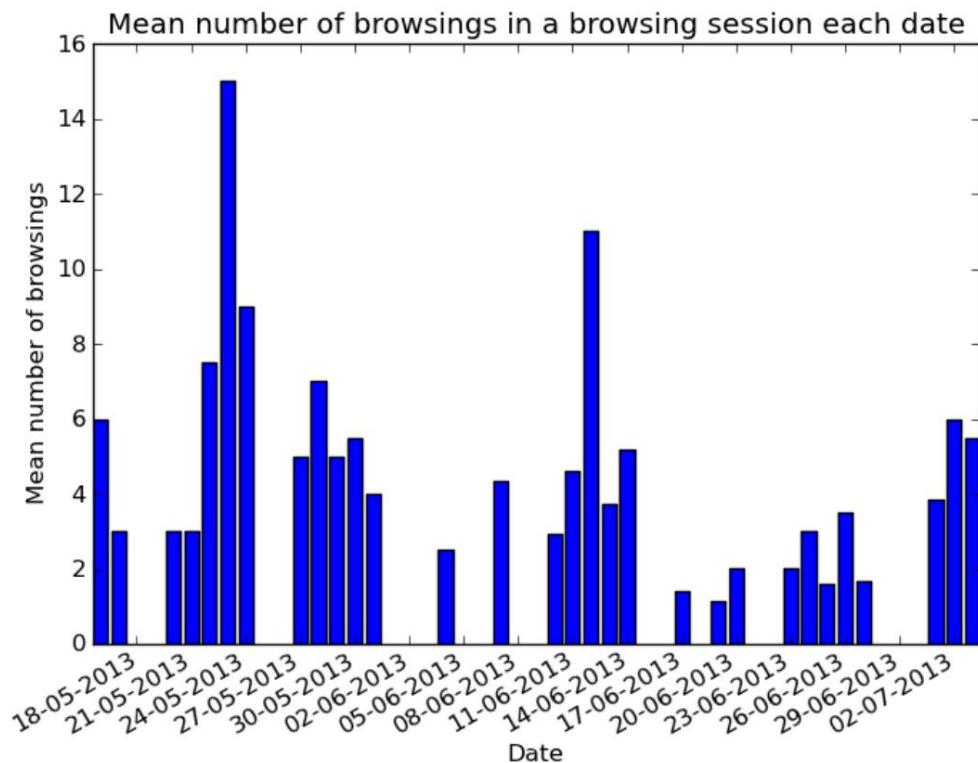


Figura 23: Número de navegaciones en una sesión de navegación. El número medio de navegaciones en una sesión de navegación se calcula en cada fecha.

### 7.3 Clasificación de elementos seleccionados y búsquedas grupales

Debido a la escasez de búsquedas grupales y registros de elementos seleccionados en la base de datos, dividirlos en subconjuntos según el tiempo y mostrar varios diagramas de barras, como se hizo anteriormente con todas las búsquedas, no dará ninguna idea. Los datos serán aún más escasos y los gráficos no serán de utilidad, por lo que no podemos realizar un seguimiento del rendimiento del algoritmo de recomendación de grupo a lo largo del tiempo. Las figuras 24 y 25 muestran las clasificaciones de los elementos seleccionados y consultados con personas añadidas como compañeros de almuerzo. Una selección solo se puede registrar en un contexto de grupo, las dos parcelas se pueden interpretar como el rendimiento del sistema de recomendación de grupo.

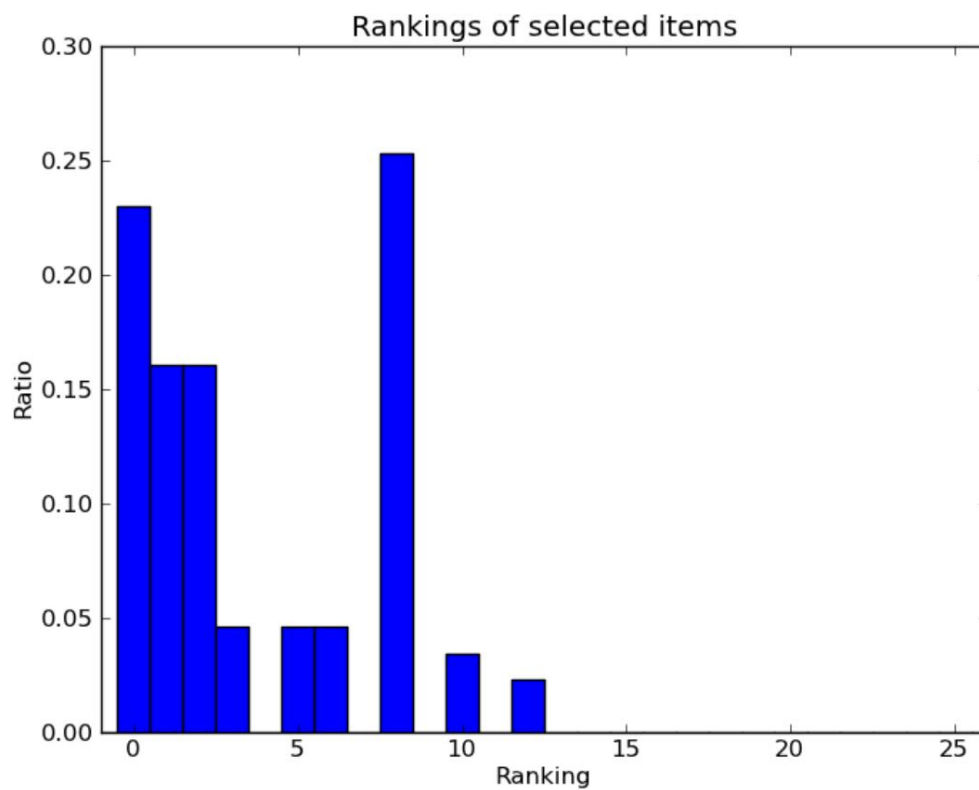


Figura 24: Posiciones que tenían los restaurantes seleccionados en la lista. En la base de datos se registró un total de 37 selecciones.

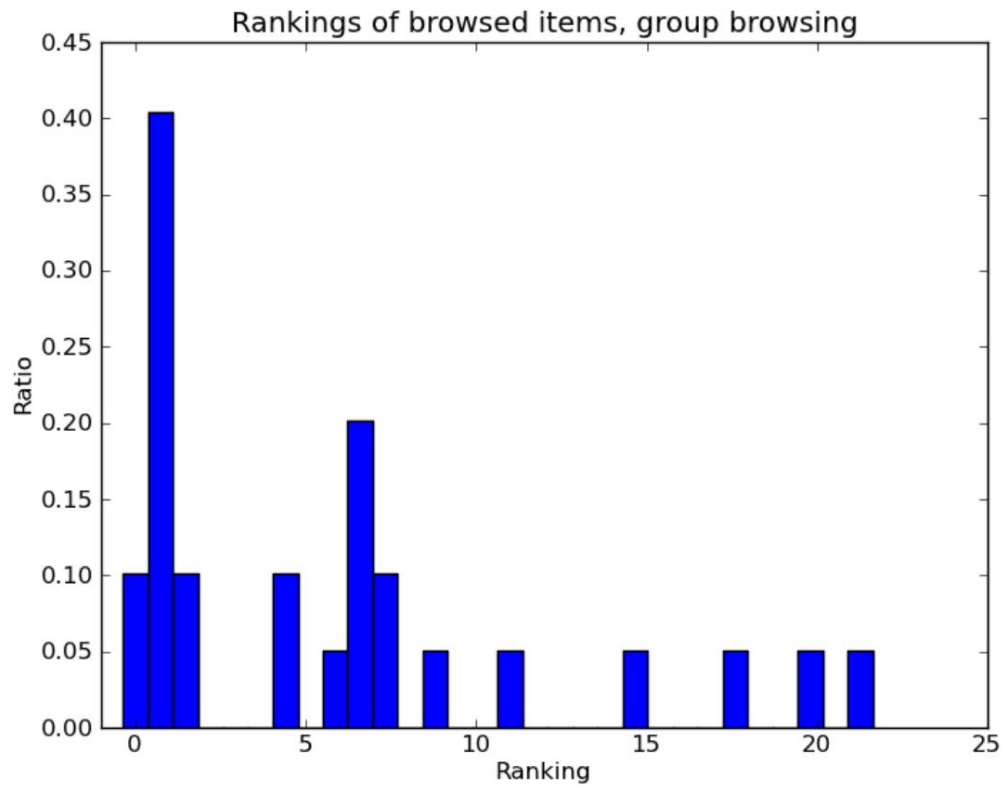


Figura 25: Ranking de los restaurantes consultados con personas añadidas en la lista de compañeros de almuerzo. En la base de datos se registró un total de 28 navegaciones grupales.

A partir de los diagramas, podemos ver que los restaurantes mejor clasificados son más propensos a la selección, pero aparte de eso, no se pueden hacer percepciones reales. Hubo un total de ocho personas que hicieron selecciones y solo cuatro personas que intentaron agregar compañeros de almuerzo al navegar. Esto significa que algunos usuarios no se molestaron en agregar compañeros de almuerzo mientras navegaban, eso se hizo después de que se había tomado una decisión, y el operador necesitaba informarle a un amigo sobre la selección. Sin embargo, no está claro si se pueden sacar tales conclusiones, ya que la gran mayoría de las búsquedas y selecciones grupales fueron realizadas por dos empleados de nuestro equipo en Ericsson, las otras personas probaron la funcionalidad una o dos veces como máximo.

## 8 Datos de usuario

Para entender la utilidad del sistema y el algoritmo de recomendación puede ser interesante analizar los datos de los usuarios de la aplicación. La Figura 26 representa el número de solicitudes de navegación que todos los usuarios han enviado al servidor a lo largo del tiempo. En la figura se ven claramente dos picos, el primero ocurrió a fines de mayo, y esto quizás se deba a la versión beta de la aplicación, que se envió al grupo de investigación por correo electrónico a mediados de mayo. La campaña de marketing y entrega de folletos se realizó a principios de junio, y sus resultados se pueden ver a mediados de junio, cuando aparece el próximo gran pico. Es evidente que hay un par de días de retraso entre los eventos y los picos, los usuarios tardan un tiempo en digerir el mensaje e instalar la aplicación.

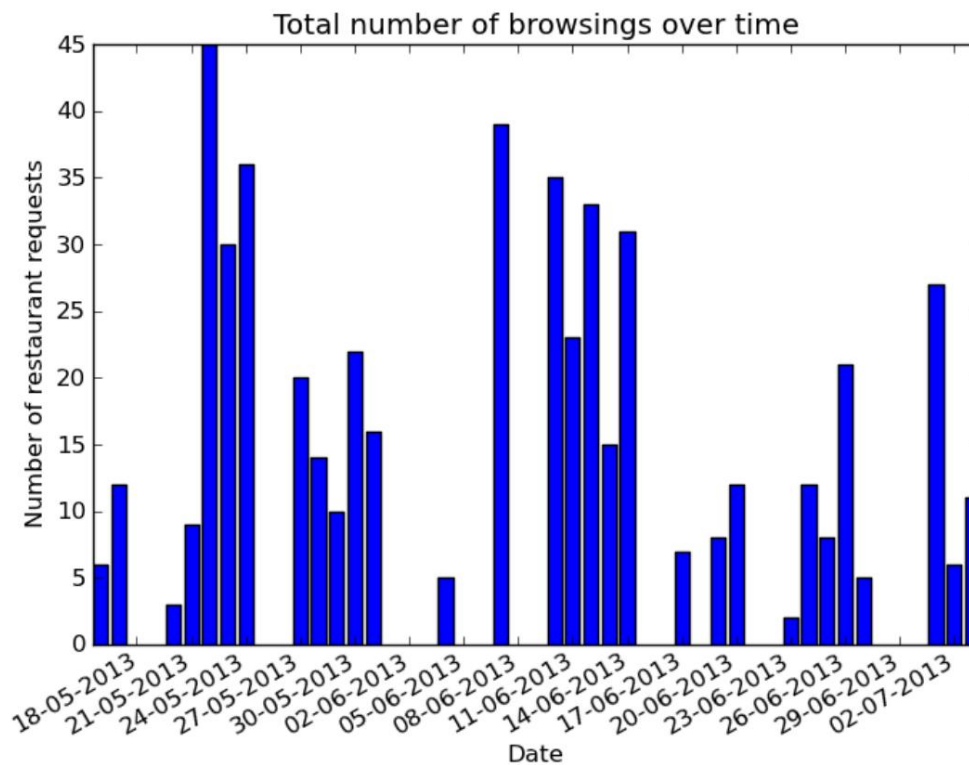


Figura 26: El número total de solicitudes de navegación enviadas al servidor a lo largo del tiempo.

La recepción pública de la aplicación no fue muy buena, y eso se puede ver en cómo se distribuyeron las navegaciones entre los usuarios. La figura 26 traza la vida útil total

búsquedas en restaurantes por parte de los usuarios, y muestra que la gran mayoría simplemente instaló la aplicación, hizo una navegación y nunca la volvió a usar. Solo dos usuarios, estos eran los dos empleados de Ericsson involucrados en el proyecto, realizaron bastantes búsquedas en restaurantes, y se pueden ver alrededor de 80 y 110 búsquedas en el diagrama.

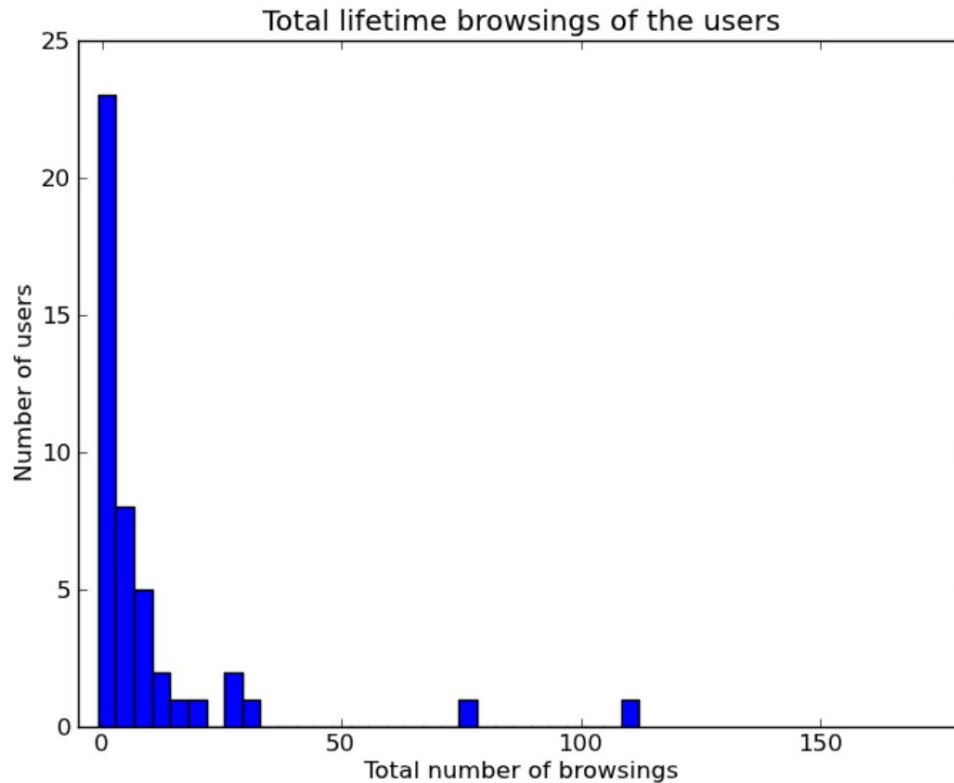


Figura 27: Distribución del número total de visitas a restaurantes de los usuarios.

También es interesante ver durante cuántos días los usuarios consumieron la aplicación. Si bien algunos usuarios realizaron relativamente muchas navegaciones, quizás solo usaron la aplicación en un día. Dado que el algoritmo de recomendación se entrena una vez al día, no habría recomendación para estos usuarios. La Figura 28 muestra cuántos días los usuarios usaron la aplicación. La mayoría de los usuarios solo usaron la aplicación después de instalarla y luego miraron uno o dos restaurantes. Los dos empleados de Ericsson implicados pueden volver a ser vistos como los usuarios más frecuentes.

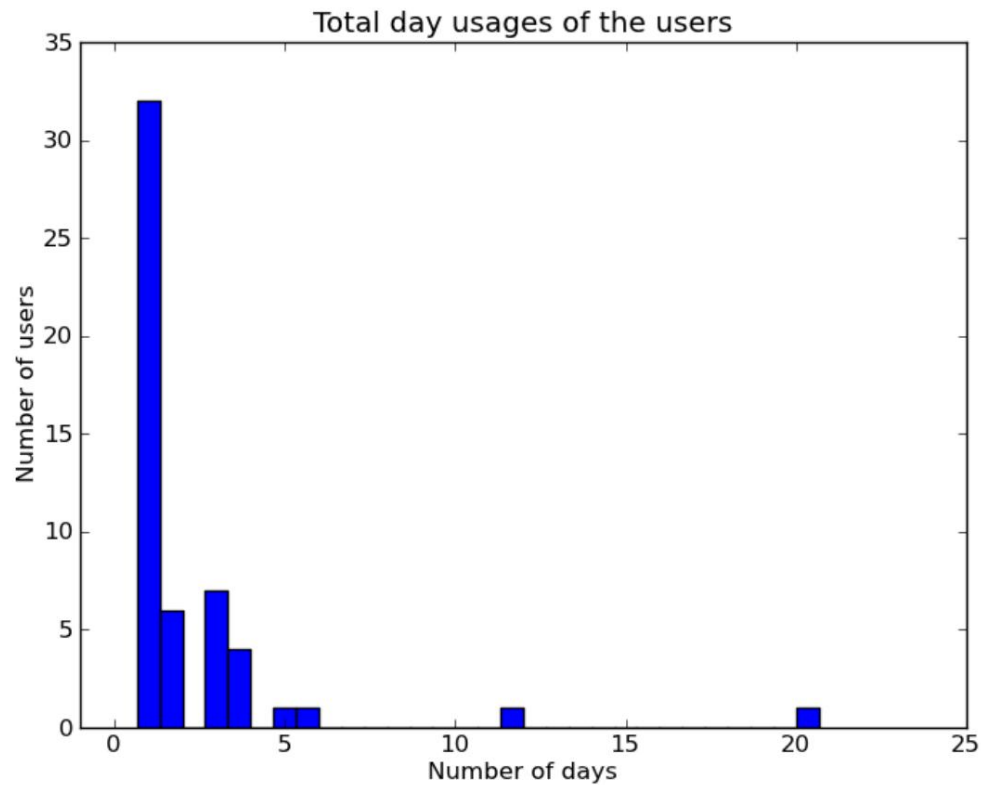


Figura 28: Distribución del total de usos diarios de los usuarios.

Como se explicó en secciones anteriores, el tiempo entre dos solicitudes de menú de un usuario puede interpretarse como el tiempo de navegación de un elemento. Se estableció un umbral de 60 segundos en todos los tiempos de navegación, y se guardaron para su posterior análisis. La distribución total de tiempos de navegación para todos los usuarios se muestra en la figura 29. Estos tiempos no están normalizados con la longitud del texto de los menús, que es la medida utilizada en el algoritmo de recomendación. Está claro que la probabilidad más alta para un tiempo de navegación en un restaurante, la moda, es de alrededor de diez segundos, que es menor que el valor esperado o la media, ya que la distribución de probabilidad es asimétrica y tiene un sesgo. La media se usa en el sistema de recomendación, por lo que esto puede causar pesos demasiado pequeños,  $w_2$  y  $v_3$ , capturando el tiempo de navegación.



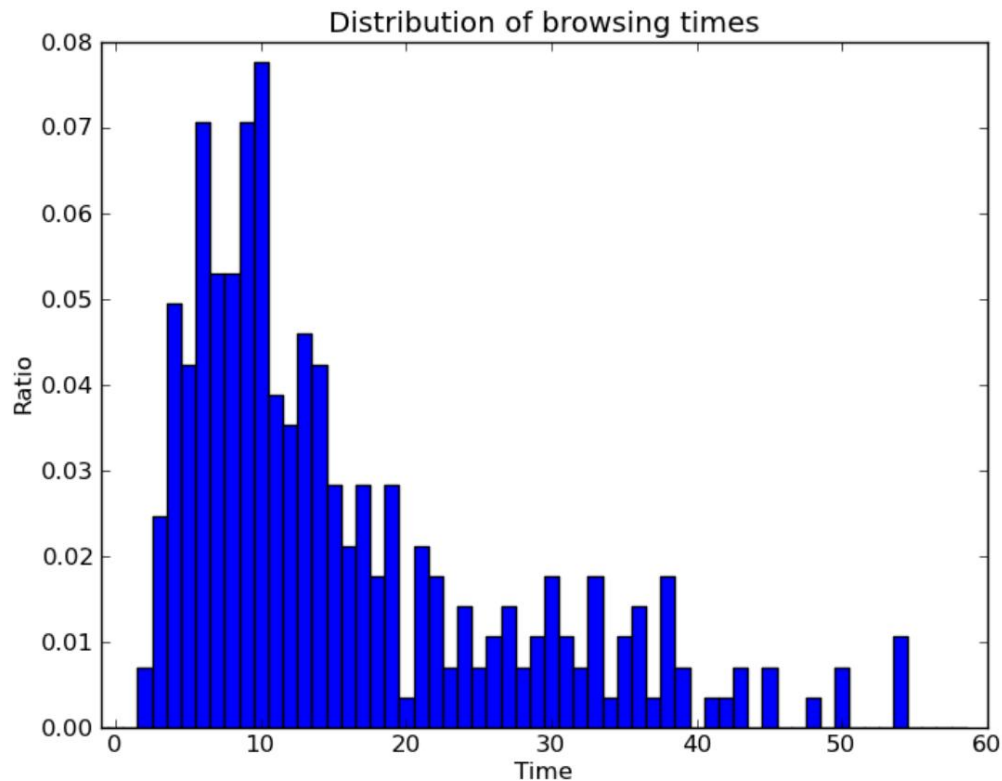


Figura 29:

## 9 Análisis textual de los menús

La funcionalidad basada en el contenido del sistema estima la preferencia de los usuarios por los menús en función del texto del menú. Como se explicó anteriormente, los descriptores de este texto fueron vectores de características tf-idf. Desde los primeros días de este proyecto, a fines de enero, el analizador se implementó por completo y comenzó a guardar los textos del menú en la base de datos. A veces, los menús no estaban escritos explícitamente en el sitio web, el cuerpo del texto solo contenía un enlace a la página de inicio del restaurante. Algunas cadenas de restaurantes o puestos de comida tenían el mismo menú todos los días, por lo que su vector tf-idf no cambiaba de un día para otro. Sería interesante visualizar los vectores de características tf-idf de los menús de los restaurantes, y tal vez se puedan sacar algunas conclusiones al comprender los datos textuales. El problema es que los vectores tienen 1000 dimensiones, lo que significa que es necesario emplear alguna técnica de reducción de dimensionalidad para visualizar los datos. El análisis de componentes principales, PCA, es un método estándar para hacer esto. El algoritmo transforma los ejes de coordenadas para que estén alineados en el

direcciones con la mayor variación en el conjunto de datos. Luego, los puntos de datos se proyectan en estos ejes, y solo se seleccionan para el trazado los dos ejes que corresponden a la varianza más grande. Los resultados de la visualización de los menús de los restaurantes se muestran en la figura 30. Para comparar restaurantes se calculan los vectores medios de los menús y se muestran en la figura 31.

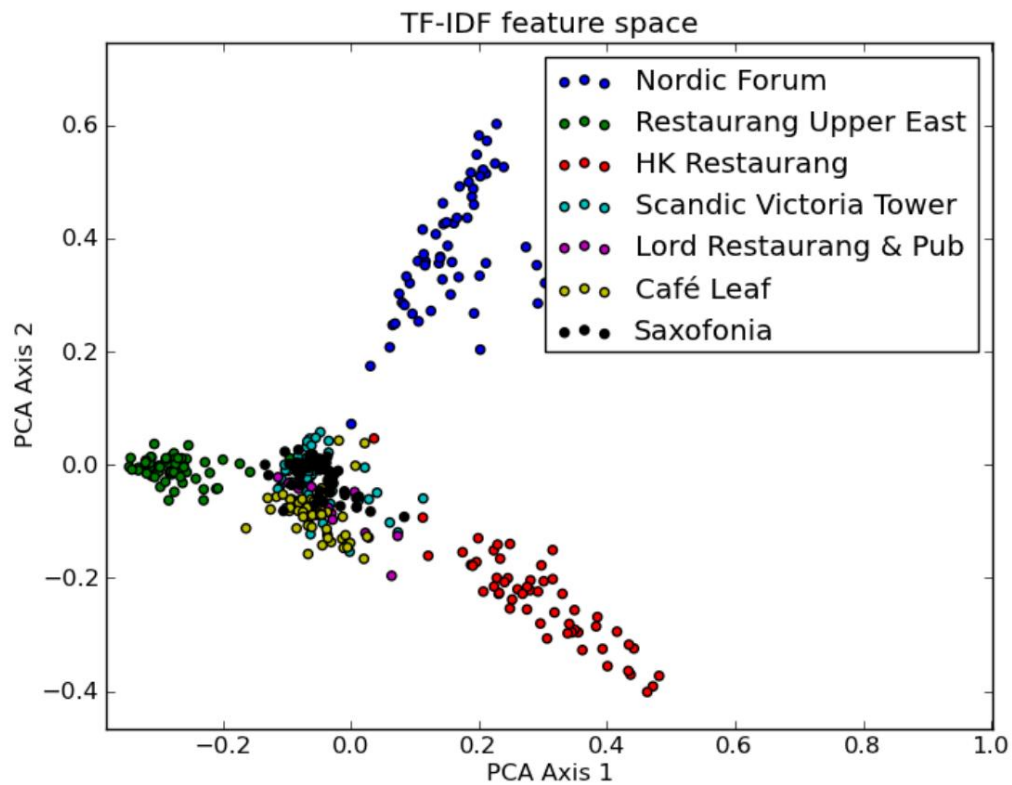


Figura 30: TF-IDF de menús, cada punto corresponde al vector tf-idf del cuerpo del texto del menú de un restaurante. Se realiza el análisis de componentes principales y los vectores se visualizan en 2D.

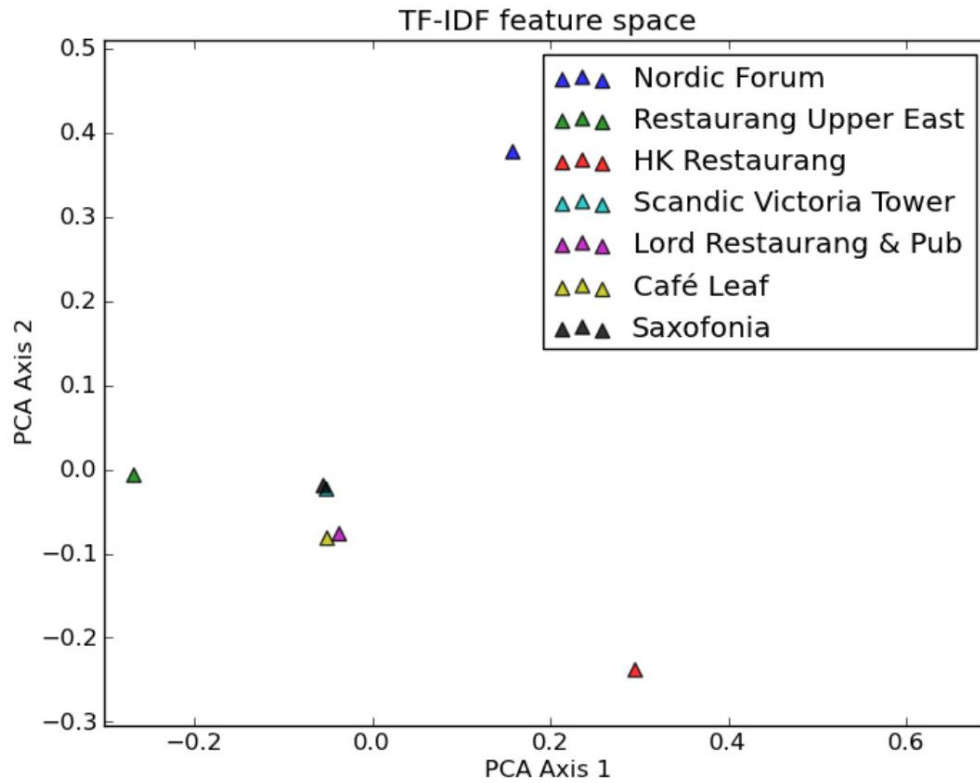


Figura 31: TF-IDF de los centroides del menú, cada punto corresponde a la media de los vectores tf-idf visualizados del cuerpo del texto del menú de un restaurante que se muestra en la figura 30

Antes de calcular los vectores tf-idf, se establecieron palabras vacías y un umbral para la aparición mínima de palabras en el corpus. Si una palabra solo aparecía una vez en todo el corpus, se descartaba al calcular los valores de tf-idf. Las palabras vacías seleccionadas, que no describían nada sobre el contenido de alimentos de los menús, se muestran en la tabla 9.

Otra cosa que se puede analizar es cómo se distribuye la ocurrencia de palabras en el corpus. La figura 9 muestra todas las palabras ordenadas por su frecuencia y su respectiva frecuencia total en todo el corpus del documento. Después de realizar el preprocesamiento, quedaron alrededor de 1200 palabras.

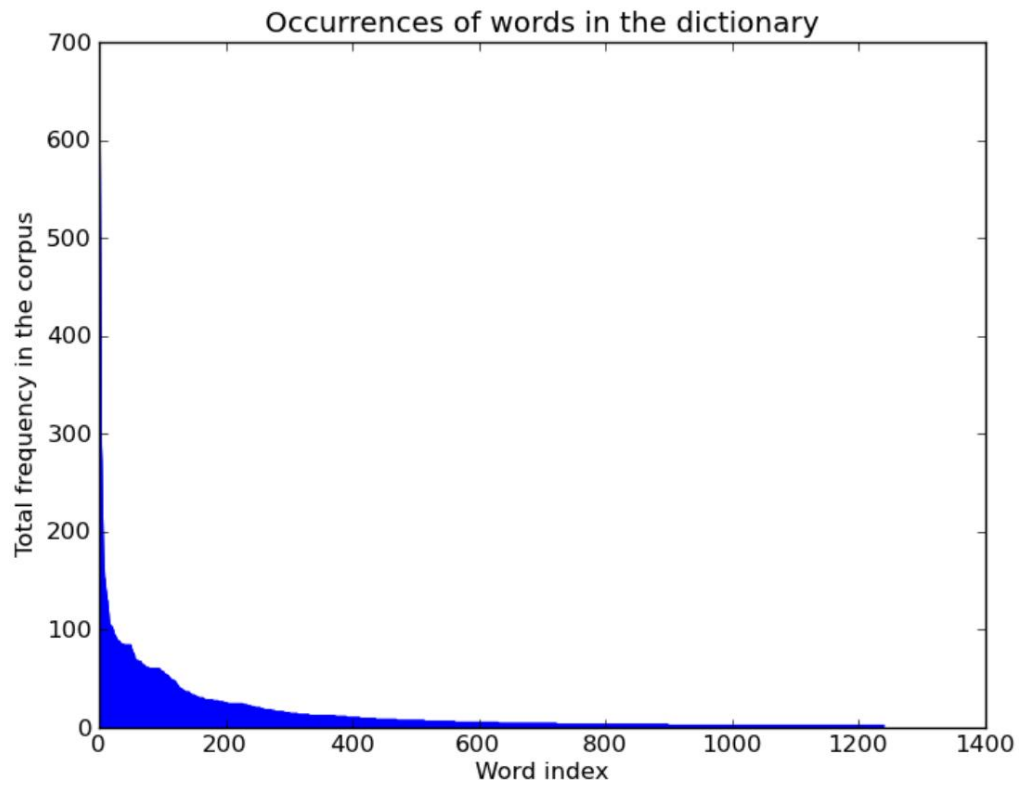


Figura 32: Frecuencia de todas las palabras en el corpus del documento después de eliminar las palabras vacías y establecer la frecuencia mínima del corpus.

Tabla 4: Frecuencia de palabras en el diccionario y selección de palabras vacías, mostradas en negrita rostro.

| Palabra     | Frecuencia de término en corpus | Traducción en inglés       |
|-------------|---------------------------------|----------------------------|
| med         | 3174                            | con                        |
| och         | 1700                            | y                          |
| dagens      | 887                             | de hoy                     |
| pÿa         | 863                             | sobre                      |
| muchos      | 768                             | menú                       |
| att         | 684                             | para                       |
| papas       | 650                             | patatas                    |
| para        | 607                             | por                        |
| clica       | 600                             | hacer clic                 |
| l'anken     | 600                             | Enlace                     |
| se          | 600                             | ver                        |
| Stekt       | 570                             | frito                      |
| Kokt        | 464                             | cocido                     |
| vendedor    | 374                             | o                          |
| Samt        | 363                             | y                          |
| risa        | 288                             | arroz                      |
| pasta       | 271                             | rollitos de primavera      |
| kyckling    | 219                             | pollo                      |
| vÿarrullar  | 195                             | pasta                      |
| AV          | 169                             | de                         |
| nem         | 168                             | nem                        |
| vegetarisco | 154                             | vegetariano                |
| servidoras  | 151                             | servido                    |
| agg         | 142                             | huevos                     |
| lingon      | 138                             | arándano rojo              |
| potatismos  | 131                             | puré de patatas            |
| tomatsÿas   | 129                             | salsa de tomate            |
| es          | 127                             | uno                        |
| persistir   | 121                             | persa                      |
| kr          | 120                             | coronas (moneda en Suecia) |
| algo        | 120                             | cual                       |
| Arkansas    | 115                             | es                         |
| hel         | 113                             | todo                       |
| flask       | 106                             | Cerdo                      |
| gradde      | 105                             | crema                      |
| sallad      | 104                             | ensalada                   |
| Panerad     | 103                             | panera                     |

## Parte IV

# Discusión y trabajo futuro

## 10 Discusión

Como se puede ver en las cifras presentadas, es difícil ver una mejora obvia con el tiempo. La mayoría de las búsquedas se concentran en la parte superior de la lista, pero la concentración relativa parece cambiar solo ligeramente durante el tiempo que estuvo en uso el sistema de recomendación. Partiendo de la premisa de que se ha desarrollado un sistema de recomendación efectivo, ¿es válida la suposición de que la mayoría de las búsquedas se concentrarán en la parte superior de la lista?

El autor de esta tesis argumentaría que así es. Un usuario primero consulta sus restaurantes favoritos y, si el contenido no es lo suficientemente satisfactorio, continúa navegando. Los usuarios confían en que la parte de contenido del sistema de recomendación encontrará platos adecuados y no quieren gastar el esfuerzo de desplazarse hacia abajo, o quieren ver sus restaurantes favoritos todos los días, independientemente del menú. En cualquier caso, dará lugar a una ponderación tal que los restaurantes relevantes terminarán en la parte superior de la lista.

Es de esperar que la parte de contenido del sistema haya seleccionado los platos correctamente, para que el usuario esté satisfecho al ver el contenido. Aunque estén satisfechos con los primeros restaurantes y menús, es posible que algunos usuarios deseen explorar varios otros restaurantes de la lista. Si el sistema de recomendación funciona como se espera, el contenido de estos será menos adecuado para el usuario y, finalmente, la persona aprenderá que solo los mejores restaurantes son relevantes.

Un tema que aún no se ha discutido es que los usuarios pueden cansarse tanto de los platos como de los restaurantes. Claramente es una falla del sistema si al usuario se le recomienda la misma comida todos los días, a pesar de que podría ser un plato favorito. Sin embargo, es poco probable que esto suceda en el sistema, ya que cada menú contiene varios platos y los ingredientes se pueden encontrar en muchas constelaciones. El vector de preferencia de alimentos tendrá valores positivos para muchas palabras clave y dirá qué ingredientes no gustan en lugar de otorgar una puntuación alta a un plato en particular. Por otro lado, puede suceder que el mismo conjunto de restaurantes termine siempre en la parte superior de la lista, de todos modos esto no es necesariamente un problema. Un usuario puede tener un restaurante favorito para almorzar, donde el menú cambia todos los días, y esto le dará suficiente variación. También se debe tener en cuenta que los restaurantes disponibles en el sitio web eran de carácter muy diferente. Todo, desde puestos de comida, restaurantes de comida rápida y cafeterías hasta restaurantes a la carta para almorzar, apareció en el sitio. Además, algunos restaurantes estaban demasiado lejos y no eran relevantes para el usuario. Esto sería evidente después de un tiempo, ya que estos restaurantes no recibirían ninguna solicitud de menú.

Al observar los datos de los usuarios, es evidente que los usuarios realmente no utilizaron la aplicación. Simplemente lo instalaron, hicieron algunas búsquedas y nunca lo volvieron a tocar. Esto se puede ver claramente en las figuras 27 y 28. Debido a esto, los resultados de la

experimento no son concluyentes, es imposible evaluar el rendimiento del sistema de recomendación. Al estudiar las solicitudes totales de navegación al sistema en la figura 26, no queda claro que disminuyan, pero esto se debe a la demora desde que el usuario ve el folleto hasta la instalación de la aplicación. Los nuevos usuarios ingresaron continuamente durante el transcurso del experimento, pero ninguno de ellos usó la aplicación de manera constante. Los dos picos que se aprecian en la figura 26 se deben a las campañas de marketing.

Al examinar la figura 20, parece que el sistema funcionó muy bien, incluso al principio, cuando no se habían recopilado calificaciones ni se habían suministrado datos al algoritmo de entrenamiento, y esto es ciertamente una contradicción. La mayoría de los folletos se entregaban a los empleados que trabajaban en el área y, por lo tanto, a menudo almorzaban en uno de los restaurantes, por lo que se podría argumentar que ya tenían opiniones sobre los diferentes lugares. Aun así, solo solicitaron ver los menús de los restaurantes en la parte superior de la lista y no se molestaron en encontrar su favorito. Si no se recopilaban calificaciones, la lista estaba simplemente en orden alfabético. Este comportamiento de navegación muestra que los usuarios realmente no intentaron usar la aplicación de acuerdo con su propósito, simplemente la probaron, señalando solo los primeros elementos. El valor que obtuvieron al conocer los menús de los restaurantes no fue sustancial y quizás no fue clave para la decisión.

Puede haber un defecto con la recomendación de platos, que se basa en el análisis textual de los menús. Como se ve en la imagen 30, algunos de los restaurantes claramente forman grupos, incluso después de que los vectores tf-idf se hayan transformado de dimensiones muy altas a solo dos dimensiones. Debido a la maldición de la dimensionalidad, es posible que estos grupos estén aún más separados en dimensiones superiores. La similitud entre los menús de los mismos restaurantes puede deberse a que sirven categorías especiales de cocina, por ejemplo, china, india, italiana, etc., y por lo tanto utilizan ingredientes especiales. Sin embargo, este no es el caso de los restaurantes de la figura. Puede ser simplemente que diferentes personas escriban los menús de los diferentes restaurantes y tengan su propia forma de expresar los platos. Este fenómeno puede llevar a que la lista de recomendaciones individuales se convierta en una lista top ordenada únicamente según el número de visitas en cada restaurante. Cuando un usuario mira el menú de un restaurante, el vector de preferencia de alimentos tf-idf aumentará en consecuencia, y al día siguiente el menú del mismo restaurante obtendrá la mayor similitud. Otro problema se puede ver en la figura 9, donde es evidente que la mayoría de las palabras solo aparecen unas pocas veces en los menús. En total, se recopilaron más de 4000 menús, y las palabras más frecuentes solo se escribieron un par de cientos de veces, lo que significa que rara vez se producirán coincidencias relevantes entre los vectores tf-idf del menú. Los vectores serán escasos y las similitudes muy pequeñas, y esto se debe a que el vocabulario de los menús es muy específico y la longitud del texto suele ser muy corta. Hay muchos platos que tienen nombres divertidos, y el chef puede querer usar un vocabulario especial solo para hacer que el plato suene especial y exótico. Quizás dos menús tendrán similitudes solo porque ambos contienen "papas" o "arroz", y esto no necesariamente dice nada sobre el gusto de los usuarios por

El menú.

El autor de esta tesis es muy escéptico sobre el supuesto de navegación grupal descrito en la sección 6.2 en la parte de metodología. Puede ser válido en otros entornos, donde el grupo tiene que tomar una decisión informada y la información solo se proporciona en un dispositivo. Pero en realidad, solo unas pocas personas necesitan ver el menú antes de ir a un restaurante con colegas, y dado que la mayoría usa teléfonos inteligentes hoy en día, pueden obtener la información desde su propio dispositivo. Dado que tan pocas personas utilizaron la recomendación grupal, es imposible evaluar su desempeño. Hay otras aplicaciones y páginas web móviles donde el usuario puede ver múltiples menús de diferentes restaurantes en una sola página y buscar platos adecuados de manera más eficiente. Además, dado que los restaurantes cercanos considerados por el grupo no eran tantos, mirar sus menús y tomar una decisión a dónde ir no es un gran problema para los usuarios. La decisión final siempre será del grupo, tener un algoritmo ordenando los restaurantes quizás no sea tan útil. Los usuarios discutirán las desventajas, comprometerán e incorporarán conocimientos en la decisión que el algoritmo no conoce.

Otra posible razón de la mala recepción de la aplicación es que tenía algunos errores y no estaba muy bien diseñada. Muchos usuarios tuvieron que reiniciar la aplicación, a veces funcionaba ya veces no. El desarrollo se retrasó mucho, debido a otras asignaciones y falta de tiempo, se tuvo que cambiar el desarrollador principal durante el transcurso de la tesis. El equipo de desarrollo no pudo encontrar los errores y tardaron en responder a las correcciones. A menudo tomaba semanas antes de que el equipo hiciera las correcciones requeridas.

## 11 Trabajo futuro

Sería interesante ver el rendimiento real del sistema de recomendación de grupos, pero esto requiere que los usuarios proporcionen información explícita a donde fue el grupo. Quizás la tecnología GPS podría usarse para esto. Otra cosa que se podría hacer es presentar la recomendación de una manera más divertida, mostrando el nombre del restaurante y diciendo al grupo el motivo de la recomendación. Se podrían utilizar otros conocimientos como el clima, el día de la semana y la ubicación de los restaurantes. Luego, el sistema de recomendación puede mostrar información sobre los restaurantes a través de una interfaz y, por ejemplo, indicar que "a las personas A, B y C les gusta este plato en particular en el restaurante, la ubicación está cerca y hoy hace mal tiempo". De esa manera, es más probable que la recomendación sea aceptada por el grupo. Si uno intenta hacer que la aplicación sea divertida y útil, tal vez se deba recopilar cierta información explícitamente. El usuario podría, por ejemplo, decir una serie de platos o ingredientes que le gustan y que no le gustan. Ahora solo podemos calcular la preferencia por menús completos basados en calificaciones implícitas, y esto obviamente no es muy preciso. Además, la entrada al sistema es la preferencia después de leer una descripción, sería mucho mejor que el usuario califique el artículo después de que el consumo real haya



lugar tomado. Finalmente, los sinónimos podrían usarse de una mejor manera en el sistema, ya que la mayoría de las palabras aparecen con poca frecuencia en el diccionario.

Para la recopilación de calificaciones implícitas, es importante que los usuarios naveguen a través de los restaurantes, pero esto no es útil en absoluto para los usuarios. Sería mucho mejor mostrar los restaurantes y los menús en la misma página. Se podrían realizar experimentos con seguimiento ocular para la recopilación de calificaciones implícitas utilizando la cámara frontal de los teléfonos inteligentes.

Dado que no se ha utilizado un enfoque colaborativo, no es necesario tener el sistema como una arquitectura de servidor cliente. Los usuarios pueden buscar restaurantes en su propio dispositivo, mientras la aplicación recopila las preferencias. La recomendación de grupo podría mostrarse en todos los dispositivos simultáneamente haciendo que los usuarios enciendan el bluetooth y agiten los teléfonos. Las preferencias de los amigos se transmitirían automáticamente.

## Parte V

# Conclusiones

Los sistemas de recomendación son útiles cuando se tienen miles de artículos y el usuario no puede considerarlos todos. El sistema desarrollado en esta tesis está ordenando solo unos pocos restaurantes en una lista y, en el mejor de los casos, el valor que los usuarios obtendrán de la aplicación es que ahorran un poco de tiempo, no tienen que desplazarse o navegar. El sistema de recomendación de grupo no es un sistema de recomendación normal en el sentido de que está tratando de estimar calificaciones de artículos, está tratando de estimar una selección para un grupo. Si solo hay unos pocos elementos disponibles para seleccionar, no es una tarea engorrosa para el grupo tomar una decisión, especialmente en este caso cuando los usuarios están familiarizados con la mayoría de los elementos. El valor que puede dar un sistema de recomendación de grupo en este caso es proporcionar información contextual, contar las características de los elementos y buscar en los elementos contenido interesante y no interesante y, por lo tanto, ayudar al grupo a tomar una decisión informada. Hacer una aplicación de este tipo es una tarea programática, y quizás el aprendizaje automático no necesite ser parte de ella. La decisión final siempre la tomarán los individuos del grupo, y el objetivo siempre debe ser la satisfacción del usuario.

## Referencias

- [1] JF McCarthy, "Buscador de restaurantes de bolsillo: un sistema de recomendación situado para grupos", Actas del taller sobre comunicación móvil ad-hoc en la Conferencia ACM de 2002 sobre factores humanos en sistemas informáticos, 2002.

- [2] G. Adomavicius y A. Tuzhilin, "Hacia la próxima generación de sistemas de recomendación: un estudio del estado del arte y posibles extensiones", *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, núm. 6, págs. 734–749, junio de 2005.
- [3] D. Jannach, M. Zanker, A. Felfernig y G. Friedrich, *Sistemas de recomendación: un Introducción*. Prensa de la Universidad de Cambridge, septiembre de 2010.
- [4] P. Melville y V. Sindhvani, "Sistemas de recomendación", *Encyclopedia of Machine Aprendizaje*, 2010.
- [5] MJ Pazzani y D. Billsus, "Sistemas de recomendación basados en contenido", *The Adaptive Web*, págs. 325–341, 2007.
- [6] K. Takano y KF Li, "Un recomendador personalizado adaptable basado en el aprendizaje del comportamiento de navegación web", *IEEE Computer Society*, pp. 654–660, 2009.
- [7] MS Pera y Y.-K. Ng, "Un recomendador grupal de películas basado en la similitud y popularidad del contenido", *Information Processing and Management: an International Journal*, vol. 49, págs. 673–687, mayo de 2013.
- [8] M. Balabanovic y Y. Shoham, "Recomendación colaborativa basada en contenido," *Comunicaciones de la ACM*, vol. 40, núm. 3, 1997.
- [9] Y.-L. Chen, L.-C. Cheng y C.-N. Chuang, "Un sistema de recomendación grupal con consideración de las interacciones entre los miembros del grupo", *Expert Systems with Applications*, no. 34, págs. 2082–2090, 2008.
- [10] S. Amer-Yahia, SB Roy, A. Chawla, G. Das y C. Yu, "Recomendación del grupo: semántica y eficiencia", *Actas de la dotación de una base de datos muy grande*, vol. 2, págs. 754–756, agosto de 2009.
- [11] L. Boratto y S. Carta, "Estado del arte en la recomendación de grupos y nuevos enfoques para la identificación automática de grupos", *Recuperación de información y minería en entornos distribuidos*, págs. 1 a 20, 2011.
- [12] E. Ntoutsi, K. Stefanidis y K. Norvag, "Recomendaciones de grupos rápidos mediante la aplicación de la agrupación de usuarios", *Actas de la 31.ª conferencia internacional sobre modelado conceptual*, págs. 126–140, octubre de 2012.
- [13] Y. Koren, R. Bell y C. Volinsky, "Técnicas de factorización de matrices para sistemas de recomendación", *IEEE Computer Society*, 2009.
- [14] R. Hastie y T. Kameda, "La robusta belleza de las reglas de la mayoría en las decisiones grupales", *Revisión psicológica*, vol. 112, núm. 2, págs. 494,508, 2005.

- [15] A. Jameson y B. Smyth, "Recomendación para grupos", *The Adaptive Web*, págs. 596–627, 2007.
- [16] M. O'Connor, D. Cosley, JA Konstan y J. Riedl, "PolyLens: un sistema de recomendación para grupos de usuarios", *ECSCW'01 Actas de la séptima conferencia sobre la Conferencia europea sobre trabajo cooperativo asistido por computadora*, págs. 199–218, 2001.
- [17] Z. Yu, X. Zhou y JG Yanbin Hao, "Recomendación de programas de televisión para múltiples espectadores basada en la combinación de perfiles de usuario", *Modelado de usuarios e interacción adaptada al usuario*, vol. 16, núm. 1, págs. 63–82, marzo de 2006.
- [18] S. Berkovsky y J. Freyne, "Recomendaciones de recetas basadas en grupos: Análisis de estrategias de agregación de datos", *Actas de RecSys'10 de la cuarta conferencia ACM sobre sistemas de recomendación*, págs. 111–118, septiembre de 2010.
- [19] L. Baltrunas, T. Makcinskas y F. Ricci, "Recomendación grupal con agregación de rango y filtrado colaborativo", *RecSys'10 Actas de la cuarta conferencia ACM sobre sistemas de recomendación*, págs. 119–126, septiembre de 2010 .
- [20] J. Masthoff, "Modelado grupal: selección de una secuencia de elementos de televisión para adaptarse a un grupo de espectadores", *Modelado de usuario e interacción adaptada al usuario*, vol. 14, págs. 37 a 85, 2004.
- [21] M. Claypool, P. Le, M. Wased y D. Brown, "Indicadores de interés implícito", en *Actas de la sexta conferencia internacional sobre interfaces de usuario inteligentes*, 2001, págs. 33–40.
- [22] D. He y D. Wu, "Hacia una fusión de datos robusta para la recuperación de documentos", en *la Conferencia internacional IEEE sobre procesamiento del lenguaje natural e ingeniería del conocimiento*, 2008.
- [23] JH Lee, "Análisis de combinación de evidencia múltiple", *Actas de SIGIR '97 de la vigésima conferencia internacional anual ACM SIGIR sobre investigación y desarrollo en recuperación de información*, págs. 267–276, 1997.
- [24] JC Platt, *Salidas probabilísticas para máquinas de vectores de soporte y comparaciones con Métodos de verosimilitud regularizada*. Prensa del MIT, 1999.
- [25] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blon del, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot y E. Duchesnay, "Scikit-learn: Aprendizaje automático en Python" *Revista de investigación de aprendizaje automático*, vol. 12, págs. 2825–2830, 2011.

