



Universidad del Bío-Bío, Chile
Facultad de Ciencias Empresariales
Departamento de Sistemas de Información

Sistema de recomendación de recetas saludables, basado en la preferencia y necesidad nutricional del usuario

Proyecto de título presentado por Boris Francisco Rioseco Elgueta de
la Carrera Ingeniería Civil Informática
Dirigida por Pedro Gerónimo Campos Soto

2020

"Si piensas que los usuarios de tus programas son idiotas, sólo los idiotas usarán tus programas"

Linus Torvalds - «padre» de Linux

Agradecimientos

Porque estoy convencido de que es difícil titularse de manera solitaria, a todas las personas que nombre en este aparatado les estaré eternamente agradecido por su apoyo, confianza y sobre todo preocupación por verme cumplir el objetivo de lograr la titulación.

Quiero partir por nombrar a mi núcleo más cercano, mi familia, por todos los consejos, por el apoyo mientras asistía a clases, ya sea con la alimentación, dinero para el transporte y todos los gastos que se requieren para llevar la vida de estudiante universitario, así como también el apoyo moral y por sobre todo el amor entregado hasta el día de hoy.

También quiero nombrar a mi polola Vianka, que sin sus consejos, apoyo y amor se me hubiera hecho más difícil lograr llegar a estas instancias.

También a mi tío Eduardo, por darme las posibilidades de trabajo y solventar algunos gastos necesarios durante la universidad, como ropa, computador, entre otros.

El desarrollo de este Proyecto de Título fue apoyado por los proyectos DIUBB 2030248 IF/R y 194810 GI/VC.

Resumen

El título del proyecto es: “Sistema de recomendación de recetas saludables, basado en la preferencia y necesidad nutricional del usuario”.

El proyecto se desarrolla en el ámbito del proceso de titulación de la carrera de Ingeniería Civil en Informática de la Universidad del Bío-Bío de Chile, para dar consentimiento de la aprobación de los requisitos exigidos por la universidad.

La motivación para realizar este proyecto, es implementar un algoritmo de recomendación para alumnos universitarios, con el fin de poder guiarlos a una vida más saludable, recomendando recetas con un contenido nutricional beneficioso para la salud, tomando en cuenta sus gustos y preferencias.

El proyecto se lleva a cabo mediante 2 procesos, el primer proceso es la literatura e investigación de algoritmos de recomendación y posterior elección del algoritmo a utilizar, y el segundo proceso es la implementación del algoritmo seleccionado, en una aplicación web, aplicación que permitirá obtener resultados de recomendación y además resultados experimentales.

Mediante la implementación del algoritmo en la aplicación creada por el alumno, se pudieron obtener los resultados esperados, es decir, la aplicación entrega recomendaciones a los usuarios, así mismo, permite la recopilación de datos experimentales.

Palabras clave: Sistemas de recomendación · Recetas saludables · Preferencias del usuario · Necesidades nutricionales · Comportamiento alimenticio · Algoritmos · Relevancia

Abstract

The title of the project is: “Healthy recipe recommendation system, based on the user's preference and nutritional need”

The project is developed in the field of the process of degreing the degree of Civil Engineering in Computer Science from the University of Bio-Bio of Chile, to give consent to the approval of the requirements required by the university.

The motivation to carry out this project, is to implement a recommendation algorithm for university students, in order to be able to guide them to a healthier life, recommending recipes with a nutritional content beneficial to health, taking into account their tastes and preferences.

The project is carried out through 2 processes, the first process is the literature and research of algorithms of recommendation and subsequent choice of the algorithm to be used, and the second process is the implementation of the selected algorithm, in a web application, application that will allow to obtain recommendations results and also experimental results.

By implementing the algorithm in the student created application, the expected results could be obtained, i.e., the application delivers recommendations to users, as well as allows the collection of experimental data

Keywords: Recommendation systems · Healthy recipes · User preferences
· Nutritional needs · Eating habits and behaviour · Algorithm · Relevance

Índice de contenido

Índice de contenido	6
Índice de tablas	7
Índice de imágenes	8
Índice de figuras	8
1 Capítulo 1: Introducción	9
1.1 Objetivos	10
1.1.1 Objetivo General	10
1.1.2 Objetivos Específicos	10
1.1.3 Justificación del Proyecto	11
1.1.4 Análisis de los Principales Trabajos Realizados en el área o tema de la propuesta	12
1.1.5 Requisitos funcionales y limitaciones	13
1.1.5.1 Funcionales	13
1.1.5.2 Limitaciones	13
1.2 Factibilidad del Proyecto	14
1.2.1 Factibilidad Técnica	14
1.2.2 Factibilidad Operativa	14
1.2.3 Factibilidad Económica	15
2 Capítulo 2: Marco Teórico (Conceptos)	16
2.1 Sistemas de Recomendación	16
2.1.1 Sistemas de recomendación basados en filtrado colaborativo	16
2.1.1.1 Recomendación basados en vecinos cercanos	18
2.1.2 Sistemas de recomendación basados en contenido	19
2.1.3 Sistemas de recomendación basados en conocimiento	20
2.1.4 Sistemas de recomendación híbridos	21
2.1.5 Recomendaciones individuales y grupales	22
2.1.6 Sistemas de recomendación de alimentos	23
2.1.6.1 Tipo 1	25
2.1.6.2 Tipo 2	35
2.1.6.3 Tipo 3	39
2.2 Capítulo 3: Estado del Arte	42
3 Capítulo 4: Algoritmos investigados	44
3.1 Descripción de algoritmos.	45
4 Capítulo 5: Implementación algoritmo	48
4.1 Descripción de implementación y estructura del algoritmo implementado	48

5	Capítulo 6: Aplicación web para ejecutar algoritmos de recomendación -----	52
5.1	Objetivo de la aplicación-----	52
5.2	Descripción de la interfaz y casos de uso-----	52
5.3	Evidencias de aplicación implementada -----	55
6	Capítulo 7: Experimentación y resultados -----	59
6.1	Descripción de Datasets -----	59
6.2	Métricas para evaluar resultados -----	60
6.3	Metodologías de evaluación -----	61
6.3.1	Metodología Traintest-set split -----	61
6.3.2	Metodología K-Fold Cross-Validation -----	62
6.4	Algoritmos usados para etapa experimental-----	64
6.5	Protocolos de prueba-----	65
6.5.1	Objetivo general -----	65
6.5.2	Condiciones de prueba-----	65
6.5.3	Procedimiento de la etapa experimental-----	66
6.6	Datos obtenidos-----	67
6.7	Análisis de resultados-----	68
7	Conclusión-----	69
8	Referencias -----	70

Índice de tablas

Tabla 1:	Evaluación simple de usuarios en una escala de 1 a 5.....	17
Tabla 2:	Base de Datos de Comidas	28
Tabla 3:	Evaluación de Resultados	28
Tabla 4:	Una lista de menús disponibles con la información correspondiente.....	37
Tabla 5:	Lista de menús recomendados para el usuario ua	38
Tabla 6:	Una lista de menús recomendados al usuario ua sobre la base de considerar su ingrediente favorito (es decir, tomate) y sus necesidades nutricionales	41
Tabla 7:	Descripción de la interfaz	53
Tabla 8:	Primera experimentación, ingrediente lettuce con calorías máximas de 300 Kcal	67
Tabla 9:	Segunda experimentación, ingrediente onion con calorías máximas de 600 Kcal.....	67
Tabla 10:	Tercera experimentación, ingrediente Cheese con calorías máximas de 900 Kcal.....	67
Tabla 11:	Ultima experimentación, ingrediente Chocolate con calorías máximas de 1200 Kcal	67

Índice de imágenes

Imagen 1: Predicción del valor de calificación para una receta objetivo utilizando un algoritmo CB propuesto por (Freyne & Berkovsky, 2010).	30
Imagen 2: Un ejemplo del gráfico de recetas para la planificación del menú "Tomate, harina, albahaca" (ingredientes que se muestran con líneas negras) son ingredientes de consulta. El plan de menú recomendado es un conjunto de recetas. {"Mozzarella, tomate y ensalada de albahaca", lasaña", "Pan italiano"} (nodos con marcos negros) cuyo costo total del menú es mínimo (Kuo et al., 2012).	34
Imagen 3: Valores de referencia para la ingesta nutricional. Bonn 2. Edición, 1. Volumen (2015) publicado por la Asociación Alemana de Nutrición, la Asociación Austriaca de Nutrición y la Asociación Suiza de Nutrición.....	36
Imagen 4: Lectura y unión de dos conjuntos de datos	49
Imagen 5: Código para generar una lista de predicción.....	50
Imagen 6: Lista con id de recetas a recomendar	50
Imagen 7: Casos de uso.....	54
imagen 8: Pantalla de inicio.....	55
imagen 9: Características de Surprise	55
imagen 10: Formulario de recomendación de recetas saludables	56
imagen 11: Formulario para experimentos	56
imagen 12: Ejemplo de lista de recetas recomendadas	57
imagen 13: Detalles por receta.....	57
imagen 14: Ejemplo de experimento usando el algoritmo SVDpp.....	58
Imagen 15: Train test set split.....	61

Índice de figuras

Ilustración 1: Arquitectura de software de aplicación desarrollada.....	52
Ilustración 3: Función train test set split.....	61
Ilustración 2: Validación cruzada de k iteraciones	62

1 Capítulo 1: Introducción

Una ingesta dietética adecuada se considera un factor importante para mejorar el bienestar general. Aunque la mayoría de las personas son conscientes de la importancia de los hábitos alimenticios saludables, por lo general tienden a descuidar los comportamientos adecuados debido a estilos de vida ocupados y/o falta de voluntad para dedicar esfuerzos a la preparación de alimentos. Estos problemas impiden que los usuarios consuman alimentos saludables (Van Pinxteren et al., 2011).

Por otro lado, la cantidad de personas que utilizan sitios de recetas está aumentando. Esta tendencia se debe especialmente al uso cada vez mayor de teléfonos móviles con acceso a Internet (Ueta et al., 2011b).

Por lo tanto, los sistemas de recomendación se investigan como una solución eficaz para ayudar a los usuarios a cambiar su comportamiento alimenticio y apuntar a opciones alimentarias más saludables.

En este informe de proyecto de título, se plantea una propuesta de implementación de un sistema de recomendación de recetas, usando algoritmos de recomendación ya existentes.

El proyecto de título, además, incluye una revisión de trabajos relacionados con los sistemas de recomendación de alimentos y/o recetas que brindan recomendaciones sobre la base de considerar las preferencias de los usuarios, así como sus necesidades nutricionales.

1.1 Objetivos

1.1.1 Objetivo General

Implementar un Sistema de Recomendación de recetas saludables, que, de acuerdo al perfil de cada usuario, sugiera recetas que prioricen las que tengan el contenido nutricional más saludable, y que tome en cuenta los gustos individuales del usuario.

1.1.2 Objetivos Específicos

1. Revisar literatura de Sistemas de Recomendación en general y Sistemas de Recomendación de recetas saludables.
2. Recopilar información de recetas y sus componentes nutricionales, así como valoración de usuarios y comentarios.
3. Implementar un algoritmo de recomendación de recetas que tome en cuenta preferencias de usuarios y contenido nutricional de la receta, para priorizar aquellas de contenido nutricional más saludable. El algoritmo a implementar se deberá basar en algoritmos descritos en la literatura del área.
4. Desarrollar una aplicación Web sencilla para recopilar perfiles de usuario, generar y visualizar recomendaciones, que permita la ejecución de experimentos con el algoritmo implementado, y que permita a futuro incorporar otras fuentes de información y algoritmos de recomendación.

1.1.3 Justificación del Proyecto

A través de los años, los estilos de vida van cambiando, así como también los tipos de alimentos y composición nutricional de dichas comidas.

En algunos casos el ritmo de vida acelerado lleva al ser humano a adaptarse a contar con muy poco tiempo libre, descuidando sus hábitos alimenticios, viéndose éstos directamente afectados. Lo anterior, ya que, debido al poco tiempo disponible para la adquisición y/o preparación de comida saludable, sumado al desconocimiento de la existencia de comidas igual de ricas, pero con un contenido nutricional saludable, hace que para las personas lo más fácil y rápido de ingerir sea la comida rápida. Generalmente se ignora el contenido nutricional y procedencia de este tipo de comida, lo que conlleva a un daño irreversible en la salud. Por consiguiente, debido a estos factores, gran parte de la población tiene enfermedades crónicas como la obesidad y la diabetes (Ge et al., 2015).

Un sistema de recomendación de recetas saludables es un software que sugiere recetas saludables, pero además toma en cuenta las preferencias y gustos personales de cada usuario, con el fin de que cada receta esté relacionada con el perfil personal establecido para cada usuario. Por ende, cada usuario obtendrá distintos tipos de recetas con una composición nutricional más saludable (Trang Tran et al., 2018).

El trasfondo de efectuar un sistema de recomendación de recetas basadas en preferencias personales, es que no se pierdan las preferencias o se restrinjan sus alimentos favoritos manteniendo un hábito de alimentación rico en nutrientes y en sabor.

1.1.4 Análisis de los Principales Trabajos Realizados en el área o tema de la propuesta

En el área de la recomendación de comidas y recetas saludables, existen diversos trabajos realizados, de los cuales destacan los siguientes:

- Yum-Me: es un sistema personalizado de recomendación de comidas saludables, diseñado para cumplir los objetivos de salud de las personas, las restricciones dietéticas y las preferencias de alimentos, proyectando el perfil aprendido de acuerdo a los parámetros establecidos, para encontrar lo que sea más atractivo para el usuario (Yang et al., 2017).
- Health- Aware Food Recommender System: es un sistema de recomendaciones de recetas de comida en general, que no solo se adaptan a las preferencias del usuario, sino que también puede tener en cuenta la salud del usuario, actualmente poseen una aplicación móvil y web (Ge et al., 2015).
- Algoritmo de recomendación de recetas: En (Trang Tran et al., 2018), se analizan los ingredientes de recetas junto a la preferencia de los usuarios, para recomendar variantes más saludables de una receta. En este trabajo se logra recomendar recetas con características derivadas de combinaciones de ingredientes e información nutricional (Trang Tran et al., 2018).

Si bien existen diferentes sistemas similares, en este proyecto se busca desarrollar una implementación propia de este tipo de sistema, como piloto que permita a futuro extender las características de este tipo de sistemas, incorporando por ejemplo otro tipo de información como enfermedades de usuarios o comentarios de recetas, así como experimentar con diferentes algoritmos de recomendación. Cabe señalar que en este Proyecto de Título sólo se implementará un algoritmo de recomendación, que utilizará información nutricional y ratings de recetas. En cualquier caso, el piloto considerará como usuarios potenciales a estudiantes universitarios.

1.1.5 Requisitos funcionales y limitaciones

1.1.5.1 Funcionales

- El sistema debe implementar un algoritmo de recomendación basado en técnicas de Machine Learning.
- El sistema debe permitir gestionar perfiles de usuarios que incluyan información sobre preferencias de recetas.
- La aplicación web con el algoritmo de recomendación ya implementado, debe mostrar a través de la pantalla los resultados al usuario.
- El algoritmo de recomendación debe permitir la selección de recetas saludables, a partir de información sobre composición nutricional de cada receta.

1.1.5.2 Limitaciones

- Lo primero y más importante es que sin un perfil de usuario es imposible hacer la recomendación, ya que no se tendrían los parámetros necesarios para ejecutar el algoritmo de forma eficiente.
- El sistema contemplado en el contexto de este Proyecto de Título no hace dietas específicas, ni toma en cuenta posibles enfermedades de usuarios, solo recomienda recetas saludables dependiendo de los gustos de cada usuario.
- Se consideran algoritmos descritos en la literatura, lo que puede estar sujeto a posibles ajustes de implementación.

1.2 Factibilidad del Proyecto

1.2.1 Factibilidad Técnica

En este proyecto el objetivo principal es implementar un algoritmo de recomendación para un sistema piloto de recetas saludables. El algoritmo a implementar corresponde a uno descrito en la literatura, lo que puede estar sujeto a posibles ajustes de implementación.

El lenguaje de programación para el diseño web y para la implementación de los algoritmos de recomendación será Python, en su versión 3.8.2. Y el editor de texto SublimeText versión 3.

Por otra parte, los conocimientos específicos sobre recetas saludables se obtendrán con el estudio e investigación durante el proyecto, complementando la información nutricional con estudiantes de último año de la carrera de nutrición y dietética.

El autor de esta propuesta cuenta con los conocimientos necesarios, tanto en diseño web como también en desarrollo de softwares. Lo que se complementará con el estudio teórico de algoritmos en artículos ya identificados y explicaciones del profesor guía.

1.2.2 Factibilidad Operativa

Este es un trabajo experimental y que considera un sistema de recomendación genérico, por lo que no hay exigencias de usuarios específicos. Durante el proyecto se implementará un software que tendrá como fin sugerir recetas más saludables que permitan mejorar la alimentación de los usuarios que hagan uso de la aplicación. Este sistema será un piloto, por lo que deberá contar con un diseño arquitectónico que permita su posterior extensión, incorporando nuevos algoritmos o fuentes de información.

1.2.3 Factibilidad Económica

El proyecto será desarrollado durante un periodo aproximado de 7 a 8 meses de trabajo, este periodo incluye planeación de los algoritmos a utilizar, herramientas necesarias, requerimientos técnicos, diseño web y desarrollo de software. Existen ciertos costos operacionales que hay que tener en cuenta como las horas humanas de trabajo, luz, internet, computador, entre otros recursos necesarios para el desarrollo, que no son contabilizados porque son parte de la actividad de Proyecto de Título, por lo que no se contempla su pago.

El beneficio que se obtiene al realizar este trabajo es la experiencia desarrollando y llevando a cabo un proyecto, conocimiento de nuevas herramientas de trabajo, conocer las necesidades de los usuarios, así como también el saber manejar y controlar la información que se tendrá en la palestra.

2 Capítulo 2: Marco Teórico (Conceptos)

2.1 Sistemas de Recomendación

Se define como Sistema de Recomendación a cualquier sistema que guía a un usuario de forma personalizada a objetos interesantes o útiles en un gran espacio de opciones posibles. Los sistemas se aplican de forma exhaustiva con el fin de recomendar servicios y productos (p. ej., *películas, libros, computadores, servicios financieros*) que satisfagan las necesidades y preferencias de los usuarios. En la actualidad se utilizan los sistemas de recomendación como una posible solución para ayudar a los usuarios a hacer frente a la cantidad masiva relacionada con recetas y alimentos saludables (Burke et al., 2011).

Los sistemas de recomendación nacen en el contexto de que ante la abundancia de información que existe actualmente, el encontrar información valiosa para el usuario se vuelve muy difícil. Bajo esta problemática nace la necesidad de crear una herramienta eficaz para extraer información útil y entregarla de forma eficiente. Un sistema de recomendación predice las preferencias de los usuarios para elementos sin clasificar y recomienda nuevos elementos a los usuarios (Trang Tran et al., 2018).

Existen distintas técnicas de recomendación que dependen del enfoque al cual va dirigido, como lo son las que se mencionan en la siguiente subsección:

2.1.1 Sistemas de recomendación basados en filtrado colaborativo

Filtrado Colaborativo (CF), es una de las técnicas de recomendación más investigadas. La idea básica es usar el conocimiento de una multitud de usuarios. En primera instancia un usuario **califica** elementos de forma implícita, es decir, asignando una valoración a un elemento, o también puede calificar un elemento de forma explícita, luego el sistema de recomendación identifica a los usuarios cuyos gustos sean similares a los de un usuario determinado y recomienda elementos que a los usuarios anteriores le han gustado (Ekstrand et al., 2011).

El dominio de información para el filtro colaborativo consiste en *usuarios* que han expresado preferencias para varios ítems. La preferencia expresada por el usuario para un ítem es nombrado como rating y usualmente es representado por una tripleta de elementos (User, Item, Rating). El rating es tomado como un número real o valor entero. En algunos casos es representado por (likes/dislikes) o escalas de 0 a 5 estrellas (Ekstrand et al., 2011).

La **Tabla 1** muestra un ejemplo de calificaciones para 3 usuarios y 4 películas en un sistema de recomendación, las celdas marcadas con “?” indican valores sin calificar.

Tabla 1: Evaluación simple de usuarios en una escala de 1 a 5

	Batman Inicia	Alicia en el País de las Maravillas	Tonto y Retonto	Equilibrio
Usuario A	4.4	¿	3	5
Usuario B	¿	5	4.4	¿
Usuario C	5	4.4	2	¿

Según (Ekstrand et al., 2011), Filtrado colaborativo se implementa usando diferentes métodos, dentro de los cuales se encuentran los siguientes:

- Predictores de línea base: Estos métodos son útiles para establecer algoritmos base no personalizados con los que se pueden comparar algoritmos personalizados. Los algoritmos que no dependen de las calificaciones del usuario también pueden ser útiles para proporcionar predicciones para nuevos usuarios.
- Basados en usuario: Busca otros usuarios cuya calificación sea similar al del usuario actual y use sus calificaciones en otros elementos para predecir lo que le gustará al usuario actual.
- Basado en ítem: En lugar de utilizar similitudes entre el comportamiento de calificación de los usuarios para predecir las preferencias, se caracteriza en si dos ítems tienden a tener los mismos gustos por usuario, entonces se espera que los usuarios tengan preferencias similares por elementos similares.
- Reducción de dimensionalidad: Un ítem y un usuario son representados mediante un vector dimensional, debido a esto, hay redundancia en estas dimensiones, ya que tanto los usuarios como los ítems suelen ser agrupados en grupos con perfiles de preferencias similares. Por ejemplo, muchas películas de ciencia ficción gustarán en grados similares al mismo grupo de usuarios.

2.1.1.1 Recomendación basados en vecinos cercanos

K-Nearest-Neighbor (kNN) es un algoritmo basado en filtrado colaborativo que encuentra similitudes entre usuarios para generar una recomendación. Este algoritmo, utiliza los (K) vecinos más similares a un usuario para sugerir un nuevo producto.

De este modo, si, por ejemplo, a un usuario le gusta el guacamole, KNN a ese usuario le recomendaría la tortilla de patata, la pizza y la pasta, ya que el algoritmo toma las preferencias de quien tiene gustos similares.

Este algoritmo se clasifica de dos formas:

- Supervisado: Se refiere a que el conjunto de datos de entrenamiento esta etiquetado con la clase o resultado esperado dado una fila de datos.
- Basado en instancia: Esto quiere decir que el algoritmo no aprende explícitamente un modelo. Este memoriza las instancias de entrenamiento que son usadas para la fase de predicción.

2.1.2 Sistemas de recomendación basados en contenido

Estos sistemas pueden hacer un servicio personalizado de recomendación explotando información sobre descripciones de elementos disponibles (p.ej., género y director de películas) y perfiles de usuario que describan sus preferencias y gustos. La tarea principal de un sistema basado en contenido es analizar la información sobre las preferencias de un usuario y las descripciones de los elementos consumidos por dicho usuario, para luego recomendar elementos basados en esta información (Pazzani & Billsus, 2007).

Para (Pazzani & Billsus, 2007), la mayoría de los sistemas de recomendación utilizan un perfil de acuerdo a los intereses del usuario. Este perfil puede constar de diferentes tipos de información. En esta ocasión se centrará en dos tipos de información del usuario:

- Un modelo de las preferencias del usuario, es decir, una descripción de los tipos de elementos que le interesan. Hay muchas representaciones alternativas posibles de esta descripción, pero una representación común es una función que para cualquier elemento predice la probabilidad de que el usuario esté interesado en ese elemento. Por motivos de eficiencia, esta función se puede utilizar para recuperar los n elementos que probablemente sean de interés para el usuario.
- Un historial de las interacciones del usuario con el sistema de recomendaciones. Esto puede incluir almacenar los artículos que un usuario ha visto junto con otra información sobre la interacción del usuario (por ejemplo, si el usuario ha comprado el artículo o una calificación que el usuario le ha dado al artículo). Otros tipos de historial incluyen guardar consultas escritas por el usuario (por ejemplo, que un usuario buscó un restaurante italiano en Roma).

2.1.3 Sistemas de recomendación basados en conocimiento

El sistema de recomendación basado en conocimiento, utiliza la información sobre los usuarios y productos para que el sistema siga como foco de recomendación dicha información recopilada, razonando sobre qué productos cumplen con los requisitos del usuario. A diferencia del filtrado colaborativo, el sistema de recomendación basado en conocimiento no depende de una base de calificaciones por parte de usuarios, bajo esta premisa el sistema de recomendación genera una sugerencia al usuario objetivo (Burke et al., 2011).

Para (Felfernig et al., 2007), existen 4 pasos básicos para un sistema de recomendación basado en conocimiento:

- 1) Especificación de requisitos: Los usuarios pueden interactuar con un sistema de recomendación para especificar sus requisitos, generalmente se hace a través de una aplicación web.
- 2) Reparación de requisitos inconsistentes: Si el sistema de recomendación no puede encontrar una solución, sugiere un conjunto de acciones de reparación, es decir, presenta alternativas a los requisitos del usuario para garantizar la identificación de una recomendación
- 3) Presentación de resultados: Se entrega un conjunto de alternativas al usuario. Generalmente son presentados en una lista clasificada de acuerdo al elemento de preferencia del usuario
- 4) Explicación: Para cada alternativa presentada, el usuario puede activar una sección de explicación para entender el por qué se ha recomendado un artículo en específico

2.1.4 Sistemas de recomendación híbridos

Estos sistemas combinan las técnicas mencionadas anteriormente. “*Un sistema que combina tecnologías A y B intentan usar las ventajas de A para corregir las desventajas de B*” (Ricci, 2010).

Por ejemplo, los métodos de CF tienen que enfrentar el problema del nuevo ítem, este problema aparece en dominios como los artículos de noticias donde hay un flujo constante de nuevos elementos y cada usuario solo califica algunos. También se conoce como el problema del "evaluador temprano", ya que, la primera persona que califica un elemento obtiene pocos beneficios al hacerlo y proporciona poca o nula información al sistema de recomendación, tales calificaciones tempranas no mejoran la capacidad del usuario para compararlo con otros (Avery & Zeckhauser, 1997).

Los enfoques CB pueden abordar este problema porque la predicción de nuevos elementos generalmente se basa en descripciones disponibles de estos artículos (Burke et al., 2011). Burke presenta algunos enfoques híbridos que combinan CB y CF, estos enfoques pueden ser de *ponderación, conmutado, mixto, combinación de funciones, cascada, función aumento y meta-nivel*.

2.1.5 Recomendaciones individuales y grupales

Los sistemas de recomendación, como está descrito en la sección **Sistemas de Recomendación**, en general se enfocan en recomendar elementos para usuarios individuales. Sin embargo, existen situaciones donde los sistemas de recomendación deberían soportar un grupo de usuarios. Por ejemplo, un paquete turístico para un grupo de amigos o un destino de fiesta para navidad para todo el grupo de una empresa. En tales situaciones un sistema de recomendación grupal (Masthoff, 2011) se considera como solución.

2.1.6 Sistemas de recomendación de alimentos

“¿Qué debemos comer para el almuerzo?” o “¿Qué tan saludable es lo que estoy comiendo?”, “¿Me estaré haciendo un daño?”, son preguntas habituales que se deben responder con mucha frecuencia. Mientras que muchos sistemas de recomendación solo buscan coincidir con las preferencias de los usuarios relacionados con música, películas o libros, recientemente también se han aplicado en el dominio de alimentos, para dar respuestas confiables a las preguntas anteriores.

En relación con el consumo de alimentos en estos días, es notable que ha habido un aumento de enfermedades relacionadas con el estilo de vida, como diabetes y obesidad, que son la causa de muchas enfermedades crónicas (Robertson et al., 2004).

En este contexto, los sistemas de recomendación de alimentos también son investigados como un medio potencial para ayudar a las personas a alimentarse más sanamente (Elsweiler et al., 2015).

Tiene sentido utilizar sistemas de recomendación de alimentos como parte de una estrategia para cambiar el comportamiento alimentario de los usuarios. En este caso, los sistemas de recomendación de alimentos no solo aprenden preferencias de los usuarios por ingredientes y estilos de comida, sino también, seleccionan alimentos saludables al tomar en cuenta los problemas de salud, las necesidades nutricionales y las conductas alimentarias anteriores.

De acuerdo con (Mika, 2011), los sistemas de recomendación de alimentos se pueden clasificar en 4 tipos.

Los del primer tipo (tipo 1) recomiendan recetas más saludables o alimentos que sean más similares a los que le gustaron al usuario en el pasado, es decir, toma en cuenta los gustos de los usuarios que fueron determinados pretéritamente y genera una recomendación comparando sus gustos pasados con las nuevas recetas.

El segundo tipo de sistema de recomendación (tipo 2) solamente recomienda a los usuarios, aquellos artículos que hayan sido identificados previamente por el recomendador de recetas saludables, la recomendación se hace de acuerdo a la necesidad nutricional del usuario.

También se investigan otros dos tipos de recomendación de alimentos, como los sistemas tipo 3 y tipo 4, que consideran otros escenarios al hacer recomendaciones.

Los sistemas de tipo 3 generan recomendaciones sobre la base de considerar los dos criterios anteriores para el propósito del equilibrio del usuario tomando en cuenta los alimentos que les gustan y los alimentos que deben consumir. Todos estos tipos de sistemas de recomendación están diseñados principalmente para usuarios individuales. Los sistemas de tipo 4 representan recomendaciones grupales en las cuales los alimentos son consumidos por grupos de usuarios en lugar de por individuos.

En las siguientes subsecciones se profundizará con más detalle en los sistemas tipo 1, 2 y 3. No se va a profundizar más sobre los sistemas tipo 4, ya que, en este trabajo no se contempla recomendación grupal.

2.1.6.1 Tipo 1

En el ámbito de la alimentación saludable, el aprendizaje de los gustos de los usuarios se reconoce como un paso esencial previo para sugerir recetas que gustarán a los usuarios.

Toda la investigación discutida en esta subsección tiene como objetivo recomendar recetas o menús a usuarios de forma individual, todo ello con la finalidad de explorar los gustos de usuarios.

Para llevar a cabo un proceso de recomendación de recetas de alimentos a un usuario de forma individual, es necesario un método para crear un perfil de usuario, para así, aplicar algunos cálculos para identificar la similitud entre una receta y el perfil del usuario, identificando sus preferencias y gustos personales.

El método para crear el perfil de usuario es conocido como TF-IDF (Term Frequency- Inverse Document Frequency).

Según (El-Dosuky et al., 2012), para recomendar una receta a un usuario, esta se puede llevar a cabo siguiendo los siguientes pasos:

- El primer paso es construir el perfil de usuario, este se construye calculando valores TF-IDF para cada término.
- El siguiente paso es tomar la información de los gustos y preferencias directamente del usuario o si previamente ya está creado el perfil de usuario en el sistema de recomendación, sacar la información directamente desde su perfil.
- Luego, usando un sistema basado en conocimientos, cada usuario califica manualmente los alimentos de una categoría específica (por ej. ejemplo: frutas, verduras, carne, etc.) como relevante o no relevante para su interés. Después de eso, el sistema de recomendación calcula la similitud entre los alimentos y el perfil de usuario previamente calculado.

Si el valor de la similitud es más alto que un rango predefinido, se recomienda el alimento, de lo contrario, se ignora.

Se determina la frecuencia de término tf_{ij} para un término t_i dentro de una receta a_j . Donde n_{ij} es el número de ocurrencias del término t_i en la receta a_j , que es dividido por el número total de términos en el documento $\sum_k n_{kj}$, como muestra la ecuación (1).

$$tf_{ij} = \frac{n_{ij}}{\sum_k n_{kj}} \quad (1)$$

La frecuencia inversa de documento (IDF) se calcula según la ecuación (2), donde \log es el logaritmo multiplicado el conjunto de recetas $|A|$, dividido un subconjunto de un término t_i que pertenece a una receta a .

$$idf_i = \log \frac{|A|}{|\{t_i \in a\}|} \quad (2)$$

Dividiendo el número total de recetas por el número de recetas que contienen el término t_i . El valor final se calcula multiplicando TF e IDF:

$$tf_i idf_i = tf_i \times idf_i \quad (3)$$

La receta de comida se puede definir mediante un conjunto de p ingredientes:

$$A = \{c_1^a, c_2^a, c_3^a, \dots, C_p^a\} \quad (4)$$

El perfil de usuario, U , consta de q ingredientes encontrados en las recetas c_i^u que califica el usuario:

$$U = \{c_1^u, c_2^u, c_3^u, \dots, C_q^u\} \quad (5)$$

La similitud entre una receta de comida y el perfil del usuario puede calcularse mediante:

$$B(U, A) = \begin{cases} 1 & \text{if } |U \cap A| > 0 \\ 0 & \end{cases} \quad (6)$$

Los autores emplean coseno binario para calcular la similitud entre las recetas y el perfil de usuario, donde A es la receta y U es el perfil del usuario.

$$B(U, A) = \frac{|U \cap A|}{|U| \times |A|} \quad (7)$$

Para comparar dos nuevos elementos n_i y n_j , se puede crear un vector V que represente la similitud entre los ingredientes presentes en las recetas y las preferencias de usuarios, Donde w_i es el peso de c_i .

$$V_l = (\langle c_1^l, w_1^l \rangle, \dots, \langle c_p^l, w_p^l \rangle) \quad l \in \{i, j\} \quad (8)$$

La similitud entre los alimentos a_i y a_j es:

$$\text{SemRel}(a_i, a_j) = \cos(V_i, V_j) = \frac{V_i \cdot V_j}{\|V_i\| \times \|V_j\|} \in [0,1] \quad (9)$$

El marco propuesto en la **Tabla 2** fue implementado por el Departamento de Agricultura de los Estados Unidos (USDA), usando el lenguaje de programación Java, esto permite al usuario formular consultas y ejecutarlas para recuperar alimentos relevantes.

Las pruebas realizadas se basan en un corpus de 300 alimentos extraídos del Departamento de Agricultura de los Estados Unidos (USDA), como se muestra en la **Tabla 2**

Tabla 2: Base de Datos de Comidas

Grupo	N° de Ítems	Grupo	N° de Ítems
Indio Americano	165	Cordero y Terneros	345
Comidas de bebés	329	Legumbres	386
Productos Horneados	497	Nuez y Semillas	128
Productos de Carne	757	Productos de Cerdo	340
Bebidas	284	Productos Avícolas	388
Cereales	408	Restaurante y comidas	121
Cereal en Granos	184	Salchichas y Almuerzo	234
Leche y Huevos	253	Aperitivos	169
Comida Rápida	385	Sopas y Salsas	510
Grasas y Aceites	220	Espicias y hierbas	61
Pescado	258	Dulces	341
Jugos y Frutas	329	Vegetales	841

Se utilizaron 5 usuarios con intereses diferentes, pero bien definidos.

Un ejemplo de interés del usuario es "Frutas", cada usuario ha calificado manualmente los alimentos como relevante o no relevante para su interés.

Para cada usuario se dividieron los alimentos en dos conjuntos diferentes: el 60% de los alimentos son el conjunto de entrenamiento y el otro 40% de los alimentos son el conjunto de prueba. Los asesores calculan la similitud entre los alimentos, elementos y perfil de usuario previamente calculado. Si el valor de similitud calculado es mayor que un valor de corte predefinido, el alimento se recomienda, de lo contrario se ignora.

La evaluación de los recomendadores se realiza midiendo la exactitud, precisión, recuperación, especificidad y medida F. Esto se hace calculando una matriz para cada usuario. La **Tabla 3** muestra los resultados de la evaluación.

Tabla 3: Evaluación de Resultados

	Exactitud	Precisión	Recuperación	Especificidad	Medida-F
TF-IDF	90%	90%	45%	99%	60%
B. Coseno	47%	23%	95%	36%	37%
Jaccard	93%	92%	58%	99%	71%
SemRel	57%	26%	92%	47%	41%
Propuesto	94%	93%	62%	99%	74%

En otra investigación (Freyne & Berkovsky, 2010) usan un algoritmo basado en contenido, para predecir el valor de calificación de una receta objetivo sobre la base de explotar la información de los ingredientes correspondientes incluidos en las recetas.

El proceso de predicción incluye los siguientes pasos:

- Descomponer una receta objetivo sin clasificar rt en ingredientes $ingr\ 1, ingr\ 2, ingr\ 3, \dots, ingr\ n$.
- Asignar el valor de calificación para cada ingrediente en la receta objetivo rt de acuerdo a ecuación (10). Particularmente, el valor de calificación del usuario u_a para un ingrediente específico $ingr\ i$ en la receta objetivo rt , es decir, $rat(u_a, ingr\ i)$ se calcula utilizando valores de calificación del usuario u_a para todas las recetas rl que contienen el ingrediente $ingr\ i$, es decir, $rat(u_a, rl)$.

El valor rl es el número de recetas que contienen $ingr\ i$.

$$rat(u_a, ingr\ i) = \frac{\sum_{l \text{ s.t. } ingr\ i \in r_l} rat(u_a, r_l)}{l} \quad (10)$$

- Predecir el valor de calificación del usuario u_a para la receta objetivo rt , es decir, $p_{pred(u_a, r_t)}$ basado en promedio de todos los valores de calificación de todos los ingredientes, $ingr\ 1, \dots, ingr\ j$ incluido en esta receta.

$$p_{pred(u_a, r_t)} = \frac{\sum_{j \in r_t} rat(u_a, ingr\ j)}{j} \quad (11)$$

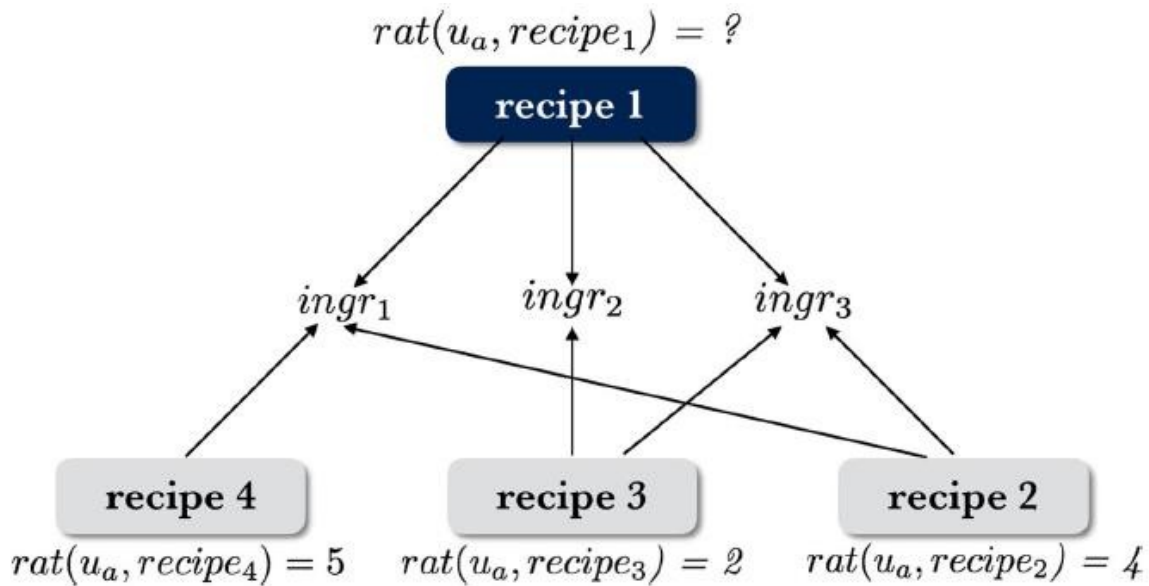
Se recomendarán recetas con un alto valor de calificación previsto al usuario u_a . A continuación, se presenta una ilustración para predecir un valor de calificación para una receta objetivo rt .

Por ej. Supongamos que la *receta 1* es una receta que no ha sido calificada por el usuario u_a . Esta receta Incluye 3 ingredientes, es decir, *ingr 1*, *ingr 2* e *ingr 3*.

- *ingr 1* está incluido en la *receta 4* y la *receta 2*.
- *ingr 2* está incluido en la *receta 3*.
- *ingr 3* está incluido en la *receta 2* y la *receta 3*.

El rating de valores del usuario u_a para la *receta 2*, la *receta 3* y la *receta 4* son respectivamente 4, 2 y 5. Ver **Imagen 1**.

Imagen 1: Predicción del valor de calificación para una receta objetivo utilizando un algoritmo CB propuesto por (Freyne & Berkovsky, 2010).



De acuerdo con la **Imagen 1**, los valores de calificación para los ingredientes de la receta 1 se evaluarán de la siguiente manera:

$$\begin{aligned} \text{rat}(u_a, \text{ingr}_1) &= \frac{\text{rat}(u_a, \text{recipe}_4) + \text{rat}(u_a, \text{recipe}_2)}{2} = \frac{5 + 4}{2} = 4.5 \\ \text{rat}(u_a, \text{ingr}_2) &= \text{rat}(u_a, \text{recipe}_3) = 2 \\ \text{rat}(u_a, \text{ingr}_3) &= \frac{\text{rat}(u_a, \text{recipe}_2) + \text{rat}(u_a, \text{recipe}_3)}{2} = \frac{4 + 2}{2} = 3 \end{aligned}$$

El valor de predicción de la receta 1 para el usuario u_a se calcula de la siguiente manera:

$$\begin{aligned} \text{pred}(u_a, \text{recipe}_1) &= \frac{\text{rat}(u_a, \text{ingr}_1) + \text{rat}(u_a, \text{ingr}_2) + \text{rat}(u_a, \text{ingr}_3)}{3} = \frac{4.5 + 2 + 3}{3} \\ &= 3.166 \end{aligned}$$

Recientemente, se han incluido algunos enfoques nuevos para los sistemas de recomendación de alimentos, como el uso de etiquetas para diferentes grupos de usuarios (Svensson et al., 2000), algoritmos de aprendizaje activo y factorización matricial (Elahi et al., 2015).

Particularmente en (Svensson et al., 2000), los autores diseñan una tienda de alimentos en línea, con el propósito de sugerir tipos de alimentos que deben ser comprado por los usuarios, basado en recetas que los usuarios han elegido antes, grupos de usuarios que son etiquetados y nombrados de acuerdo con su contenido, por ejemplo, "amantes de la carne", "vegetarianos", "Amantes de las especias", etc.

Las recetas recomendadas se determinan en base a tres diferentes características elegidas por los usuarios: grupos de usuarios, categorías de alimentos (p.ej., pescado, oriental, italiano, carnes rojas, pollo) e ingredientes (p. ej., Arroz, tallarines, curry, tomates). Los usuarios seleccionan recetas de la lista de recomendaciones y las colocan en una cesta de la compra.

Luego, todos los ingredientes de las recetas elegidas se agregan automáticamente a la lista de artículos que se entregaran al usuario. Además, para mejorar la interacción entre los usuarios y las recetas, se agregan algunas características adicionales (p.ej., el valor de calificación promedio o los comentarios de otros usuarios) en cada receta recomendada.

(Elahi et al., 2015) propone un sistema de recomendación de alimentos utilizando un algoritmo de aprendizaje activo y factorización matricial. Esta investigación proporciona a los usuarios una interacción completa con las tecnologías computacionales con el propósito de recopilar las preferencias del usuario a largo plazo en términos de calificaciones y etiquetas de recetas.

Además, al solicitar recomendaciones, se requieren usuarios para proporcionar preferencias a corto plazo en referencia a los ingredientes que desean cocinar o incluir en la comida. Luego, el sistema utiliza ambos tipos de preferencias del usuario para realizar recomendaciones.

Las preferencias a largo plazo son explotadas por un modelo de predicción de calificación de factorización matricial, modelo diseñado para considerar tanto las etiquetas de usuario como las calificaciones. Cada usuario y cada receta están modelados por vectores que representan sus características latentes. El valor de calificación de un usuario para un artículo en específico se encuentra calculando el producto interno del usuario y los vectores del artículo.

Con preferencias a corto plazo, el sistema filtra las recetas de acuerdo con las preferencias actuales del usuario, el sistema recomienda al usuario las recetas con los valores de calificaciones más altos.

Si bien la mayor parte de la investigación existente en el ámbito de los alimentos solo se centra en hacer recomendaciones sobre alimentos o recetas, pero también es necesario que los usuarios planifiquen menús con la combinación de muchas recetas en comidas nutricionalmente completas. Con esta idea (Kuo et al., 2012) propone un mecanismo de inteligencia de planificación de menú eficaz, que sugiere un conjunto de recetas mediante el uso de un algoritmo gráfico. Primero, se construye un gráfico de recetas no dirigido, donde cada nodo es una receta que posee un conjunto de ingredientes, cada línea representa la relación entre dos recetas y el peso de la línea representa la distancia entre dos recetas ver (**Imagen 2**).

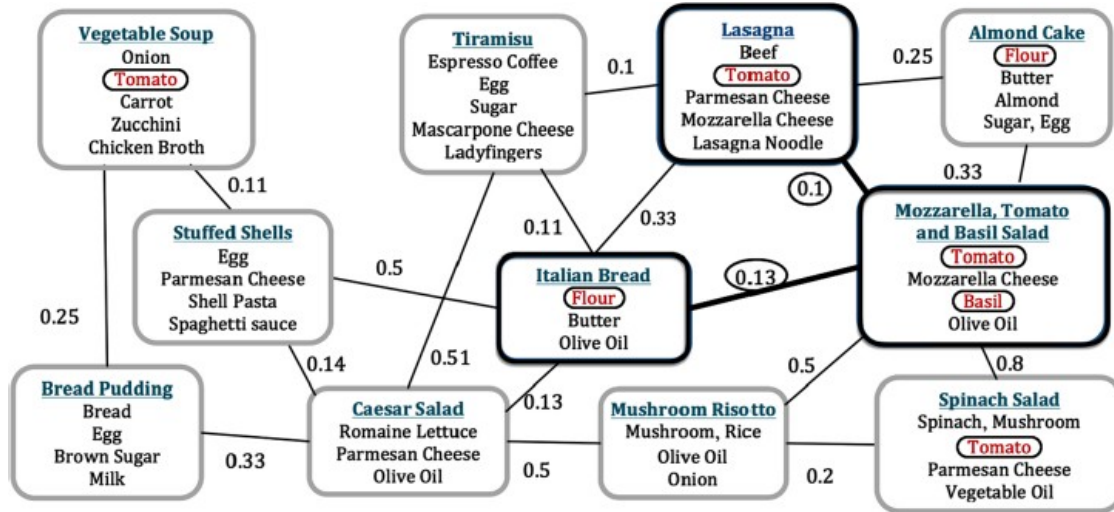
El peso de cada borde que conecta dos recetas diferentes describe el costo de un menú que incluye estas dos recetas. Cuanto menor sea el peso, mayor será la probabilidad de que dos recetas coexistan en un menú.

Por ejemplo, en la **Imagen 2**, la receta "Pan italiano" tiene una relación de coincidencia con cinco recetas, es decir, "Tiramisú", "Lasaña", "Ensalada de mozzarella, tomate y albahaca", "Ensalada César" y " Almejas estofadas". Entre estas cinco recetas, " *Tiramisú* " tiene la mayor relación de coocurrencia con " *Pan italiano* " ya que el peso de su borde es más bajo (es decir, 0,11). Mientras que " *conchas rellenas* " tiene la relación más baja con " *pan italiano* " porque el peso de su borde es más alto (es decir, 0.5).

Además, el costo de un menú también se define como una suma ponderada de los bordes del árbol de expansión mínima en el subgrafo inducido. A partir de eso, se crea un plan de menú eligiendo un conjunto de recetas que contiene todos los ingredientes de la consulta (es decir, los ingredientes solicitados por los usuarios) y el costo del menú es mínimo.

Por ejemplo, en la **Imagen 2**, con los ingredientes de consulta {tomate, harina, albahaca}, podemos encontrar muchos conjuntos diferentes de recetas, por ejemplo; {"Mozzarella, Tomate y Ensalada de albahaca", "Lasaña", "Pan italiano"}, {"Mozzarella, tomate y albahaca", "Lasaña", "Pastel de almendras"}, {"Mozzarella, tomate y albahaca", "Pan italiano", "Ensalada de espinacas "}, Etc. Sin embargo, el primer conjunto {"Ensalada de mozzarella, tomate y albahaca", "Lasaña", "Pan italiano"} se recomendará a los usuarios porque el costo total del menú es mínimo (es decir, 0.23).

Imagen 2: Un ejemplo del gráfico de recetas para la planificación del menú "Tomate, harina, albahaca" (ingredientes que se muestran con líneas negras) son ingredientes de consulta. El plan de menú recomendado es un conjunto de recetas. {"Mozzarella, tomate y ensalada de albahaca", lasaña", "Pan italiano"} (nodos con marcos negros) cuyo costo total del menú es mínimo (Kuo et al., 2012).



2.1.6.2 Tipo 2

Hoy en día, los hábitos alimenticios poco saludables y la nutrición desequilibrada aumentan las posibilidades de personas con obesidad y otras afecciones relacionadas con la dieta, como diabetes, hipertensión, etc.

Como tratamiento o medida preventiva, los nutricionistas o dietistas generalmente recomiendan ejercicios regulares y diseñan planes de comidas individualizados para sus pacientes. Desafortunadamente, estos expertos en nutrición, están sobrecargados con demasiados pacientes para adaptar manualmente un plan de comidas individualizado para cada usuario. Ahí es donde los sistemas de recomendación de alimentos pueden usarse como un sistema inteligente de consulta nutricional. En esta subsección, se describen sistemas de recomendación que tienen en cuenta las necesidades nutricionales.

Primero, se discute un escenario de recomendación simple que muestra cómo los elementos del menú pueden ser recomendados a los usuarios, sobre la base de considerar sus necesidades nutricionales, así como sus problemas de salud. En este contexto, un usuario ingresa cierta información personal (por ejemplo, edad, sexo, ocupación, actividades físicas, problemas de salud, etc.). Esta información es la base para seleccionar los alimentos que mejor se adaptan a las necesidades nutricionales del usuario. El siguiente ejemplo será una ilustración de este escenario.

En un sistema de recomendación de menús, suponemos que hay 5 menús con la información correspondiente, p.ej., ingredientes, calorías, grasas (consulte la **Tabla 4**). Un usuario u_a ingresa lo siguiente información:

- Edad: 52.
- Sexo: masculino.
- Ocupación: oficinista.
- Actividad física: caminar (10 minutos / día).
- Problema de salud: cardiovascular.

Para recomendar menús apropiados al usuario u_a se deben realizar los siguientes pasos:

Paso 1: Se utiliza una tabla de energía (ver **Imagen 3**) para estimar la cantidad de calorías (en kcal) que el usuario u_a debería obtener por día.

La cantidad de calorías consumidas por día para cada persona se estima de acuerdo con la edad, el sexo y el valor de PAL (Nivel de actividad física). El valor PAL se clasifica en 3 tipos:

+ PAL = 1.4: Se utiliza para personas que tienen estilos de vida exclusivamente sedentarios (como trabajadores de oficina, mecánicos de precisión) con muy poca o ninguna actividad de ocio.

+ PAL = 1.6: Se utiliza para personas que tienen estilos de vida sedentarios, pero que requieren una energía adicional para caminar y/o pararse durante mucho tiempo, como asistentes de laboratorio, estudiantes y trabajadores de la línea de producción.

+ PAL = 1.8: Se utiliza para personas que tienen estilos de vida extensos, por ejemplo, vendedores, camareros, mecánicos, artesanos.

Imagen 3: Valores de referencia para la ingesta nutricional. Bonn 2. Edición, 1. Volumen (2015) publicado por la Asociación Alemana de Nutrición, la Asociación Austriaca de Nutrición y la Asociación Suiza de Nutrición.

Adults (years old)	Reference values for energy intake in kcal/day					
	PAL-value (1.4)		PAL-value (1.6)		PAL-value (1.8)	
	male	female	male	female	male	female
15 to 19	2600	2000	3000	2300	3400	2600
19 to 25	2400	1900	2800	2200	3100	2500
25 to 51	2300	1800	2700	2100	3000	2400
51 to 65	2200	1700	2500	2000	2800	2200
> 65	2100	1700	2500	1900	2800	2100

Tabla 4: Una lista de menús disponibles con la información correspondiente

Menús	Ingredientes principales	Calorías (kcal)	Grasas (%)
<i>menú 1</i>	Mantequilla, pollo, papa, pepino, crema, ajo, sal, pimienta.	2010	27
<i>menú 2</i>	Cerdo, champiñones, brócoli, pimentón, cebolla verde, aceite, sal, pimienta.	2200	30
<i>menú 3</i>	Pollo, champiñones, ensalada, cebolla, aceite de oliva, tomate, sal, pimienta.	1500	21
<i>menú 4</i>	Carne de res, camarones, tomate, ajo, huevo, sal, pimienta.	2400	31
<i>menú 5</i>	Carne de cerdo, frijoles, tomate, aceite de calabaza, ensalada, huevo, sal, pimienta.	1700	25

En este ejemplo, el usuario u_a es un oficinista con muy poca actividad física (solo 10 minutos / día para caminar), eso significa que su valor PAL pertenece al primer tipo. Mirando información sobre edad, sexo y actividad física de la **Imagen 3**, podemos encontrar la ingesta diaria de calorías por u_a es de 2200kcal.

Paso 2: Filtrado de menús con la cantidad de calorías menor o igual a 2200 kcal / día.

Paso 3: Clasificar los menús filtrados en orden ascendente de grasa (dado que u_a tiene una enfermedad cardíaca, primero se le mostrarán menús menos grasos).

En la **Tabla 4**, podemos ver que el menú 4 no se agregará a la lista de recomendaciones porque sus calorías son más de 2200 kcal. La lista de menús recomendados se clasifica en orden ascendente de acuerdo al porcentaje de grasas (consulte la **Tabla 5**).

Con el fin de mejorar las condiciones de salud de los usuarios, (Ueta et al., 2011) propone una recomendación de receta orientada a objetivos, con el fin de proporcionar una lista de platos que contienen el tipo correcto de nutriente para tratar los problemas de salud de los usuarios.

Para hacer eso, primero que nada, un usuario ingresa su problema de salud, por ejemplo, "Quiero curar mi acné". A continuación, el sistema analiza la solicitud del usuario e identifica las palabras clave que describen el problema de salud (por ejemplo, acné). El sustantivo se inserta en la base de datos de coocurrencia para buscar el nutriente que coocurre, es decir, buscar el nutriente que tenga relación con la palabra clave.

Por ejemplo, al buscar el sustantivo acné en la base de datos de coocurrencia, el ácido pantoténico se encuentra como un componente nutritivo que se puede usar para curar el acné porque coexiste con el "acné" con más frecuencia que cualquier otro nutriente.

Por último, los nutrientes identificados en el paso 3 se utilizan para encontrar los platos más cercanos a esos nutrientes en una base de datos de alimentos.

Tabla 5: Lista de menús recomendados para el usuario u_a

Menús	Ingredientes principales	Calorías (kcal)	Grasas (%)
<i>menú 1</i>	Mantequilla, pollo, patata, pepino, nata, ajo, sal, pimienta	1500	21
<i>menú 2</i>	Carne de cerdo, champiñones, brócoli, pimentón, cebolla verde, aceite, sal, pimienta	1700	25
<i>menú 3</i>	Pollo, champiñones, ensalada, cebolla, aceite de oliva, tomate, sal, pimienta	2010	27
<i>menú 5</i>	Cerdo, frijol, tomate, aceite de calabaza, ensalada, huevo, aceite de oliva, sal, pimienta	2200	30

Esta base de datos de alimentos incluye dos sub-bases de datos, una base de datos de nutrientes de ingredientes y una base de datos de información nutricional para recetas.

La base de datos de nutrientes de los ingredientes contiene información sobre el valor nutricional de cada ingrediente, mientras que la base de datos de información nutricional incluye tipos de recetas y la cantidad de nutrientes que contiene cada receta.

Los ingredientes de cada receta se identifican y luego se calculan sus elementos nutricionales utilizando la base de datos de nutrientes de ingredientes.

2.1.6.3 Tipo 3

Tener en cuenta las preferencias del usuario o las necesidades nutricionales de forma aislada a veces conduce a recomendaciones poco óptimas de alimentos. Por ejemplo, si los recomendadores solo tienen en cuenta las preferencias del usuario, también se fomentarán los malos hábitos alimenticios. Por el contrario, si solo se consideran las necesidades nutricionales, los alimentos propuestos a veces no serán atractivos para los usuarios.

Por lo tanto, considerando tanto las preferencias del usuario como las necesidades nutricionales parece brindar la mejor solución ya que los usuarios reciben recomendaciones más relevantes, se interesan más y se involucran cada vez más en usarlas.

En este ejemplo se discute un escenario de recomendación simple que muestra cómo un sistema de recomendación de alimentos puede sugerir elementos del menú, sobre la base de considerar tanto las preferencias del usuario como las necesidades nutricionales. En este ejemplo, se asume la existencia de una tabla de menú como se muestra en la **Tabla 4**.

Un usuario u_a proporciona información personal de la siguiente manera:

- Edad: 52.
- Género: masculino.
- Ocupación: oficinista.
- Actividad física: caminar (10 minutos / día).
- Problema de salud: cardiovascular.
- Ingredientes favoritos: tomate.

En este escenario, el sistema de recomendación considera tanto los ingredientes preferidos por el usuario u_a como la información adicional relacionada con el usuario (por ejemplo, edad, sexo, ocupación, actividad física y problema de salud). La lista de menús recomendados se crea realizando los siguientes pasos:

- Paso 1: Estimación de la cantidad diaria de calorías para el usuario u_a consultando la tabla de energía que se muestra en la **Imagen 4**. El usuario u_a es un trabajador de oficina y tiene muy poca actividad física por día (solo 10 minutos / día para caminar), por tanto, la ingesta de nutrientes del usuario u_a es de 2200 kcal.
- Paso 2: Filtrar los menús de la **Tabla 4** que contienen 2200 kcal de calorías menores o iguales e incluyen el ingrediente favorito “tomate”.
- Paso 3: Clasificar los menús filtrados en orden ascendente de grasa (debido a que u_a tiene una enfermedad vascular, se le mostrarán primero los menús con menos grasa).

Después de realizar estos pasos, hay dos menús, es decir, se recomendarán menú 3 y menú 5 al usuario u_a (ver **Tabla 6**).

También con el fin de equilibrar las preferencias de los usuarios y las necesidades nutricionales, (Elsweiler et al., 2015) propone dos enfoques para integrar aspectos nutricionales en recomendaciones.

El primer enfoque determina las compensaciones entre dar al usuario algunos alimentos que realmente le gustan y algunos alimentos que son realmente saludables para ella. Este enfoque se implementa mediante los siguientes pasos.

- Primero, un algoritmo de predicción estima las mejores recetas para el usuario, es decir, un conjunto de recetas con probabilidad predicha por encima de un cierto rango.
- A continuación, se calcula la cantidad de calorías y grasas por gramo de cada receta del conjunto elegido.
- Finalmente, las comidas con menos grasas o calorías por gramo se elegirán para la recomendación final.

En el segundo enfoque, en lugar de recomendar recetas individuales, propone planes de alimentación más completas, que se generan no solo en función de las preferencias alimenticias de los usuarios, sino que también se ajustan a las pautas nutricionales diarias (Harvey & Elswailer, 2015).

Para hacer recomendaciones, el usuario proporciona información al sistema recomendante sobre sus preferencias al calificar una serie de recetas en el sistema, con una escala de calificación de hasta 5 estrellas.

Además, el sistema de recomendación también tiene en cuenta la información personal adicional de los usuarios, como la altura, el peso, la edad, el nivel de actividad diaria y el objetivo (perder, ganar o mantener el peso) para calcular las necesidades nutricionales.

Los requerimientos nutricionales de los usuarios se calculan utilizando una versión actualizada de la ecuación de Harris Benedict (Roza & Shizgal, 1984).

Después de eso, el sistema de recomendación predice las calificaciones de las recetas no calificadas y envía una lista clasificada de recetas con calificaciones altas (por ejemplo, 4 o 5 estrellas) al "Planificador", quien en este caso es el sistema que planifica la receta.

El "Planificador" toma las n recetas principales de la lista clasificada de recetas y las divide en dos conjuntos separados: uno para los desayunos y otro para las comidas principales.

Finalmente se realiza una búsqueda completa para encontrar todas las combinaciones de estas recetas en la secuencia {desayuno, comida principal, cena principal} que satisfaga las necesidades nutricionales objetivo.

Tabla 6: Una lista de menús recomendados al usuario u_a sobre la base de considerar su ingrediente favorito (es decir, tomate) y sus necesidades nutricionales

Menús	Ingredientes principales	Calorías (kcal)	Grasas (%)
<i>menú 1</i>	Pollo, champiñones, ensalada, cebolla, aceite de oliva, tomate, sal, pimienta	1500	21
<i>menú 2</i>	Cerdo, frijol, tomate, aceite de calabaza, ensalada, huevo, sal, pimienta	1700	25

2.2 Capítulo 3: Estado del Arte

Es necesario hacer una compilación de otras investigaciones y establecer que se ha hecho recientemente sobre el tema seleccionado. Con el fin de determinar el enfoque que pueda marcar la diferencia entre lo ya investigado y lo próximo a investigar. Pues al revisar otras investigaciones se puede establecer de qué forma otros investigadores han estudiado el tema y así tomar esta información como guía.

Conceptos: Sistemas de recomendación, sistemas de recomendación de alimentos, recetas saludables, recetas de alimentos.

Resultados: La cantidad de artículos, papers e investigaciones encontrados fueron sobre los 329 resultados en Google Scholar y Web of Science “WERKEN UBB”.

Criterios de selección: Se revisaron 30 artículos, basándose por los más citados, investigaciones desde el año 2010 en adelante y tengan relación con sistemas de recomendación de alimentos y sistemas de recomendación de recetas.

Propósito: Saber los últimos trabajos y/o investigaciones relacionadas con el ámbito de sistemas de recomendación de alimentos y nuevos conceptos incorporados.

Fuente de Búsqueda: Web of Science “WERKEN” y Google Scholar.

Idioma de Búsqueda: inglés, español.

Los resultados seleccionados fueron los siguientes:

Fuente-Cita	Articulo/Paper	Enfoque Algorítmico	Tipo	Data set
(Gao et al., 2020)	Hierarchical Attention Network for Visually-Aware Food Recommendation	Filtrado colaborativo	1	https://www.kaggle.com/elisaxxygao/foodrecsysv1
(Bianchini et al., 2017)	PREFer: A prescription-based food recommender system	Filtrado colaborativo	1	Privado
(Ge, Elahi, et al., 2015)	Using Tags and Latent Factors in a Food Recommender System	Basado en Contenido	No especificado	Privado
(Lee et al., 2020)	A Multi-Period Product Recommender System in Online Food Market based on Recurrent Neural Networks	Filtrado colaborativo	1	Privado
(Cui et al., 2017)	HiRecS: A Hierarchical Contextual Location Recommendation System	Sistema híbrido	No especificado	No especificado

De esta manera podemos concluir que existen proyectos e investigaciones no del todo igual al expuesto en este proyecto, pero que usan la misma idea de utilizar los sistemas de recomendación con el fin de que los usuarios puedan tener una alimentación beneficiosa para su salud.

Tomando en cuenta lo mencionado anteriormente, es que se decide tomar un rumbo parecido, utilizando algoritmos y técnicas de recomendación descritas durante la literatura de algoritmos de recomendación.

3 Capítulo 4: Algoritmos investigados

Para este proyecto se investigaron algoritmos basados en factorización matricial, filtrado colaborativo y vecinos cercanos, utilizando la librería de recomendación “SURPRISE “.

Surprise es un scikit de Python para analizar y crear sistemas de recomendación que tratan generalmente con conjuntos de datos recopilados a través de feedbacks de usuarios.

Surprise fue diseñado con varios propósitos en mente, de los cuales podemos describir los siguientes:

- Brindar un control sobre sus experimentos. Para poder lograr esto, surprise pone un fuerte énfasis en la documentación de sus algoritmos, para que los usuarios no tengan grandes complicaciones al momento de querer implementar uno de sus algoritmos.
- Permite el control de grandes conjuntos de datos. Dando la posibilidad de utilizar conjuntos de datos integrados como, por ejemplo: Movielens, Jester, así como también nos permite usar conjuntos de datos propios.
- Proporciona varios algoritmos de predicción listos para su uso, como algoritmos de línea base, métodos del vecino más cercano, basados en factorización matricial (SVD, PMF, SVD++, NMF) y muchos otros. Además, medidas de similitud (coseno, MSD, Pearson) ya están integradas.
- Facilita la implementación de nuevas ideas de algoritmos.
- Proporciona herramientas para evaluar, analizar y comparar el desempeño de los algoritmos.

Una **medida de similitud** es cuanto se parecen las preferencias de un usuario y rankings respecto a un ítem comparado con otro usuario en un conjunto de datos en particular.

La factorización matricial, es un algoritmo de filtrado colaborativo que funciona descomponiendo una matriz de usuario-elemento en matrices de menor dimensión.

3.1 Descripción de algoritmos.

Para efecto del proyecto de título, para la etapa de recomendación de recetas nos enfocaremos en los algoritmos de factorización matricial, más específicamente en SVD++ o descrito también como SVDpp.

El algoritmo SVD++, es una extensión de SVD que toma en cuenta los ratings implícitos de un conjunto de datos.

La predicción \hat{r}_{ui} se establece como:

$$\hat{r}_{ui} = u + b_u + b_i + q_i^T \left(p_u + |I_u|^{-\frac{1}{2}} \sum_{j \in I_u} y_j \right) \quad (12)$$

Donde el termino y_j es un nuevo conjunto de factores de elementos que obtienen calificaciones implícitas. Una calificación implícita describe el hecho de que un usuario u calificó un artículo j , independientemente del valor de la calificación.

Si el usuario u es desconocido, entonces b_u y los factores p_u se asumen en cero. Lo mismo aplicaría para los ítems i , b_i , q_i , e y_i .

- $N_factors$: Cantidad de factores a tener en cuenta para la recomendación.
- N_users : Cantidad de usuarios en el conjunto de datos.
- N_items : Numero de ítems en el conjunto de datos.
- p_u : Los factores del usuario (n_users , $n_factors$).
- q_i : Los factores del artículo (n_items , $n_factors$)
- b_u : Los sesgos del usuario (n_users)
- b_i : Los sesgos del elemento (n_items)

Cada algoritmo es parte del espacio global de Surprise, solo se necesita importar sus nombres desde la librería Surprise, un ejemplo de esto es:

```
from surprise import KNNBasic
algo = KNNBasic()
```

Las líneas de base se pueden calcular de dos maneras:

- Usando el descenso de gradiente estocástico (SGD).
- Uso de mínimos cuadrados alternos (ALS).

Se puede configurar la forma en que se calculan las líneas de base utilizando el código **bsl_options**, el cual es un parámetro para la creación de un algoritmo. Este parámetro es un diccionario para que la clase **method** indique el método a utilizar. Los valores aceptados como mencionamos anteriormente son **ALS** (predeterminado) y **SGD**.

Para ambos procedimientos (**ALS** y **SGD**), los sesgos del usuario y del elemento se inicializan en cero.

Ejemplos de uso:

- Enfoque ALS

```
print('Using ALS')
bsl_options = {'method': 'als',
              'n_epochs': 5,
              'reg_u': 12,
              'reg_i': 5
              }
algo = BaselineOnly(bsl_options=bsl_options)
```

- Enfoque SGD

```
print('Using SGD')
bsl_options = {'method': 'sgd',
              'learning_rate': .00005,
              }
algo = BaselineOnly(bsl_options=bsl_options)
```

Algunas medidas de similitud pueden usar líneas de base, como la similitud **pearson_baseline**.

La configuración funciona de la misma manera que las configuraciones mencionadas anteriormente, ya sea que utilicen las líneas de base con predicción real \hat{r}_{ui} o no.

```
bsl_options = {'method': 'als',
              'n_epochs': 20,
              }
sim_options = {'name': 'pearson_baseline'}
algo = KNNBasic(bsl_options=bsl_options, sim_options=sim_options)
```

Esto nos lleva a la configuración de la medida de similitud que revisaremos a continuación.

La mayoría de los algoritmos utilizan una medida de similitud para estimar una calificación, la forma en que se pueden configurar se realiza de manera similar a las calificaciones de referencia: solo se necesita pasar un parámetro **sim_options** que es un argumento en la creación de un algoritmo. Este argumento es un diccionario con las siguientes claves (todas opcionales):

- **Name:** Es el nombre de la similitud que se utilizara, tal como se define en el módulo **similarities**, El valor predeterminado es 'MSD'.
- **User_based:** Especifica si se calcularan similitudes entre usuarios o elementos.
- **Min_support:** El número mínimo de elementos comunes o el número mínimo de usuarios comunes.
- **Shrinkage:** Parámetro de contracción a aplicar (solo relevante para similitud **Pearson_baseline**). El valor predeterminado es 100.

Algunos ejemplos de uso:

- Similitud de **Cosine**

```
sim_options = {'name': 'cosine',
              'user_based': False # compute similarities between items
              }
algo = KNNBasic(sim_options=sim_options)
```

- Similitud **Pearson_baseline**.

```
sim_options = {'name': 'pearson_baseline',
              'shrinkage': 0 # no shrinkage
              }
algo = KNNBasic(sim_options=sim_options)
```

4 Capítulo 5: Implementación algoritmo

En esta sección se explican las principales estructuras y descripción de las métricas utilizadas para lograr realizar un sistema de recomendación de recetas saludables, desde cómo se procesan los datos, el por qué y cuál es el objetivo de dicha implementación.

4.1 Descripción de implementación y estructura del algoritmo implementado

De acuerdo a las definiciones de filtrado colaborativo y basados en contenidos, el algoritmo al usar información nutricional directamente desde datasets y además, de considerar las preferencias de los usuarios, se puede concluir que la aplicación utiliza un método de recomendación híbrido.

Para que el sistema de recomendación de recetas saludables funcione de manera apropiada, se necesitan 2 parámetros importantes, los cuales son: Preferencias del usuario (Ingredientes) y necesidad del usuario (Kcal máximas de una receta), ambos argumentos son ingresados por el usuario manualmente.

Surprise necesita de un conjunto de datos(dataframe) con una estructura determinada para funcionar adecuadamente, el dataframe debe cumplir con tener datos de la forma: User, Item, Ranking.

El resultado final que debemos mostrar al usuario debe ser el nombre de la receta y su composición nutricional ordenada por las Kcal, si recordamos la estructura que necesita Surprise para funcionar, podemos concluir que nos sería imposible mostrar al usuario, el nombre y la composición nutricional de la receta, es por esto, que un método para solucionar este problema fue basarse en las id de recetas, como trabajamos con diferentes data sets y cada uno tiene información diferente necesitamos un atributo que se repita en todos los conjuntos de datos, es por esta razón que se eligió el id de receta.

Teniendo en cuenta esto, en un principio leemos 2 data sets, cada uno con datos diferentes, pero que tienen el común de tener el mismo atributo id receta, el fin de esta medida es hacer una unión de ambos conjuntos de datos para así poder obtener los nombres y composición nutricional que debemos mostrar al usuario, esto lo hacemos de acuerdo a la **Imagen 4**.

Imagen 4: Lectura y unión de dos conjuntos de datos

```
12 columns = ['user_id', 'recipe_id', 'date', 'rating']
13
14 df = pd.read_csv('raw_interactions.csv', sep=';', names=columns, low_memory=False)
15
16 columns = ['name', 'recipe_id', 'minutes', 'contributor_id', 'submitted', 'tags', 'nutrition', 'steps', 'description',
17            'ingredients', 'n_ingredients']
18
19 recipes = pd.read_csv('RAW_recipes.csv', sep=';', names=columns, encoding='latin-1', low_memory=False)
20
21
22 recipes_names = recipes[['name', 'recipe_id']]
23
24 combined_recipes_data = pd.merge(df, recipes_names)
```

Cada usuario es un universo diferente, por lo tanto, hay que tener en cuenta que cada usuario tiene preferencias y necesidades distintas. Es por esto, que en el proyecto de título, como primer paso después de la lectura y unión de los data sets, es leer un conjunto de datos y contar la cantidad de reviews de todos los usuarios respecto a las recetas que valoró y se seleccionan las que tengan sobre 50 valoraciones. Lo siguiente es indicar la escala de rating, para este caso es de 0 a 5, cero el mínimo y cinco el máximo, después se obtienen los identificadores(**recipe_id**) de las recetas, además una lista de recetas que el usuario a calificado y del conjunto de datos resultante se eliminan dichas recetas calificadas.

Como indica la **Imagen 5**, se crea una lista vacía llamada **my_recs**, esta lista se comienza a llenar con posibles recetas a recomendar para el usuario 1533 por ejemplo, esta predicción se genera a partir de la comparación de las recetas que el usuario calificó con anterioridad y comparando con las calificaciones de recetas de los demás usuarios.

Imagen 5: Código para generar una lista de predicción

```

60     algo = SVDpp()
61     algo.fit(data.build_full_trainset())
62     my_recs = []
63     for iid in recipes_to_predict:
64         my_recs.append((iid, algo.predict(uid=1533,iid=iid).est))

```

Se genera una lista con los datos mostrados en **Imagen 6**.

Imagen 6: Lista con id de recetas a recomendar

	recipe_id	predictions
0	607	4.403196
1	749	4.039257
2	916	4.093482
3	1005	4.615019
4	1356	4.990280
5	2496	3.988652
6	2886	5.000000
7	3180	5.000000
8	3470	4.082593
9	3595	4.643470

Como se puede apreciar en **Imagen 6** generada, solo contiene 2 columnas (**recipe_id**, **predictions**), por esta razón la necesidad de leer y juntar otros dos data sets, con el fin de poder unir todas las columnas que contienen la información necesaria para el usuario. ¿Cuál es el fin de esto?, la respuesta es buscar cada **recipe_id** en **combined_recipes_data** como se muestra en la **Imagen 4**, con el objeto de generar un dataframe solo con las recetas que coincidan con las **recipe_id** mostradas en **Imagen 6**.

Luego de generar la lista con las id de las posibles recetas a recomendar, el sistema solicita un parámetro que debe ser ingresado por el usuario, específicamente un ingrediente que sea de su gusto, en el caso de ejemplo se ingresa el ingrediente azúcar(sugar).

Ingrese un ingrediente:sugar

El ingrediente es comparado por cada fila presente en el conjunto datos, esto con el motivo de filtrar todos los datos y solo dejar las recetas que contengan dicho ingrediente.

Luego de filtrar las recetas y obtener un nuevo data set filtrado por las preferencias del usuario, se procede a filtrar las recetas con los ingredientes que el usuario prefiere mediante un filtro de calorías que el mismo usuario ingresa al sistema, este no puede ser superior a las 2.000 Kcal.

```
Ingrese la cantidad máxima de calorías que quiere en su receta:300
La receta '2' tiene '398.0 calorías'
La receta '3' tiene '165.0 calorías'
La receta '5' tiene '189.0 calorías'
La receta '6' tiene '272.0 calorías'
La receta '7' tiene '92.7 calorías'
```

Finalmente, y pasado por todos los filtros correspondientes, se muestra por pantalla un mensaje al usuario con las recetas recomendadas.

```
De acuerdo al ingrediente 'sugar' te vamos a recomendar las siguientes receta:
      name                                     nutrition
161229  pizza dough and sauce                165.1, 5.0, 8.0, 18.0, 8.0, 2.0, 9.0
69688   dark chocolate cake                  189.1, 10.0, 80.0, 10.0, 5.0, 7.0, 10.0
22168   best banana bread                    272.8, 16.0, 97.0, 14.0, 7.0, 31.0, 14.0
220759  vanishing oatmeal raisin cookies     92.7, 3.0, 33.0, 3.0, 3.0, 2.0, 5.0
```

5 Capítulo 6: Aplicación web para ejecutar algoritmos de recomendación

En esta sección se explica la implementación de la aplicación web para generar recomendaciones de recetas saludables, sus métricas y componentes utilizados.

5.1 Objetivo de la aplicación

El objetivo de implementar una aplicación web para un algoritmo de recomendación, es emular el funcionamiento del algoritmo con usuarios reales, además de verificar si los resultados obtenidos son los correctos, por otra parte, medir las métricas de experimentos y comparar los datos obtenidos para concluir resultados del algoritmo.

5.2 Descripción de la interfaz y casos de uso

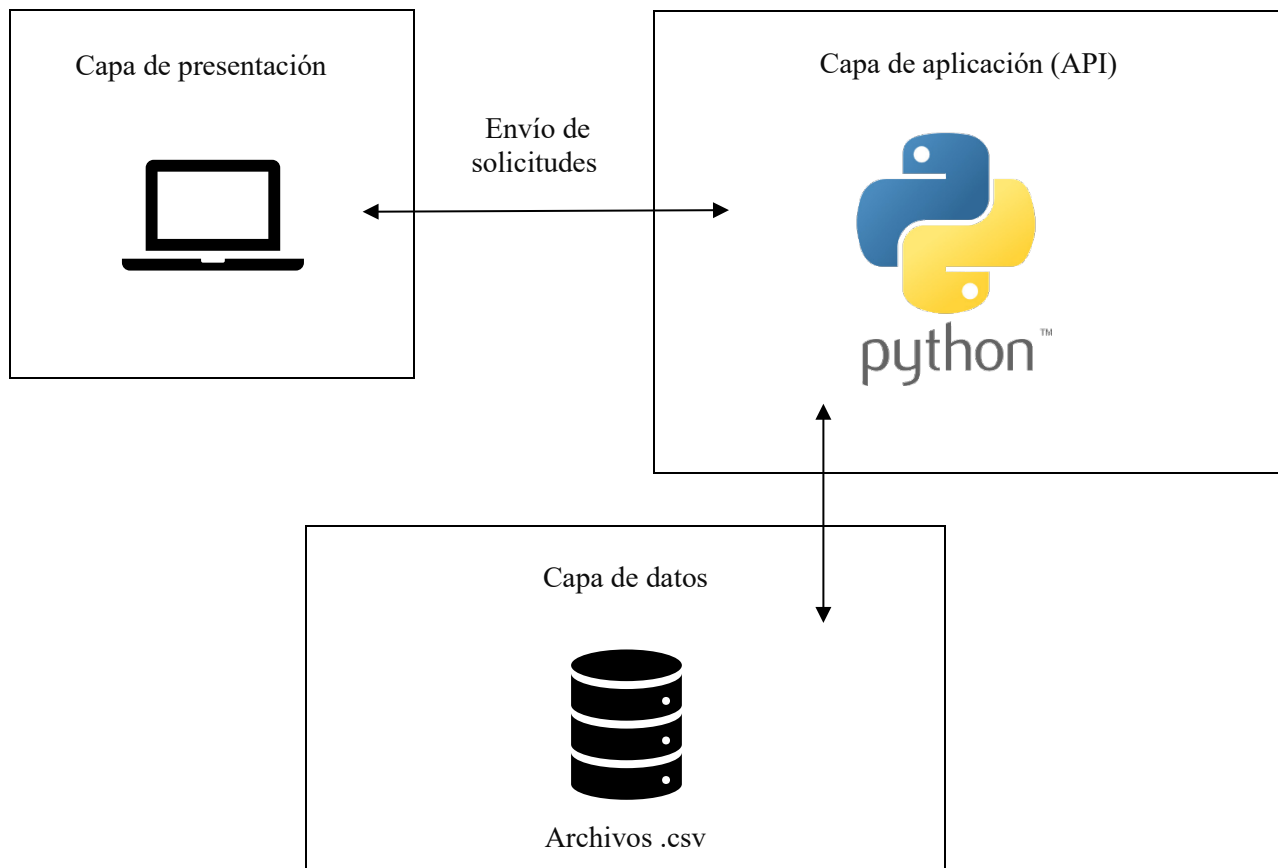


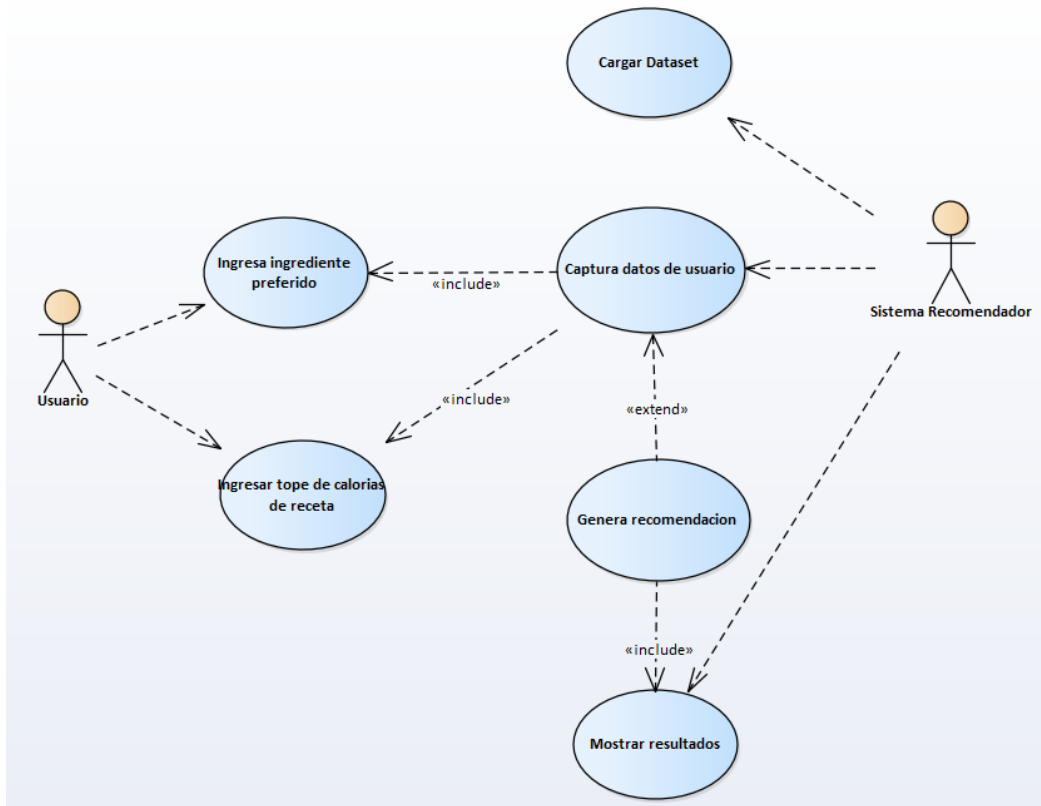
Ilustración 1: Arquitectura de software de aplicación desarrollada

Tabla 7: Descripción de la interfaz

Actor	Nombre	Descripción
Usuario	Ingresar ingrediente	Ingresar un ingrediente al sistema para poder recomendar una receta que contenga dicho ingrediente.
Usuario	Ingresar tope de calorías	Ingresar la cantidad máxima de calorías que desea que tenga la receta a recomendar.
Sistema	Cargar Dataset	Carga los datos para trabajar con los algoritmos de recomendación y generar una predicción de acuerdo a todos los datos de usuarios y recetas presentes en el dataset.
Sistema	Cargar datos de usuario	Lleva los datos ingresados por el usuario al sistema, para que el algoritmo use la información que necesita para funcionar.
Sistema	Generar recomendación	Luego de ejecutar los algoritmos y pasado los filtros con las preferencias del usuario, el sistema genera la recomendación tomando en cuenta los parámetros ingresados por el usuario.
Sistema	Mostrar resultados	El sistema muestra los resultados con las recetas recomendadas.

En la **Imagen 7** se muestra el funcionamiento de la aplicación web, como interactúa el usuario con el sistema de recomendación, los datos que usa el algoritmo y los procesos que sigue el sistema recomendador para finalmente entregar una recomendación al usuario.

Imagen 7: Casos de uso

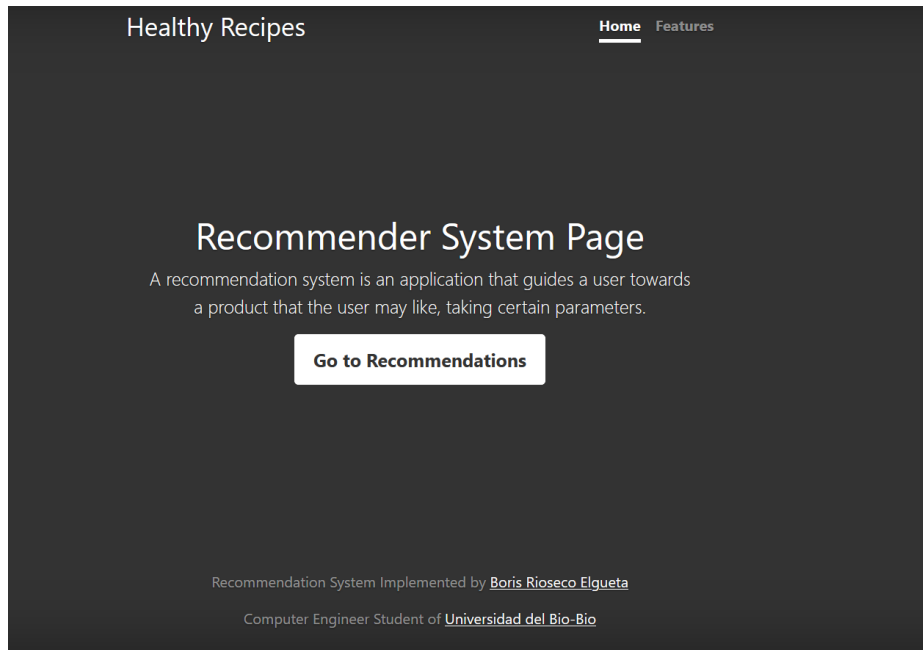


Este modelo de casos de uso representa el funcionamiento de la aplicación web implementada, en primera instancia el usuario ingresa los parámetros necesarios para que el software pueda ejecutar las instrucciones de los algoritmos. La aplicación web por defecto carga un dataset con toda la información de las recetas y ratings. Una vez ingresado los datos de usuario, el software ejecuta el algoritmo de recomendación y muestra por pantalla las recetas recomendadas.

5.3 Evidencias de aplicación implementada

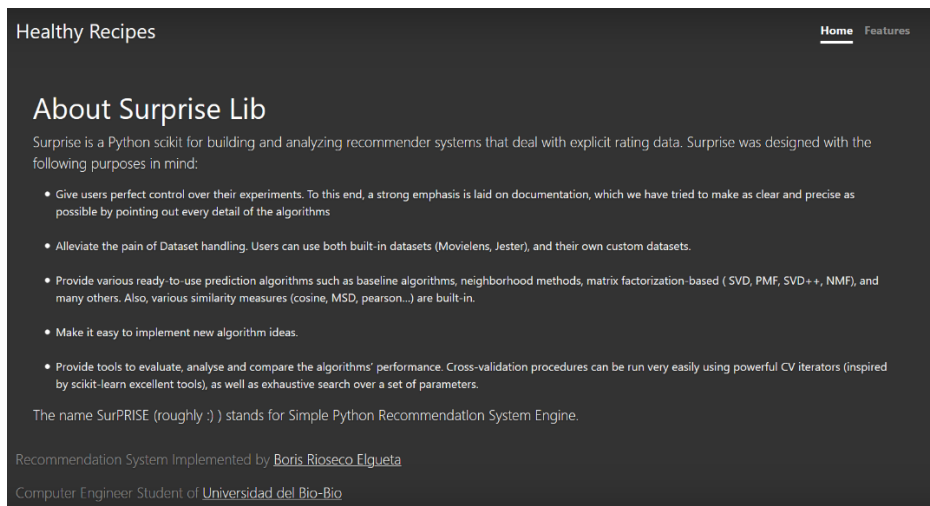
En esta sección se evidenciará la aplicación web implementada para ejecutar los algoritmos de recomendación y además, ejecutar la etapa experimental.

imagen 8: Pantalla de inicio



En la **imagen 8**, se muestra la pantalla que el usuario mira por primera vez.

imagen 9: Características de Surprise



En la **imagen 9** anterior se observa la pantalla que se le muestra al usuario para que aprenda un poco sobre las características de la librería Surprise.

imagen 10: Formulario de recomendación de recetas saludables

Recommender systems of healthy recipes

Make Experiments Only Recommendations

By default, the recommendation system uses the svdpp algorithm and it may take a few minutes to display the results.

User

Ingredient

Calories

Start

En la imagen anterior, se aprecia un formulario que le permite al usuario ejecutar la aplicación solo para que se le muestren las recetas saludables recomendadas.

imagen 11: Formulario para experimentos

Recommender systems of healthy recipes

Make Experiments Only Recommendations

This operation may take several minutes.

Ingredient

Calories

Start

En **imagen 11**, podemos ver que el usuario ha seleccionado realizar experimentos, por lo cual, a diferencia de la **imagen 10**, el formulario del usuario ha cambiado y ahora además de ingresar un ingrediente y un tope de calorías, el usuario debe seleccionar el algoritmo que quiere usar para ejecutar el experimento.

imagen 12: Ejemplo de lista de recetas recomendadas

We Recommend the following recipes

Recipe Name	Ingredients	Details
easy crock pot salsa chicken	boneless skinless chicken breasts", "salsa", "corn", "black beans", "flour tortilla", "sour cream", "guacamole", "lettuce", "tomatoes", "cheese"	Details
mexican ceviche	halibut fillets", "limes", "fresh tomato", "green pepper", "parsley", "salt", "pepper", "oregano", "jalapeno peppers", "white vinegar", "onion", "fresh cilantro", "tabasco sauce", "lettuce leaf", "avocado", "black olives"	Details
the ultimate greek salad	olive oil", "fresh lemon juice", "red wine vinegar", "garlic cloves", "dried oregano", "lettuce", "plum tomatoes", "english cucumber", "red onion", "green pepper", "kalamata olive", "feta cheese"	Details
kittencal s taco salad for a crowd	lean ground beef", "taco seasoning mix", "cheddar cheese", "red onion", "yellow onion", "iceberg lettuce", "tomatoes", "avocados", "black olives", "nacho chips", "catalina dressing"	Details
jalapeno red bean bbq burgers	olive oil", "onion", "plum tomato", "jalapeno chiles", "garlic", "chili powder", "kidney beans", "dry breadcrumbs", "barbecue sauce", "egg white", "whole grain buns", "cheddar cheese", "lettuce", "tomatoes", "avocado"	Details
winter fruit salad with lemon poppy seed dressing	white sugar", "lemon juice", "diced onions", "dijon mustard", "salt", "vegetable oil", "poppy seed", "romaine lettuce", "swiss cheese", "cashews", "dried cranberries", "apple", "pear"	Details
egg salad sandwich	hard-boiled eggs", "onion", "celery seed", "dijon mustard", "lettuce", "mayonnaise", "bread", "butter", "salt and pepper"	Details
kittencal s famous caesar salad	head romaine lettuce", "garlic-flavored croutons", "parmesan cheese", "black pepper", "anchovy fillets", "fresh garlic cloves", "mayonnaise", "half-and-half cream", "lemon juice", "dijon mustard", "salt and pepper", "worcestershire sauce", "buttermilk"	Details

En la **imagen 12**, se muestran las recetas que el sistema de recomendación le recomienda al usuario, esta lista muestra el nombre de las recetas e ingredientes.

imagen 13: Detalles por receta

We Show the following details

Preparation						
place first four ingredients in crock pot", "cook on low 6-8 hours", "30-60 minutes prior to serving , remove chicken , shred and return to crock pot", "to serve , use chicken mixture as filling inside tortillas", "add desired condiments"						
Calories	Fat	Sugar	Sodium	Protein	Saturated fat	Carbohydrates
157.0	3.0	18.0	24.0	30.0	2.0	7.0

Si en la **imagen 12**, hago click en “details”, se muestran dos nuevas tablas que tienen la información nutricional y además la preparación de la receta, como vemos en la **imagen 13**.

imagen 14: Ejemplo de experimento usando el algoritmo SVDpp

The Results of Experiments using algorithm SVDpp are..

Algorithm	Precision	Recall	Accuracy RMSE
SVDpp	0.9312	0.9329	0.8886

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
RMSE(testset)	1.0966	0.9555	1.0178	1.0941	0.9932
MAE(testset)	0.6377	0.5770	0.6052	0.6133	0.6025
MSE(testset)	1.2026	0.9130	1.0359	1.1969	0.9864
Fit time	0.1080	0.1080	0.1080	0.1070	0.1070
Test time	0.0020	0.0020	0.0020	0.0020	0.0020

En la **imagen 14**, se aprecian los resultados del experimento, luego de que el usuario eligió el algoritmo SVDpp para ejecutar la aplicación.

6 Capítulo 7: Experimentación y resultados

En el capítulo de experimentación y resultados, se realizarán experimentos con algunos algoritmos disponibles de la librería Surprise, para llevar a cabo esta etapa, se tendrá en cuenta el enfoque trainset, testset split y las métricas presentadas a continuación.

6.1 Descripción de Datasets

El conjunto de datos se compone principalmente de 2 datasets que manejan los mismos datos, el primero llamado INTERACTIONS tiene datos de interacción del usuario con las recetas, es decir, calificaciones por ítems. Este conjunto tiene 1.125.284 valores únicos, está conformado por 3 columnas y el rango de calificaciones para las recetas van desde 0 a 5.

El segundo dataset tiene por nombre RAW_RECIPES, contiene información sobre los nombres de las recetas, ingredientes por receta, preparación, minutos de preparación, quienes calificaron la receta, su información nutricional, tags, número de pasos del menú, fecha de calificación, una breve descripción y la cantidad de ingredientes que componen la receta, este conjunto tiene 230.168 datos únicos y se compone de 12 columnas.

Ambos datasets trabajan con un conjunto de datos en inglés, por lo tanto, la información de entrada y de salida tendrán este idioma, el formato de los datasets es .CSV separando sus celdas por punto y coma en ambos casos. Datasets utilizados¹.

¹ Datasets utilizados para algoritmos de recomendación, disponibles en:
https://www.kaggle.com/shuangli94/food-com-recipes-and-user-interactions?select=PP_users.csv

6.2 Métricas para evaluar resultados

Para que el sistema de recomendación trabaje en igualdad de condiciones para cada algoritmo, se realizaron 4 iteraciones con diferentes parámetros, cada iteración conservara los parámetros hasta ejecutar el ultimo algoritmo de la lista de algoritmos disponibles para los experimentos. Por ej. Para la primera evaluación de los algoritmos, los datos de entrada serán para el ingrediente “lettuce” y para la cantidad máxima de calorías “300 Kcal”, estos parámetros permanecerán constantes hasta que todos los algoritmos en la lista sean ejecutados. Luego se pasa a la siguiente evaluación con parámetros distintos, donde nuevamente se conservarán los datos hasta que se ejecuten todos los algoritmos y así, hasta terminar las evaluaciones.

Se van a utilizar las siguientes métricas de evaluación de resultados:

- Precision: Es una medida de cuan relevante es la recomendación.

	$\text{Precision@k} = \frac{ \{\text{Ítems recomendados que son relevantes}\} }{ \{\text{Ítems recomendados}\} }$	
--	---	--

- Recall: Es una medida de cuantos resultados verdaderamente relevantes se devuelven.

	$\text{Recall@k} = \frac{ \{\text{Ítems recomendados que son relevantes}\} }{ \{\text{Ítems relevantes}\} }$	
--	--	--

Una recomendación se considera relevante si su calificación r_{ui} es mayor que un umbral dado, para este experimento el umbral será de recetas que tengan una valoración de usuario (rating) mayor o igual a 3.5, todas las recetas que sean mayores que el umbral dado, serán consideradas como elemento recomendado.

6.3 Metodologías de evaluación

6.3.1 Metodología Train-test set split

La metodología train-test set split, permite dividir nuestro conjunto de datos en 2 subconjuntos, normalmente se llaman “Conjunto de entrenamiento” y “Conjunto de pruebas”, respectivamente, el conjunto de entrenamiento tiene la mayor parte del conjunto de datos para entrenar el algoritmo. En contraparte, el conjunto de pruebas tiene una porción muy menor del total de datos presentes en el dataset, estos datos son usados para evaluar el modelo entrenado. Cabe destacar que este método es recomendado siempre y cuando el conjunto de datos tenga una gran cantidad de información.

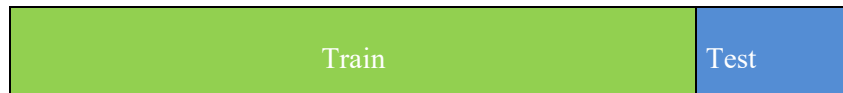


Imagen 15: Train-test set split

La manera para realizar la división en ambos subconjuntos se hace mediante la siguiente instrucción:

```
trainset, testset = train_test_split (data_r, test_size=.25)
```

Ilustración 2: Función train-test set split

Esto significa que nuestro conjunto de datos “data_r” quedara dividido en un 75% para conjunto de entrenamiento y un 25% para conjunto de pruebas, como se muestra en **Imagen 8**.

6.3.2 Metodología K-Fold Cross-Validation

La validación cruzada es una metodología que sirve para estimar la habilidad que tiene el modelo para predecir.

Para el proyecto de título nos enfocaremos en la validación cruzada de k-iteraciones. Este método consiste básicamente en dividir el conjunto entero de datos en k subconjuntos, uno de los subconjuntos divididos se utiliza como datos de prueba y los otros como datos de entrenamiento, este proceso se repite k veces. Para efectos de este proyecto, seleccionamos un valor arbitrario $k = 5$, este valor será único y constante durante toda la fase experimental.

El método cross-validation se puede ilustrar de la siguiente manera:

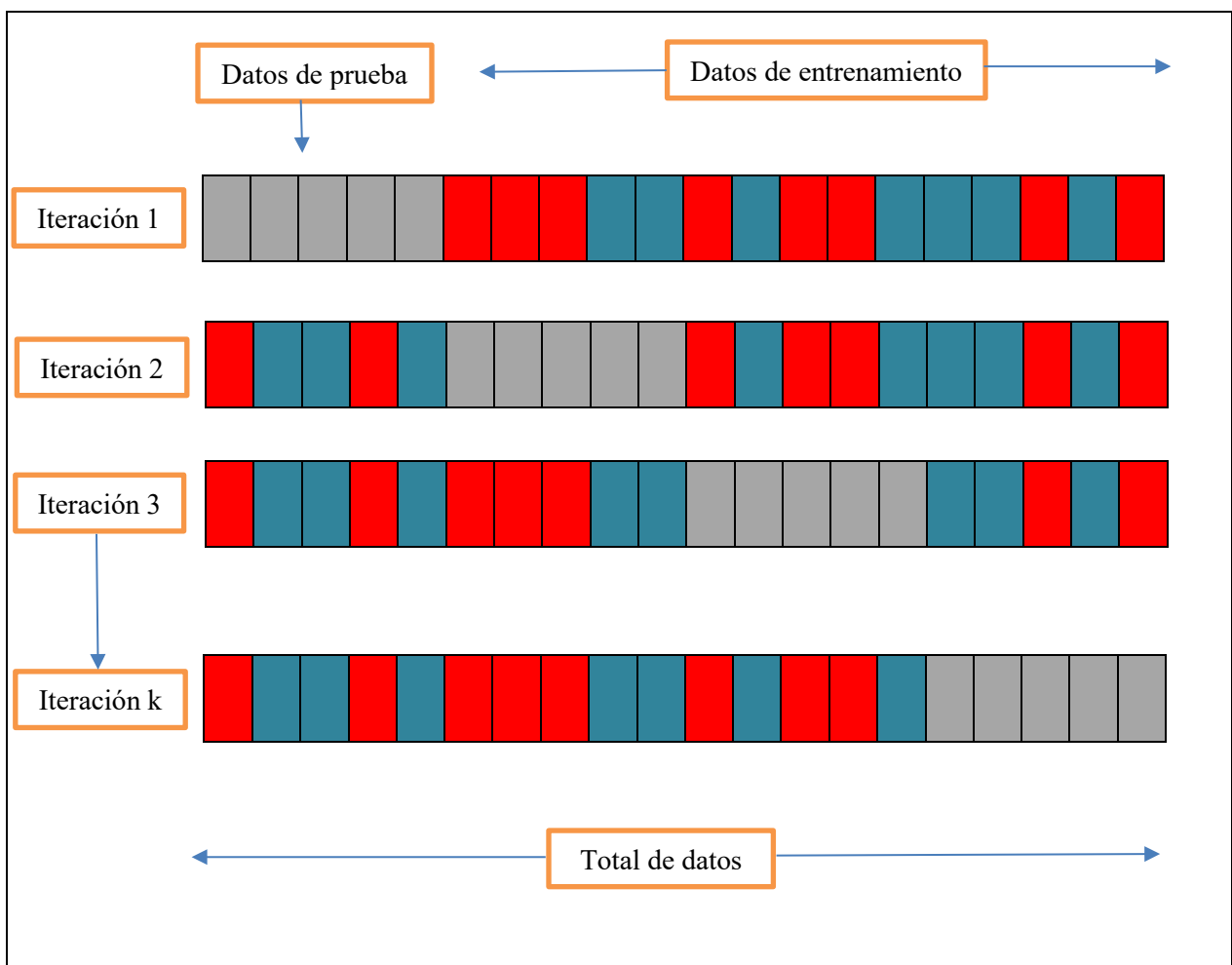


Ilustración 3: Validación cruzada de k iteraciones

A esta metodología de evaluación le podemos entregar métricas de precisión, estas métricas se utilizan principalmente para evaluar las tasas de error de predicción y rendimiento.

- RMSE (Root mean squared error): Mide el error de predicción promedio realizado por el modelo. Es decir, la diferencia promedio entre los valores de resultado conocidos observados y los valores predichos por el modelo. Cuanto menor sea el RMSE, mejor será el modelo.
- MAE (Mean absolute error): Es una alternativa al RMSE, que es menos sensible a los valores distintos, corresponde a la diferencia absoluta promedio entre los resultados observados y pronosticados, cuanto menor sea el mae, mejor será el modelo
- MSE (Mean squared error): Representa la diferencia entre los valores originales y predichos extraídos por el promedio al cuadrado sobre el conjunto de datos, mientras menor sea, mejor será el modelo.

6.4 Algoritmos usados para etapa experimental

Para la etapa de experimentación se usarán algoritmos de factorización matricial, específicamente los algoritmos SVD, SVDPP, también el algoritmo de vecinos más cercanos KNNBasic con similitud de coseno y basado en ítems, por otra parte, dos algoritmos de filtrado colaborativo tales como SlopeOne y CoClustering. Manteniendo el mismo conjunto de datos para cada uno de los algoritmos.

El sistema de recomendación implementado para la etapa experimental, toma tres parámetros, por ejemplo: Quiero obtener los datos del experimento usando el algoritmo SlopeOne, de una receta que contenga el ingrediente chocolate, pero que el máximo de calorías sea de 300 Kcal.

La aplicación web, para la etapa de experimentos consta de 3 inputs, uno para el ingrediente y otro para la cantidad de calorías, el otro input es una lista desplegable que permite al usuario que va a realizar el experimento seleccionar el algoritmo que desea utilizar. Una vez seleccionado los 3 parámetros necesarios, el algoritmo implementado comienza su tarea para la realizar el experimento.

6.5 Protocolos de prueba

6.5.1 Objetivo general

El objetivo de este proceso es verificar si las recomendaciones entregadas al usuario son relevantes, además de verificar la habilidad que tienen los algoritmos para predecir recetas y las tasas de error que presentan los algoritmos al realizar una recomendación.

6.5.2 Condiciones de prueba

El sistema de recomendación para su etapa experimental tendrá las siguientes condiciones:

Experimento 1

Algoritmo	Ingrediente	Calorías tope
SVD	Lettuce	300 kcal
SVDpp	Lettuce	300 kcal
KNNBasic	Lettuce	300 kcal
CoClustering	Lettuce	300 kcal
SlopeOne	Lettuce	300 kcal

Experimento 2

Algoritmo	Ingrediente	Calorías tope
SVD	Onion	600 kcal
SVDpp	Onion	600 kcal
KNNBasic	Onion	600 kcal
CoClustering	Onion	600 kcal
SlopeOne	Onion	600 kcal

Experimento 3

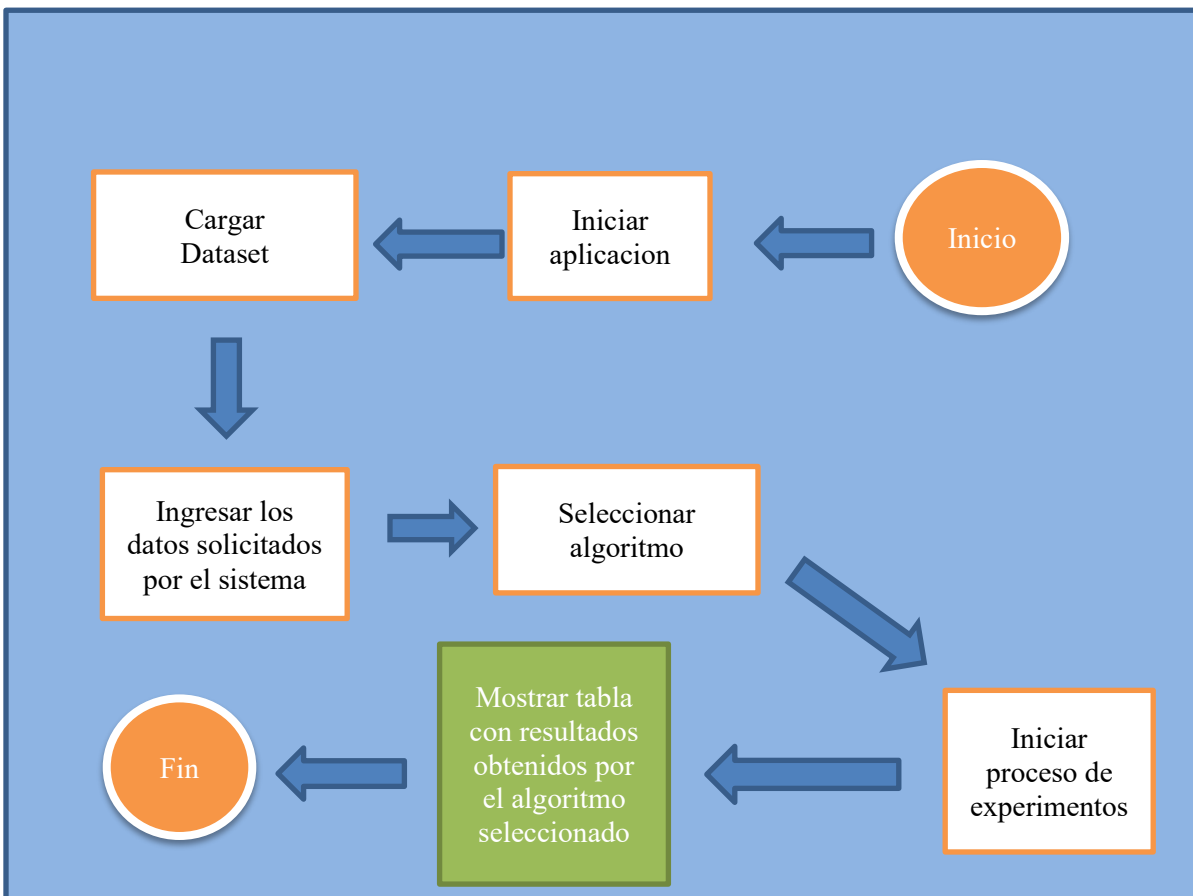
Algoritmo	Ingrediente	Calorías tope
SVD	Cheese	900 kcal
SVDpp	Cheese	900 kcal
KNNBasic	Cheese	900 kcal
CoClustering	Cheese	900 kcal
SlopeOne	Cheese	900 kcal

Experimento 4

Algoritmo	Ingrediente	Calorías tope
SVD	Chocolate	1200 kcal
SVDpp	Chocolate	1200 kcal
KNNBasic	Chocolate	1200 kcal
CoClustering	Chocolate	1200 kcal
SlopeOne	Chocolate	1200 kcal

6.5.3 Procedimiento de la etapa experimental

El esquema que se muestra a continuación, explica el procedimiento del software para el proceso de experimentos, principalmente, para que el software funcione deben procesarse los siguientes datos, primero: Cargar dos datasets (El software lo hace automáticamente), uno con información de ratings de usuario por ítem y el otro con información de recetas. Luego el usuario ingresa los datos solicitados en el formulario (Ver **imagen 11**) y luego debe elegir el algoritmo al cual le va a hacer los experimentos y apretar el botón para iniciar el proceso experimental.



6.6 Datos obtenidos

En esta sección tendremos como objetivo mostrar los resultados de los experimentos, tomando en cuenta los valores descritos en el sub capítulo **Condiciones de prueba**. Para los datos obtenidos, los valores resaltados en **negrita** identificaran a los mejores resultados por columna, por el contrario, los resultados marcados en **rojo** indican el peor resultado por columna.

Tabla 8: Primera experimentación, ingrediente lettuce con calorías máximas de 300 Kcal

Algoritmo	Precision	Recall	RMSE (Accuracy)	RMSE(Mean)	MSE(Mean)	MAE(Mean)
SVDpp	0.9515	0.9552	0.7198	1.0129	1.0615	0.6338
SVD	0.8897	0.8897	0.8793	1.0163	1.0781	0.6419
KNNBasic	0.9118	0.9118	1.2093	1.0534	1.1135	0.6541
CoClustering	0.9343	0.9343	0.8817	1.0554	1.1273	0.6609
SlopeOne	0.8467	0.8467	1.346	1.0533	1.1382	0.6665

Tabla 9: Segunda experimentación, ingrediente onion con calorías máximas de 600 Kcal

Algoritmo	Precision	Recall	RMSE (Accuracy)	RMSE(Mean)	MSE(Mean)	MAE(Mean)
SVDpp	0.8493	0.8475	1.2727	1.2511	1.5653	0.7398
SVD	0.8244	0.8192	1.2505	1.2445	1.5491	0.7902
KNNBasic	0.8383	0.8395	1.3106	1.3153	1.7302	0.7477
CoClustering	0.8216	0.8212	1.3428	1.3482	1.8189	0.7690
SlopeOne	0.8197	0.816	1.3554	1.3211	1.7454	0.7709

Tabla 10: Tercera experimentación, ingrediente Cheese con calorías máximas de 900 Kcal

Algoritmo	Precision	Recall	RMSE (Accuracy)	RMSE(Mean)	MSE(Mean)	MAE(Mean)
SVDpp	0.865	0.8652	1.2139	1.2291	1.5110	0.7214
SVD	0.8391	0.8357	1.2215	1.2233	1.4969	0.7731
KNNBasic	0.8377	0.8387	1.3105	1.2864	1.6552	0.7270
CoClustering	0.8259	0.823	1.2939	1.3013	1.6955	0.7390
SlopeOne	0.8292	0.8236	1.2897	1.2958	1.6792	0.7520

Tabla 11: Ultima experimentación, ingrediente Chocolate con calorías máximas de 1200 Kcal

Algoritmo	Precision	Recall	RMSE (Accuracy)	RMSE(Mean)	MSE(Mean)	MAE(Mean)
SVDpp	0.8874	0.8903	1.2304	1.2470	1.5576	0.7253
SVD	0.8589	0.8598	1.3021	1.2405	1.5393	0.7747
KNNBasic	0.87	0.8723	1.3202	1.3040	1.7018	0.7478
CoClustering	0.859	0.8607	1.303	1.3126	1.7238	0.7553
SlopeOne	0.8559	0.8567	1.3044	1.3225	1.7504	0.7769

6.7 Análisis de resultados

De los resultados obtenidos, podemos concluir que en las 4 iteraciones el algoritmo implementando SVDpp obtiene los mejores resultados de precisión y recall, esto quiere decir que dentro de todos los algoritmos, el algoritmo SVDpp presenta recomendaciones más relevantes para los usuarios.

Por el contrario, los algoritmos SlopeOne y CoClustering, presentan los valores más bajos del modelo de recomendación, por lo tanto, los resultados obtenidos por estos dos algoritmos, no serán relevantes para el usuario.

Al evaluar los resultados de RMSE accuracy, RMSE promedio, MAE promedio y MSE promedio entre las 4 etapas de experimentos. Como vimos anteriormente, mientras menor sea el valor dado, mejor será el modelo de recomendación, por consiguiente, en la **Tabla 8** el algoritmo SVDpp presenta el mejor resultado (menor) para RMSE y MSE, pero en las tablas restantes, el algoritmo SVD tiene mejor rendimiento en ambas columnas, por lo cual, para esta ocasión, el mejor algoritmo recomendador es SVD.

En contraparte, nuevamente los algoritmos SlopeOne y CoClustering, muestran para ambas ocasiones valores que representan un mal rendimiento a la hora de recomendar recetas al usuario.

Como último punto de análisis, queda analizar el MAE promedio, y para este caso, en las cuatro iteraciones el algoritmo SVDpp, presenta los valores más pequeños, seguidos de SVD y KNNBasic, con esto podemos decir que el algoritmo SVDpp es mejor modelo para predecir recetas saludables, en lo que al MAE concierne.

7 Conclusión

A lo largo de este proyecto, podemos concluir que los sistemas de recomendación se hacen cada vez más populares, tanto comercialmente como en la comunidad de investigación, donde se han sugerido muchos enfoques para proponer recomendaciones.

Los sistemas de recomendación abren nuevas oportunidades para recuperar información personalizada en internet. También ayuda a aliviar el problema de la sobrecarga de información, que es un fenómeno cada vez más común en los sistemas de información y permite a los usuarios tener acceso a productos y servicios que no están fácilmente disponibles para su uso.

Como pudimos evidenciar en la subsección [Descripción de Datasets](#), damos cuenta que se logró alcanzar los objetivos de recopilación de recetas, composición nutricional y las respectivas calificaciones de los usuarios. En el [Capítulo 6: Aplicación web para ejecutar algoritmos de recomendación](#), dejamos en evidencia, la implementación del algoritmo, el cual entrega al usuario los resultados que como objetivo fueron planteados al principio de este proyecto.

A modo de conclusion de la etapa experimental, podemos llegar a concluir, luego de haber analizado las cuatro tablas, que los algoritmos de factorizacion matricial, SVD y SVDpp tienen el mejor rendimiento y presentan resultados mas relevantes para los usuarios.

El algoritmo que usa el metodo de vecinos mas cercanos, KNNBasic, se mantuvo en el promedio de resultados. Se puede decir que es un algoritmo de recomendación neutro, ya que, en las cuatro iteraciones siempre se mantuvo en el medio.

Al contrario de los algoritmos de factorizacion matricial, los algoritmos de filtrado colaborativo SlopeOne y CoClustering, se quedan atrás al momento de entregar un resultado relevante al usuario, es probable que al implementar estos dos metodos, al usuario tal vez no le guste la recomendación.

En este proyecto de titulo, hubiese sido ideal poder haber comparado todos los algoritmos disponibles en la librería surprise, para así, concluir de alguna manera, cual es el algoritmo que por lo general tiene mejor rendimiento, como por ejemplo, la velocidad con la cual el sistema recomienda una receta o tambien el error o precision de los resultados. No se pudieron utilizar algunos algoritmos porque la cantidad de datos presentes en los datasets, mas la cantidad de recursos que exigen esos algoritmos hizo que la cantidad de memoria RAM y disco duro necesario para la ejecución fueran extremadamente grandes, por lo que al intentar usar estos algoritmos daban error de memoria.

Como trabajo futuro, seria bueno continuar este proyecto de titulo, pero ademas de considerar las preferencias del usuario y las necesidades nutricionales, considerar la condicion de salud del usuario (Por ej. Enfermedad o discapacidad), para así abordar los problemas de salud sin perder el foco de los gustos que cada usuario tiene.

8 Referencias

- Avery, C., & Zeckhauser, R. (1997). Recommender systems for evaluating computer messages. *Communications of the ACM*, 40(3), 88–89. <https://doi.org/10.1145/245108.245127>
- Bianchini, D., De Antonellis, V., De Franceschi, N., & Melchiori, M. (2017). PREFer: A prescription-based food recommender system. *Computer Standards and Interfaces*, 54, 64–75. <https://doi.org/10.1016/j.csi.2016.10.010>
- Burke, R., Felfernig, A., & Göker, M. H. (2011). Recommender systems: An overview. *AI Magazine*, 32(3), 13–18. <https://doi.org/10.1609/aimag.v32i3.2361>
- Cui, Q., Wu, S., Huang, Y., & Wang, L. (2017). *A Hierarchical Contextual Attention-based GRU Network for Sequential Recommendation*. <http://arxiv.org/abs/1711.05114>
- Ekstrand, M., Riedl, J., & Konstan, J. (2011). Collaborative Filtering Recommender Systems. *Foundations and Trends® in Human–Computer Interaction*, 4(2), 81–173. <https://doi.org/10.1561/1100000009>
- El-Dosuky, M., Rashad, M., Hamza, T., & El-Bassiouny, A. (2012). Food Recommendation using Ontology and Heuristics. *International Conference on Advanced Machine Learning Technologies and Applications*, 423–429.
- Elahi, M., Ge, M., Ricci, F., Fernández-Tobías, I., Berkovsky, S., & David, M. (2015). *Interaction Design in a Mobile Food Recommender System*.
- Elswailer, D., Harvey, M., Ludwig, B., & Said, A. (2015). *Bringing the “healthy” into Food Recommenders*.
- Felfernig, A., Teppan, E., & Gula, B. (2007). KNOWLEDGE-BASED RECOMMENDER TECHNOLOGIES FOR MARKETING AND SALES. In *International Journal of Pattern Recognition and Artificial Intelligence* (Vol. 21, Issue 2). www.worldscientific.com
- Freyne, J., & Berkovsky, S. (2010). Intelligent food planning: Personalized recipe recommendation. *International Conference on Intelligent User Interfaces, Proceedings IUI*, 321–324. <https://doi.org/10.1145/1719970.1720021>
- Gao, X., Feng, F., He, X., Huang, H., Guan, X., Feng, C., Ming, Z., & Chua, T. S. (2020). Hierarchical Attention Network for Visually-Aware Food Recommendation. *IEEE Transactions on Multimedia*, 22(6), 1647–1659. <https://doi.org/10.1109/TMM.2019.2945180>
- Ge, M., Elahi, M., Fernández-Tobías, I., Ricci, F., & Massimo, D. (2015). Using tags and latent factors in a food recommender system. *ACM International Conference Proceeding Series, 2015-May*, 105–112. <https://doi.org/10.1145/2750511.2750528>
- Ge, M., Ricci, F., & Massimo, D. (2015). Health-aware food recommender system. *RecSys 2015 - Proceedings of the 9th ACM Conference on Recommender Systems*, 333–334.

<https://doi.org/10.1145/2792838.2796554>

- González, A. (2014). *Ingeniería de Software : Metodologías Algunas Metodologías*.
- Harvey, M., & Elsweler, D. (2015). *Automated Recommendation of Healthy, Personalised Meal Plans*. <http://nrl.northumbria.ac.uk/23215/>
- Kuo, F.-F., Li, C.-T., Shan, M.-K., & Lee, S.-Y. (2012). *Intelligent Menu Planning: Recommending Set of Recipes by Ingredients*.
- Lee, H. I., Choi, I. Y., Moon, H. S., & Kim, J. K. (2020). A multi-period product recommender system in online food market based on recurrent neural networks. *Sustainability (Switzerland)*, 12(3). <https://doi.org/10.3390/su12030969>
- Masthoff, J. (2011). *Group Recommender Systems: Combining Individual Models*. https://doi.org/10.1007/978-0-387-85820-3_21
- Mika, S. (2011). Challenges for Nutrition Recommender Systems. *Challenges for Nutrition Recommender Systems*, 25–33. www.nhs.uk/livewell/healthy-
- Pazzani, M. J., & Billsus, D. (2007). Content-based recommendation systems. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 4321 LNCS, 325–341. https://doi.org/10.1007/978-3-540-72079-9_10
- Peters, D. P. C., Savoy, H. M., Ramirez, G. A., & Huang, H. (2020). AI Recommender System with ML for Agricultural Research. *IT Professional*, 22(3), 30–32. <https://doi.org/10.1109/MITP.2020.2986125>
- Ricci, F. (2010). *Mobile Recommender Systems Create new project “ONE Project” View project Music recommender systems View project Mobile Recommender Systems*. <https://doi.org/10.3727/109830511X12978702284390>
- Robertson, A., Tirado, C., Lobstein, T., Knai, C., Jensen, J., Ferro-Luzzi, A., & James, W. (2004). *Food and Health in Europe: A New Basis for Action (European Series No 96)*.
- Roza, A. M., & Shizgal, H. M. (1984). The Harris Benedict equation reevaluated: Resting energy requirements and the body cell mass. *American Journal of Clinical Nutrition*, 40(1), 168–182. <https://doi.org/10.1093/ajcn/40.1.168>
- Svensson, M., Laaksolahti, J., Hook, K., & Waern, A. (2000). Recipe based on-line food store. *International Conference on Intelligent User Interfaces, Proceedings IUI*, 260–263. <https://doi.org/10.1145/325737.325866>
- Trang Tran, N., Atas, M., Felfernig, A., & Stettinger, M. (2018). An overview of recommender systems in the healthy food domain. *Journal of Intelligent Information Systems*, 50(3), 501–526. <https://doi.org/10.1007/s10844-017-0469-0>
- Ueta, T., Iwakami, M., & Ito, T. (2011a). A recipe recommendation system based on automatic nutrition information extraction. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 7091 LNAI, 79–90. https://doi.org/10.1007/978-3-642-25975-3_8

- Ueta, T., Iwakami, M., & Ito, T. (2011b). Implementation of a goal-oriented recipe recommendation system providing nutrition information. *Proceedings - 2011 Conference on Technologies and Applications of Artificial Intelligence, TAAI 2011*, 183–188. <https://doi.org/10.1109/TAAI.2011.39>
- van Pinxteren, Y., Geleijnse, G., & Kamsteeg, P. (2011). Deriving a recipe similarity measure for recommending healthful meals. *Proceedings of the 15th International Conference on Intelligent User Interfaces - IUI '11*, 105. <https://doi.org/10.1145/1943403.1943422>
- Yang, L., Hsieh, C., Pollak, J. P., Dell, N., Belongie, S., Cole, C., Estrin, D., Hsieh, C., Yang, H., Belongie, S., Estrin, D., Hsieh, C.-K., & Yang, H. (2017). Yum-me: A Personalized Nutrient-based Meal Recommender System. In *ACM Transactions on Information Systems* (Vol. 9, Issue 4).