

# Optimización de la asistencia al cliente en “iBuyFlowers”: Un enfoque basado en modelos de recomendación

3 de Junio 2024



**Universidad**  
Internacional  
de Valencia

Titulación:

Máster Universitario en Big  
Data y Ciencia de Datos

Curso académico

2023 – 2024

Alumno/a:

Zuleta Mejia, Martin Jose

D.N.I: 1234091960

Director/a de TFM:

Junior Altamiranda

Convocatoria:

Tercera

De:

 Planeta Formación y Universidades

## Índice

Índice de Ilustraciones .....	4
Índice de Tablas .....	7
Resumen .....	8
Abstract .....	9
1. Introducción .....	10
1.1. Estructura del documento .....	13
2. Objetivos .....	14
2.1. Objetivo General .....	14
2.2. Objetivos Específicos .....	14
3. Estado del Arte y Marco teórico .....	15
3.1. Estado del arte .....	15
3.2. Marco teórico .....	19
3.2.1. Sistema de Recomendación .....	19
3.2.2. Filtrado Colaborativo .....	21
3.2.3. Similitud del Coseno .....	22
3.2.4. SBXCloud .....	23
3.2.5. Mínimos cuadrados alternos .....	23
4. Desarrollo del proyecto y resultados .....	24
4.1. Metodología .....	24
4.1.1. Comprensión del negocio .....	25
4.1.2. Comprensión de los datos .....	25
4.1.3. Preparación de los datos .....	25
4.1.4. Modelado .....	25
4.1.5. Evaluación .....	26
4.1.6. Implementación o Despliegue .....	26
4.2. Planteamiento del problema .....	26
4.3. Desarrollo del proyecto .....	28
4.4. Resultados .....	52
4.5. Evaluación del modelo .....	56
5. Conclusión y trabajos futuros .....	58
5.1. Conclusión .....	58
5.2. Trabajos futuros .....	60

Apéndice: Código del Proyecto .....	61
Referencias .....	70

# Índice de Ilustraciones

Ilustración 1. Descripción grafica de usuarios con gustos similares en productos. Fuente: (Munkholm et al, 2024).....	21
Ilustración 2. Fases del modelo de referencia de CRISP-DM. Fuente: (Chapman, et al., 2000) .....	24
Ilustración 3. Secuencia del proceso de compra de un usuario en iBuyFloers. Fuente: Elaboración propia.....	30
Ilustración 4. Previsualización de un fragmento de la tabla Purchase con algunas propiedades e información sensible oculta. Fuente: Elaboración propia .....	31
Ilustración 5. Previsualización de la tabla State con algunas de sus propiedades. Fuente: Elaboración propia.....	32
Ilustración 6. Previsualización de un fragmento de la tabla Cart_box con algunas propiedades. Fuente: Elaboración propia.....	33
Ilustración 7. Previsualización de un fragmento de la tabla Cart_box_item con algunas propiedades. Fuente: Elaboración propia.....	34
Ilustración 8. Previsualización de un fragmento de la tabla Customer con algunas propiedades e información sensible oculta. Fuente: Elaboración propia .....	35
Ilustración 9. Previsualización de un fragmento de la tabla variety con algunas propiedades. Fuente: Elaboración propia.....	36
Ilustración 10. Previsualización de un fragmento de la tabla product_group con algunas propiedades e información sensible oculta. Fuente: Elaboración propia .....	37
Ilustración 11. Previsualización de un fragmento dataset "all_2023_ibf_data.csv "con algunas propiedades e información sensible oculta. Fuente: Elaboración propia.....	38
Ilustración 12. Previsualización de un fragmento del conjunto de datos con las características seleccionadas. Fuente: Elaboración propia. ....	39
Ilustración 13. Porcentaje de datos faltantes en el conjunto de datos por columna. Fuente: Elaboración propia. ....	41
Ilustración 14. Porcentaje de datos faltantes en el conjunto de datos por columna después de la imputación por columna. Fuente: Elaboración propia.....	41
Ilustración 15. Resultado del agrupamiento del conjunto de datos. Fuente: Elaboración propia.....	42

Ilustración 16. Previsualización de un fragmento del conjunto de datos con las frecuencias por compras por usuario e información sensible oculta. Fuente: Elaboración propia.....	43
Ilustración 17. Previsualización de un fragmento del conjunto de datos con las frecuencias por compras por usuario con los ID de referencia para el usuario y producto con información sensible oculta. Fuente: Elaboración propia .....	43
Ilustración 18. Conjunto de datos que contiene las recomendaciones a un cliente con sus puntajes. Fuente: Elaboración propia .....	45
Ilustración 19. Cálculo de ejemplo de la similitud entre dos usuarios representado en el plano y mostrando el ángulo entre ellos. Fuente: (Munkholm et al., 2024).....	47
Ilustración 20. Fragmento del conjunto de datos luego de la inserción del nuevo usuario visualizado al final de los datos. Fuente: Elaboración propia.....	48
Ilustración 21. Visualización parcial del conjunto de datos luego de la aplicación de la función <code>get_dummies</code> . Fuente: Elaboración propia .....	49
Ilustración 22. Visualización parcial del conjunto de datos luego de la aplicación de la similitud de coseno con 100 Datos. Fuente: Elaboración propia.....	51
Ilustración 23. Previsualización de un fragmento del conjunto de datos resultante, conformada por nombre del producto y previsualización de la imagen. Fuente: Elaboración propia.....	52
Ilustración 24. Valores de similitud con índice conformado por los ID de los usuarios y sus valores. Fuente: Elaboración propia .....	53
Ilustración 25. Tabla conformada por Usuario objetivo y el usuario más similar a el. Fuente: Elaboración propia .....	53
Ilustración 26. Previsualización de un fragmento del conjunto de datos resultante para el nuevo usuario, conformada por nombre del producto y previsualización de la imagen. Fuente: Elaboración propia .....	56
Ilustración 27. Importación de las librerías a utilizar. Fuente: Elaboración propia. ....	61
Ilustración 28. Código para la importación de los datos. Fuente: Elaboración propia..	61
Ilustración 29. Previsualización del código para la selección de columnas. Fuente: Elaboración propia.....	62
Ilustración 30. Código para realizar una copia del conjunto de datos anterior y renombrar las columnas. Fuente: Elaboración propia. ....	62
Ilustración 31. Código para verificar el porcentaje de valores null en las filas por cada columna. Fuente: Elaboración propia.....	63

Ilustración 32. Código para realizar la asignación de valores por defectos en las filas cuyas columnas tienen datos faltantes. Fuente: Elaboración propia. ....	63
Ilustración 33. Código para realizar el agrupamiento de los usuarios basados en la frecuencia de compra por producto. Fuente: Elaboración propia.....	63
Ilustración 34. Código para realizar la asignación de columnas con id numéricas para los productos y usuarios. Fuente: Elaboración propia.....	64
Ilustración 35. Código para realizar la inicialización del modelo. Fuente: Elaboración propia.....	64
Ilustración 36. Código para realizar la validación del modelo y obtención de los mejores hiper parámetros. Fuente: Elaboración propia. ....	65
Ilustración 37. Código para el entrenamiento de los datos. Fuente: Elaboración propia. ....	65
Ilustración 38. Código para la generación de recomendaciones de los datos. Fuente: Elaboración propia.....	66
Ilustración 39. Resultado al aplicar el Mean Average Precision. Fuente: Elaboración propia.....	66
Ilustración 40. Código para realizar el agrupamiento de los usuarios basados en sus características. Fuente: Elaboración propia. ....	67
Ilustración 41. Código la creación de un nuevo dataset que contiene al nuevo usuario registrado, tratamiento de datos faltantes y concatenación con el conjunto de datos existentes. Fuente: Elaboración propia. ....	67
Ilustración 41. Previsualización del código para convertir valores categóricos en numéricos. Fuente: Elaboración propia.....	68
Ilustración 42. Código para la aplicación de la similitud del coseno. Fuente: Elaboración propia.....	68
Ilustración 43. Código para obtener los usuarios más semejantes ordenados por puntuación y guardar el más cercano. Fuente: Elaboración propia. ....	69
Ilustración 44. Código creación de un dataframe comparativo. Fuente: Elaboración propia.....	69
Ilustración 45. Código para la generación de recomendaciones para un nuevo usuario. Fuente: Elaboración propia. ....	69
Ilustración 46. Código previsualizar las imágenes de los productos recomendados. Fuente: Elaboración propia. ....	69

# Índice de Tablas

Tabla 1. Tiempo de ejecución para cada medida de similitud. Fuente: (Khatte et al. 2021) .....	46
--	----

## Resumen

La creación y auge de los sitios web, en especial los *e-commerce* donde miles de productos son ofrecidos cada día convirtiendo a sitios como Amazon, Mercado Libre, entre otros, son los más deseados por los usuarios para realizar sus compras, pero ante la cantidad tan abrumadora de productos existente en ellos, manejarlos, ofertarlos y sugerirlos o recomendarlos a nivel individual se vuelve humanamente imposible. Ante esta problemática nacieron los sistemas de recomendaciones, como herramienta para procesar grandes volúmenes de datos, y generar sugerencias a los usuarios de sitios web.

Las recomendaciones pueden ser de diferentes tipos, por ejemplo: de productos (en plataformas de compras en línea), de películas o series (en plataformas de *streaming*), de música (en aplicaciones de música), entre otros. Estos sistemas utilizan algoritmos de aprendizaje automático y procesamiento de datos para analizar el comportamiento del usuario y predecir sus preferencias, generando así recomendaciones personalizadas.

Este trabajo está enfocado en el diseño de un modelo para un sistema de recomendaciones para un *e-commerce* denominado *iBuyFlowers*, basado en filtrado colaborativo enfocado en retroalimentación implícita o "*Implicit feedback*", en donde se utilizan como métricas los comportamientos en el pasado por usuarios (clicks, visitas a la página o su histórico de compras), evaluando así, la confianza de un producto basado en la frecuencia de las acciones de un usuario aplicando la técnica de Mínimos Cuadrados Alternos (*Alternating Least Squares algorithm*), así mismo se tuvo en cuenta escenarios diferentes para usuarios nuevos, en el cual se realizó la implementación de la métrica la similitud del coseno y así obtener el usuario más parecido basado en las preferencias, para todo ello, se tendrá en cuenta todos los datos recopilados en el año 2023, cumpliendo con las características de los clientes que han realizado compras a lo largo de ese año y finalmente obtener como resultado un listado de productos sugeridos para un usuario objetivo.

**Palabras clave:** Sistema de recomendación, *iBuyFlowers*, similitud de coseno, filtro colaborativo, Selección de vecinos más cercano, Retroalimentación implícita, Mínimos cuadrados alternos.



## Abstract

The creation and rise of websites, especially e-commerce where thousands of products are offered every day, making sites like Amazon, Mercado Libre, among others, the most desired by users to make their purchases, but given the such an overwhelming amount of products existing in them, managing them, offering them and suggesting or recommending them on an individual level becomes humanly impossible. Faced with this problem, recommendation systems were born as a tool to process large volumes of data and generate suggestions for website users.

Recommendations can be of different types, for example: products (on online shopping platforms), movies or series (on streaming platforms), music (on music applications), among others. These systems use machine learning and data processing algorithms to analyze user behavior and predict their preferences, thus generating personalized recommendations.

This work is focused on the design of a model for a recommendation system for an E-commerce called iBuyFlowers, based on collaborative filtering focused on implicit feedback, where past user behavior is used as metrics. (clicks, page visits or purchase history), thus evaluating the trust of a product based on the frequency of a user's actions by applying the Alternating Least Squares algorithm. into account different scenarios for new users, in which the implementation of the cosine similarity metric was carried out and thus obtain the most similar user based on preferences, for all this, all the data collected in the year 2023 will be taken into account. , meeting the characteristics of the clients who have made purchases throughout that year and finally obtaining as a result a list of suggested products for a target user.

**Keywords:** Recommendation system, iBuyFlowers, Cosine Similarity, collaborative filter, nearest neighbour selection, Implicit feedback, Alternating Least Squares algorithm.

# 1. Introducción

*iBuyFlowers* es un mayorista de flores B2B en línea, que vende flores frescas de granja y ramos preensamblados a floristas, organizadores de eventos, lugares para bodas, entre otros. Fundada en 2017, sus flores se envían directamente desde granjas de todo el mundo a la puerta de su tienda o a una ubicación específica en los Estados Unidos. Ofrecen flores para elegir, perfectas para cualquier ocasión. Así mismo, cuentan con precios competitivo, por lo que puedes ahorrar dinero sin sacrificar la calidad. Finalmente, los clientes pueden pedir flores en línea las 24 horas del día, los 7 días de la semana (*iBuyFlowers*, 2017). Actualmente, se llega a muchos clientes y cerrar ventas, lo cual la gran mayoría se encuentran satisfechos, pero a nivel de empresa no se consigue llegar al nivel de ventas deseado, el cual consiste en tomar como referencia el año anterior al actual, tomar el valor facturado y vender ese total + 10%. Esto se debe en parte a que en los acercamientos que se tienen con los usuarios y clientes para escuchar los reclamos, sugerencias y peticiones por parte de estos a la empresa, en su mayoría expresan que en el momento de buscar y encontrar productos que alguna vez compraron o leyeron acerca de ellos en la página web de la plataforma, emails informativos o redes sociales, por razones ajenas a ellos no los encuentran en la interfaz de *iBuyFlowers*.

Ahora, la empresa requiere el diseño de un sistema de recomendación para analizar el comportamiento de los clientes mediante su histórico de compras, de esta manera, ofrecer sugerencias de productos a un usuario objetivo en el momento de realizar sus compras para mejorar la experiencia y así aumentar el porcentaje de ventas exitosas, las cuales son consideradas efectivas por la empresa a través del proceso de conversión de usuarios a clientes, por medio de una práctica positiva que los motive a registrarse si son nuevos en la página web, seleccionar los productos, realizar una compra y seguir utilizando la plataforma en el futuro. Por lo tanto, en este trabajo se plantea realizar el diseño de un sistema de recomendación usando el filtrado colaborativo (*collaborative filtering*).

El filtrado colaborativo es una técnica de aprendizaje automático que recomienda artículos a los usuarios según su historial de compras anterior. Funciona encontrando clientes que tienen un historial de compras similar y luego sugerir artículos que estos han comprado. Esta puede ser una forma muy eficaz de recomendar artículos a las personas, ya que tiene en cuenta sus preferencias individuales (Pradel et al., 2011).

El presente trabajo estará enfocado en el filtro colaborativo, a través de la retroalimentación implícita, la cual refleja las preferencias de los usuarios basado en su comportamiento en el pasado. Este enfoque fue escogido debido que los datos adquiridos no cuentan con una calificación directa o una valoración por parte de los clientes que permitan identificar si hay un gusto o disgusto de ellos hacia los productos. Para el uso de la retroalimentación implícita es de suma importancia identificar

características únicas, esto permite optimizar el tiempo, a continuación, se mencionarán las principales: (Hu, 2015):

1. No se cuenta con comentarios negativos: Debido que no hay una valoración explícita, se hace difícil determinar de manera confiable por qué un usuario optó por consumir un producto o servicio, así mismo, genera incertidumbre en torno a por qué no escogió otro producto, lo cual puede deberse a distintos factores que pueden ser tomados en cuenta según la comprensión del negocio.
2. Puede ser ruidosa: En el caso de un *e-commerce*, no garantiza que el usuario haya comprado un producto por gusto, puede ser un regalo que desee obsequiar y por eso realizó la compra, dando una valoración que puede generar ruido dentro del modelo propuesto.
3. Valores numéricos indican confianza: Esto se debe a que en al realizar una retroalimentación explícita se cuenta con un valor que indica si un usuario prefiere un producto o no, expresándolo a través de calificaciones o un botón que indique me gusta o no me gusta, la retroalimentación implícita describe la frecuencia de una acción realizada por un usuario, en lo cual es importante tener en cuenta que un valor mayor, no indica que haya una preferencia por un artículo.
4. Evaluación: Para evaluar un sistema de recomendación implícita requiere de medidas específicas, debido que se puede tener en consideración temas como disponibilidad de un producto, competencia entre los productos o incluso la retroalimentación repetida.

Para la retroalimentación implícita se escogió el algoritmo de Mínimos Cuadrados Alternos (*Alternating Least Squares algorithm*) para la creación de un modelo que realiza la recomendación de los productos que serán sugeridos a los usuarios.

Así mismo se tuvo en cuenta el escenario en donde se necesitaba hacer recomendaciones para usuarios nuevos, en los cuales la retroalimentación implícita se queda corta porque solo evalúa lo que está dentro del *dataset*. Para ello se complementó el sistema de recomendación con el uso de la técnica de selección del vecino más cercano.

Para la aplicación de esta técnica se estudia, en primer lugar, como identificar que vecinos pueden ser una base que permita la generación de recomendaciones de productos para la persona seleccionada y a su vez como darle un uso indicado a la información que fue proporcionada por estos vecinos. Para ello, se utiliza una métrica

de similitud que permita la identificación de estos, teniendo en cuenta de manera general dos cosas (Bellogín et al., 2014):

1. Identificar un conjunto de datos en el usuario y el cliente hayan interactuado.
2. Realizar un análisis para hallar los puntos en el cual los usuarios y los clientes mostraron comportamientos similares en el conjunto de datos.

Teniendo en cuenta lo anterior, se utiliza un conjunto de datos obtenido de la base de datos del negocio donde está toda la información de los clientes y productos comprados en todo el año 2023. El cual está conformado por las preferencias de cada cliente (nombre, preferencias, dirección, entre otros), productos comprados (nombre, cantidad, entre otros) e información detallada de las compras (dirección, estado, valor total). Con este conjunto de datos, se analizará el comportamiento de los usuarios basado en el histórico de compras, siendo esta la frecuencia escogida para medir la confianza de un cliente referente a un producto así mismo se tendrá las preferencias guardadas en la plataforma como punto clave para definir que clientes serán los vecinos más cercanos al usuario objetivo como complemento de este análisis. Este conjunto de datos será preprocesado para identificar valores faltantes, errores en las columnas o filas, descartar valores, así mismo se realizan transformaciones para optimizar el uso de los datos; por ejemplo: renombrando columnas para facilitar la utilización de los datos y finalmente se aplica la similitud de coseno. Ésta es una métrica que facilita encontrar que tan parecidos son los elementos sin importar las dimensiones de estos (Sumathi et al., 2023), se calcula la similitud entre un usuario y un cliente midiendo el ángulo entre los vectores clasificados. Obteniendo como resultado el vecino más cercano basado en los valores de la similitud del coseno (cuanto menor es el coseno de dos vectores, más similares son) y los productos comprados por este (Munkholm et al., 2024).

Finalmente, esto permite dado un usuario objetivo, con sus preferencias previamente configuradas desde el sitio web del negocio donde realiza el registro previo realizar sus compras, calcular cuál es su vecino más cercano de los clientes que hayan realizado compras previamente y para luego obtener los productos comprados por éstos para posteriormente ser sugeridos o recomendados al usuario objetivo antes de finalizar su compra.

## 1.1. Estructura del documento

El trabajo está organizado de la siguiente manera:

En el segundo capítulo se presenta los objetivos en este trabajo.

El tercer capítulo contiene los aspectos teóricos del trabajo: *iBuyFlowers*, sistema de recomendación, filtrado colaborativo, similitud del coseno, SBXCloud, Mínimos cuadrados alternos.

En el cuarto capítulo se expone el desarrollo del proyecto para la recomendación de productos para la plataforma *iBuyFlowers*, en la cual se hace la explicación de la metodología seleccionada y aplicada para la comprensión del negocio, la comprensión de los datos, la preparación de los datos, el modelado evaluación e implementación o despliegue, finalmente se realiza un análisis de los resultados obtenidos para evaluar la eficiencia del modelo propuesto.

El quinto capítulo contiene las conclusiones del proyecto propuesto y las recomendaciones para trabajos futuros.

## 2. Objetivos

### 2.1. Objetivo General

Diseñar un sistema de recomendación basado en el análisis y procesamiento del histórico de datos de las compras de los usuarios de la plataforma *'iBuyFlowers'*.

### 2.2. Objetivos Específicos

- Estudiar los conceptos teóricos de los Sistemas de Recomendación.
- Explorar las técnicas y herramientas utilizadas para sistemas de recomendación.
- Analizar la base de datos de *'iBuyFlowers'* de las compras realizadas durante el año 2023 para obtener el dataset.
- Evaluar el modelo de recomendación propuesto.

## 3. Estado del Arte y Marco teórico

### 3.1. Estado del arte

En (Li, (2024)), se analiza un sistema de recomendación de comercio electrónico personalizado y colaborativo para sitios pequeños y medianos. Esta emplea principalmente el algoritmo de filtrado colaborativo para producir sugerencias altamente personalizadas y adaptables basadas en el historial de compras del usuario. El algoritmo también puede proponer nuevos conocidos basándose en sus comportamientos de compra similares, lo que aumenta la credibilidad del producto y la fidelidad a largo plazo. El modelo de recomendación propuesto incorpora además un elemento de ventana temporal para tener en cuenta la naturaleza en tiempo real de los datos y su escasez, así como resultados de investigaciones sobre redes sociales para mejorar las sugerencias de amigos. El sistema está construido en Java, utilizando Microsoft SQL Server 2005 como base de datos, y el servidor web Tomcat6.0. El sistema se organiza en torno a un módulo de recomendaciones que recoge información del usuario y genera recomendaciones de productos y amigos. Se realizaron experimentos para evaluar la eficacia del sistema, que revelaron una alta precisión, una mejor cobertura y mayores índices de popularidad de los productos recomendados en comparación con otros modelos.

En (Sumathi, et al (2023)) se centra en la mejora del método de filtrado colaborativo en los sistemas de recomendación, especialmente para las recomendaciones de alimentos. La propuesta adopta una fórmula de similitud mejorada, que incorpora el cálculo de la media y un parámetro de umbral temporal en la ponderación de la similitud, y añade ponderaciones basadas en el número de ítems compartidos a partir del cálculo previo de la similitud. Además, el estudio examina algoritmos de filtrado colaborativo basados en usuarios y en elementos, así como la forma en que determinan la similitud entre individuos u objetos. Para validar el método se utilizó el conjunto de datos MovieLens, uno de los más populares en el campo de los sistemas de recomendación en la actualidad bajo dicho contexto.

En (Vullam et al (2023)), se analiza los sistemas de recomendación personalizados para el comercio electrónico, haciendo hincapié en el uso de la agrupación de usuarios en un sistema Multiagente para aumentar la precisión y eficacia de los procesos de recomendación. El artículo define tres tipos de sistemas de recomendación: híbridos, basados en el contenido y colaborativos. Los sistemas basados en el contenido tienen en cuenta las propiedades de los objetos recomendados, mientras que los sistemas colaborativos emplean medidas de similitud para sugerir artículos compartidos por personas con intereses comparables.

El proceso de filtrado colaborativo es el más popular y eficaz en los sistemas de recomendación, pero puede verse obstaculizado por el tiempo necesario para localizar al vecino más cercano al usuario objetivo en todo el espacio del usuario, a medida que

crece el número de usuarios y productos en el sistema. Como resultado, se ofrece una solución basada en la agrupación de usuarios en un sistema Multiagente. Los usuarios se dividen en categorías de productos en función de sus valoraciones de preferencia, y sólo se buscan los vecinos más cercanos de su categoría. La estrategia de *clustering*, junto con el filtrado colaborativo, aumenta el rendimiento del sistema de recomendación, medido por la precisión, el recuerdo y la especificidad.

En (Fan, Hu, (2023)), se describe un sistema de recomendación de películas basado en la cognición y el filtrado colaborativo que utiliza la similitud emparejada del coseno como medida de evaluación de múltiples calidades de películas. Se examina el conjunto de datos de películas y créditos, y se ofrecen técnicas de filtrado basadas tanto en el contenido como en la colaboración de los usuarios. Se emplea Python para el análisis de datos y se demuestra cómo el filtrado colaborativo y el basado en el contenido pueden integrarse en un sistema híbrido de recomendación de películas. También se analizan las limitaciones y puntos fuertes de cada método de filtrado, así como la forma en que el enfoque híbrido puede ayudar a superar algunas de estas limitaciones, destacando el valor potencial del uso de sistemas de recomendación para mejorar la experiencia del usuario.

El enfoque de la similitud del coseno puede utilizarse para determinar la similitud de los usuarios en un sistema de recomendación. Esta estrategia es ampliamente utilizada en el campo de la recomendación, ya que es sencilla de ejecutar y tiene éxito. Para calcular esta similitud según Munkholm, et al., (2024), se utiliza una fórmula que consiste en calcular el producto escalar de los dos vectores y dividirlo por el producto de sus magnitudes. La métrica de similitud coseno es adecuada cuando la información dada se representa como valores booleanos, como «Me gusta» o «No me gusta». Además, esta estrategia es muy eficaz con conjuntos de datos escasos o dispersos. En general, el enfoque de la similitud del coseno es una herramienta importante en el campo de la recomendación, ya que permite medir de forma eficaz y directa la similitud entre el usuario y el producto.

En (Li, (2021)), se estudia un sistema de recomendación de comercio electrónico móvil. El sistema utiliza información personalizada del usuario, como rasgos del usuario, comportamiento histórico y objetos, para proponer cosas que pueden ser de su interés. Se muestra una matriz de preferencias de comportamiento, en la que cada usuario está representado por un vector que incluye sus valoraciones de diversos artículos. El método utiliza la similitud coseno para evaluar la similitud de los intereses de los usuarios e identificar a los usuarios más cercanos a un usuario objetivo seleccionando una colección de vecinos con los intereses y preferencias más comparables. Los resultados de las pruebas de rendimiento del algoritmo, evaluados por el error medio absoluto (MAE), que evalúa la precisión de los resultados de la recomendación, demuestran que el método de filtrado colaborativo funciona mejor para las recomendaciones personalizadas.

En (Zhang, Yan, ((2010))), se desarrolla un sistema de recomendación colaborativo basado en la técnica de filtrado colaborativo y aplicado en el ámbito del comercio



electrónico. Se propone un nuevo algoritmo que combina la medida de similitud coseno basada en vectores difusos y la correlación de Pearson con la similitud direccional de los elementos. Para lidiar con el problema de la escasez de datos y mejorar la precisión de las recomendaciones, se introduce un nuevo método de conversión de matriz que utiliza conjuntos difusos para describir la popularidad de los elementos. Además, se utiliza la similitud direccional de los elementos para ajustar la escala de similitud y reducir el ruido de los elementos similares. Finalmente, los resultados experimentales son obtenidos con la aplicación del nuevo algoritmo en un conjunto de datos MovieLens. Los resultados demuestran que el algoritmo propuesto tiene una alta precisión de predicción y es resistente al tamaño del vecindario.

En (Zhang, Yang, (2011)), se propone un sistema de recomendaciones personalizado para el comercio electrónico, basado en la retroalimentación del usuario. Se divide en dos partes: una offline y otra online. En la primera (offline), se utiliza un algoritmo de reglas de asociación para recomendar productos en función de las características de navegación en línea del usuario. El sistema recopila la retroalimentación del usuario en una base de datos de *feedback* de clientes, lo que permite modificar las recomendaciones en futuras visitas. En cambio, en la parte online, el motor de recomendación utiliza la conversación actual del cliente para producir una página set de recomendaciones personalizadas que se agregan a la parte inferior de la página solicitada en forma de hipervínculos. El sistema ajusta dinámicamente las recomendaciones según la retroalimentación del usuario, incluyendo el ajuste del valor de recomendación y la adaptación de la táctica de recomendación.

En (Singh, Rishi, (2020)), se implementa un sistema de recomendación colaborativa de comercio electrónico basado en la acción del usuario (por ejemplo, clics, selecciones y compras) en lugar de en valoraciones o reseñas de productos. Además, el sistema sugiere el uso de una base de conocimientos basada en redes para complementar las recomendaciones dadas por el sistema de recomendación colaborativa. El rendimiento del sistema se compara con los modelos de base utilizando un conjunto de datos de comercio electrónico de Amazon en tiempo real y se mide utilizando métricas de precisión como la precisión, el recuerdo y la NDCG. En general, el sistema de recomendación colaborativa propuesto y su integración con la base de conocimientos basada en grafos aumentaron la calidad de las sugerencias proporcionadas. Los resultados demuestran un aumento de la precisión en comparación con los modelos de referencia, lo que indica que la incorporación de los conocimientos del usuario puede ayudar a mejorar el rendimiento del sistema de recomendación colaborativa.

En (Yadav et al., (2018)), se examina el desarrollo de sistemas recomendadores de comercio electrónico, que ofrecen sugerencias basadas en las preferencias del usuario y mejoran la experiencia de compra en línea. El artículo presenta un método híbrido basado en el contenido y el filtrado colaborativo que emplea la distancia coseno para determinar la similitud entre los perfiles de usuario y los artículos. El rendimiento del sistema se evalúa determinando la precisión y el recuerdo de los resultados de las sugerencias. El despliegue del sistema de recomendación aumenta la precisión de las sugerencias y puede utilizarse para impulsar las ventas del comercio electrónico. La

técnica propuesta mejora la precisión de las sugerencias generadas y tiene potencial para ofrecer una mejor experiencia de usuario en el comercio electrónico.

## 3.2. Marco teórico

### 3.2.1. Sistema de Recomendación

Un sistema de recomendación es una herramienta que brinda sugerencias de elementos que probablemente sean de intereses para un usuario en particular a través de técnicas de software. Los sistemas de recomendación se centran normalmente en un tipo de elemento específico, por ejemplo, películas, artículos de noticias, entre otros. Por lo tanto, su interfaz gráfica de usuario se personaliza en base a proporcionar sugerencias útiles y efectivas para el tipo de elemento establecido inicialmente.

Se iniciaron a partir de observaciones tales como: un grupo de personas dependen de recomendaciones proporcionadas para tomar decisiones cotidianas, en los negocios o en general, por ejemplo, preguntar a un conocido que libro leer, que música escuchar, otro ejemplo sería, un empleador solicita una carta de recomendación para tomar decisiones en sus contrataciones.

Estos sistemas, están dirigidos para personas o negocios que escasea de competencia para la evaluar la gran cantidad de elementos que un sitio web tiene para ofrecer y así personalizar la experiencia para cada cliente. En algunas ocasiones estos sistemas pueden generar recomendaciones sencillas y no personalizadas, por ejemplo, en un sitio de películas, el top 10 de mejores películas, llegando a ser útiles cuando no se dispone de suficiente información sobre las preferencias, características o intereses del usuario objetivo.

Así mismo, los sistemas de recomendación intentan predecir los productos o servicios más adecuados, siempre en base a las afinidades del usuario, para llevar a cabo esto, recopilan información de los usuarios sobre sus preferencias que se expresan explícitamente, por ejemplo, calificaciones a productos o interpretando acciones que realizan usuarios al usar el sistema, un ejemplo de ello sería que el sistema de recomendación considere la compra del artículo por parte de un usuario e implícitamente entienda la preferencia hacia ese producto (Ricci et al., 2011).

En (Ricci et al., 2011) se plantean seis distintos tipos de sistemas de recomendación:

- **Recomendación basada en contenidos:** En ella se recomienda al usuario artículos similares a los que él ha preferido en el pasado. Los sistemas de recomendación basados en contenido analizan un conjunto de artículos y/o descripciones preferidas en el pasado por un usuario, y construye un modelo o perfil de los intereses o preferencias del usuario basado en las características de estos artículos.
- **Recomendación colaborativa:** Este tipo de recomendación recomienda al usuario artículos que personas con gustos o preferencias similares usaron en el pasado. En la recomendación colaborativa (o de filtrado colaborativo), los

sistemas predicen el interés del usuario en nuevos artículos basados en las recomendaciones realizadas a otras personas con intereses similares.

- **Recomendación demográfica:** En ella se clasifica los usuarios en clases demográficas de acuerdo con los atributos de su perfil personal, y hace recomendaciones basadas en dichas clases.
- **Recomendaciones basadas en utilidad:** En ella se realizan recomendaciones sobre la base de un cálculo de utilidad de cada elemento para un usuario. Lo anterior requiere del planteamiento y uso de una función de utilidad.
- **Recomendación basada en conocimiento:** En ella se sugieren artículos basados en inferencias lógicas sobre las preferencias del usuario. Es necesaria una representación del conocimiento (por ejemplo, reglas) acerca de cómo un artículo responde a la necesidad de un usuario.
- **Recomendación híbrida:** En este tipo de recomendación se combinan dos o más tipos de recomendación de los descritos anteriormente, con el fin de obtener un mejor rendimiento y abordar las deficiencias de cada tipo de recomendación.
- **Sistemas recomendadores embebidos en entornos de aprendizaje mejorado con tecnología (TEL *Technology Enhanced Learning*):** El despliegue de los sistemas recomendadores ha cobrado mayor interés a partir del instante en que el área de recuperación de información (information retrieval) se convirtió una actividad fundamental en TEL (entendido como búsqueda de recursos de aprendizaje relevantes para soportar profesores y aprendices). Lo anterior se fundamenta en el hecho de que un problema tradicional en TEL ha sido el mejorar la búsqueda de recursos de aprendizaje digitales. A consecuencia de lo anterior, el concepto de sistemas recomendadores se ha convertido en un área muy atractiva en la investigación en TEL.

Según (Manouselis et al, 2014) de todos los esfuerzos realizados han surgido múltiples observaciones interesantes, entre las que se destacan:

- Hay un gran número de sistemas de recomendación que han sido desplegados en configuraciones TEL.
- Las metas de recuperación de información que persiguen los sistemas recomendadores TEL son a menudo diferentes a los identificados en otros sistemas (por ejemplo, en recomendación de productos).
- Existe necesidad de identificar las particularidades de los sistemas recomendadores TEL, buscando elaborar métodos propios para sus diseños semánticos, desarrollo y evaluación.

### 3.2.2. Filtrado Colaborativo

El filtrado colaborativo es técnica recursiva de recomendación de productos basados en la opinión de otros usuarios, que comparten gustos similares y permite mejorar la experiencia del cliente. Está es una técnica de proximidad y relacionable (Sumathi et al., 2023).

Para usar está técnica, se deben proporcionar datos que representen una constancia o evidencia de las preferencias o características de los usuarios, debido que son valores explícitos que permiten estudiar el interés de estos hacia los elementos. Esto permitirá a los algoritmos de filtrado colaborativo que al ser los datos tan detallados no necesiten descripción de los elementos para realizar una recomendación, puesto que solo toma la información proporcionada entre los usuarios y los elementos, obteniendo así una recomendación potencialmente novedosa basado en la experiencia de otros usuarios (ver ilustración 1).

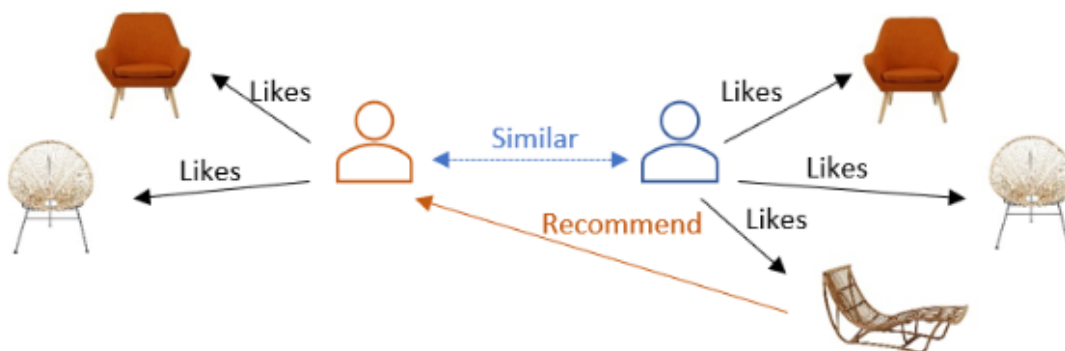


Ilustración 1. Descripción grafica de usuarios con gustos similares en productos. Fuente: (Munkholm et al, 2024)

El filtrado colaborativo se clasifica en dos categorías:

- Basado en contenido o elemento (*Item/Content based*): se realiza la sugerencia teniendo en cuenta los resultados de similitud con los elementos en base a las actividades realizadas por otros clientes y el usuario objetivo (Sumathi et al., 2023), creando patrones de calificación de usuario/elemento que realizan calificaciones automáticas dadas por una predicción obtenida de estos patrones (Bellogín et al., 2014).
- Basado en usuario o vecino más cercano (*Nearest Neighbour*): se realiza las recomendaciones basado en las preferencias o características de un usuario cuyos intereses sean afines a los intereses de otro usuario, estos son llamados vecinos. Los enfoques de las recomendaciones pueden verse afectados por la selección y peso o ponderación que se les asigne a las características de estos usuarios,

identificando así primeramente que vecinos hacen parte de la base que permita la generación de una recomendación y realizar el uso adecuado de la información obtenida para su posterior uso (Bellogín et al., 2014).

### 3.2.3. Similitud del Coseno

La similitud del coseno es una medida utilizada para comparar documentos o dar una clasificación de documentos dado un vector determinado con palabras de consulta (Munkholm et al., 2024). Sean A y B dos vectores a comparar, se obtiene la ecuación (1):

$$\text{similarity}(A, B) = \cos\theta = \frac{A \cdot B}{\|A\| \cdot \|B\|} \quad (1)$$

Dónde:

A, B son los dos vectores.

A.B es el producto escalar de dos vectores.

$\|A\| \cdot \|B\|$  es la multiplicación de la magnitud de los vectores.

Por ejemplo, si se tienen los vectores A y B que se muestran a continuación:

$$A = [1, 1, 1, 1, 1, 0, 0]$$

$$B = [0, 0, 1, 1, 0, 1, 1]$$

Se calcula el producto escalar entre los vectores A y B y se obtiene:

$$A \cdot B = (1 \times 0 + 1 \times 0 + 1 \times 1 + 1 \times 1 + 1 \times 0 + 1 \times 0 + 0 \times 1) = 2$$

La magnitud de un vector se obtiene como la raíz cuadrada de la suma de los cuadrados de sus componentes. En el caso del ejemplo se tiene para cada vector lo siguiente:

$$\|A\| = \sqrt{1^2 + 1^2 + 1^2 + 1^2 + 1^2 + 0^2 + 0^2} = \sqrt{5}$$

$$\|B\| = \sqrt{0^2 + 0^2 + 1^2 + 1^2 + 0^2 + 1^2 + 1^2} = \sqrt{4}$$

Finalmente, aplicando la ecuación (1) se obtiene la similitud del coseno para los vectores A y B, como:

$$\cos\theta = \frac{A \cdot B}{\|A\| \cdot \|B\|} = \frac{2}{\sqrt{5} \cdot \sqrt{4}} = \frac{2}{\sqrt{20}}$$

$$\cos\theta = 0.4472$$

La similitud se mide por el coseno del Angulo entre dos vectores y así obtener como resultado si los dos vectores señalan hacia la misma dirección. Un ejemplo de cómo se visualiza sería a través de la comparación de documentos, un documento puede llegar

a contener muchos atributos, por los cuales registra frecuencias de una palabra en particular, por lo tanto, cada documento es un objeto representado bajo la denominación de término-frecuencia (Han et al., 2012). Dado estos atributos, se puede obtener la similitud entre un documento u otro, obteniendo resultados que serán utilizados según los objetivos con los que se haya planteado el análisis.

### 3.2.4. SBXCloud

SBXCloud es una plataforma como servicio (PaaS) la cual provee herramientas, servicios y funcionalidades que permiten desarrollar, probar, alojar, administrar y ejecutar aplicaciones web o móviles. Los usuarios pueden gestionar una base de datos creada para uno o más proyectos, dado que les permite crear, actualizar, eliminar tablas, propiedades, así mismo, crear servicios que se conecten a esos datos, finalmente permite la administración de archivos como un repositorio donde el usuario puede ejecutarlos manualmente o puede conectarse al proyecto al cual está relacionado en la plataforma y manipular la información desde una aplicación (SBXCloud, s.f.).

### 3.2.5. Mínimos cuadrados alternos

La técnica de mínimos cuadrados alternos consiste en la factorización matricial comúnmente usada en sistemas de recomendación con filtrado colaborativo. Su propósito consiste en tomar una matriz que contiene valores dispersos entre usuarios y elementos, descomponiéndolas en dos matrices inferiores cuyo producto se aproxima a la original. La implantación de esta técnica se vuelve útil cuando el enfoque está basado en filtrado colaborativos y el objetivo es predecir calificaciones de los elementos/productos que aún no han sido calificados por un usuario. Esta calificación se puede calcular teniendo en cuenta el producto escalar de los vectores correspondiente de un usuario y un elemento (Loukili, et al., 2023) como lo se puede observar a continuación en la ecuación (2):

Para cada usuario denominado  $u$  el vector del usuario  $u$  puede ser computado resolviendo el problema de mínimos cuadrados expuesto a continuación:

$$u = ((U^t * Cu * I) + \text{lambda} * E)^{-1} * I^t * Cu * r(u) \quad (2)$$

En donde:

- $I$  Es la matriz de elementos
- $r(u)$  Es el vector de calificación dado por un usuario  $u$
- $Cu$  Es la matriz diagonal de ponderaciones de confianza para el usuario
- $\text{lambda}$  Es una regularización como parámetro que previene el sobre ajuste
- $E$  Es la matriz de identidad del tamaño apropiado

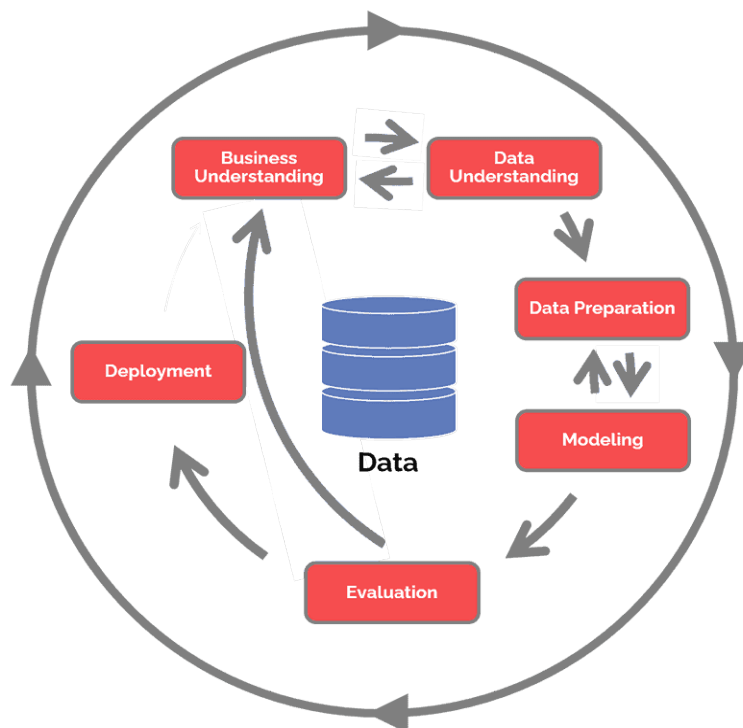
## 4. Desarrollo del proyecto y resultados

### 4.1. Metodología

En este trabajo se ha utilizado la metodología CRISP-DM, la cual proporciona una descripción general del ciclo de vida de un proyecto, describiendo sus fases, sus tareas y las relaciones entre estas. Estas relaciones se establecen dependiendo de los objetivos, antecedentes, intereses del usuario y fundamentalmente de los datos (Chapman, et al., 2000).

El ciclo de vida de un proyecto está conformado por seis fases o etapas (ver ilustración 2):

- Comprensión del negocio.
- Comprensión de los datos.
- Preparación de los datos.
- Modelo.
- Evaluación.
- Implementación o despliegue.



*Ilustración 2. Fases del modelo de referencia de CRISP-DM. Fuente: (Chapman, et al., 2000)*



Esta metodología es flexible debido que no es secuencial o método de cascada en un solo sentido, está diseñada para tener en cuenta el resultado de cada fase, y analizar si es necesario volver a la fase anterior tomando en cuenta objetivos, situaciones o reglas del negocio que puedan variar durante el proceso y así mejorar el resultado.

A continuación, se describe cada fase de la metodología CRISP-DM (Chapman et al., 2000).

#### 4.1.1. Comprensión del negocio

Es la etapa inicial que está focalizada en comprender los objetivos y requerimientos del proyecto a empezar teniendo en cuenta el enfoque empresarial que este va a llevar, esto permitiría conocer y definir el problema que se va a abordar y definir un plan de acción preliminar (Chapman et al., 2000).

#### 4.1.2. Comprensión de los datos

En esta etapa se inicia con la recolección y estudio de los datos disponibles a través de la identificación de problemas de calidad e identificación de patrones significativos mediante técnicas como la reducción de dimensionalidad o la selección de características. Además de formular diferentes hipótesis de las relaciones entre variables, así mismo se aborda la información faltante para evaluar la calidad de los datos que eviten problemas en el uso de estos en las siguientes fases del proyecto (Chapman et al., 2000).

#### 4.1.3. Preparación de los datos

Esta etapa consta en la construcción del conjunto de datos final que será utilizado en la fase de modelado, aquí se lleva a cabo diferentes técnicas de limpieza, identificaciones de campos faltantes, transformación de los datos, entre otras tareas de preprocesamiento para el conjunto de datos seleccionado inicialmente para el desarrollo del proyecto. En esta fase, no existe un orden prescrito para realizar el preprocesamiento de los datos, por lo tanto, es probable que esto sea repetido una y otra vez hasta conseguir el resultado deseado en el conjunto final de datos, debido que puede incluir la selección de tablas, registro o atributos de una fuente de datos, tales como bases de datos, repositorios, entre otras, que estarán involucrados en el proceso para la obtención del conjunto final deseado (Chapman et al., 2000).

#### 4.1.4. Modelado

En esta etapa, se suele tener varias iteraciones, debido a que se puede utilizar diferentes técnicas de minería de datos: reconocimiento de patrones, asociación, clasificación, *clustering*, árboles de decisión, análisis de regresión, dependiendo del problema que se desea resolver, en este paso, se realizan pruebas, para ajustar los parámetros necesarios para obtener los valores óptimos, dado que cada técnica que se

puede utilizar requiere la presentación de los datos de manera distinta, de esta manera, se selecciona cual se adapta mejor a los datos y a los resultados que se desean obtener en el desarrollo del proyecto o se debe regresar a la fase de preparación de datos para realizar mejoras en ellos. Finalmente, no se escoge un solo modelo en la primera ejecución con los parámetros que se tienen, sino que se toma en cuenta las diferentes alternativas que existen para abordar el problema. (Chapman et al., 2000).

#### 4.1.5. Evaluación

En esta etapa, se habrá llevado a cabo gran parte del proyecto, así mismo se ha tomado la decisión en la fase anterior de modelado donde, el o los modelos seleccionados son técnicamente correctos en función de los objetivos y criterios al iniciar el proyecto. Para ello se deben evaluar los resultados utilizando métricas de precisión correspondientes, aplicadas a la técnica o estrategia implementada en el proyecto. Es importante tener en cuenta si hay algún tema empresarial que no haya sido considerado y que pueda afectar el modelo, puesto que al final de esta fase se debe tomar la decisión sobre la técnica o modelo a utilizar a partir de los resultados obtenidos (Chapman et al., 2000).

#### 4.1.6. Implementación o Despliegue

A partir de los requisitos planteados por el negocio la implementación puede ser un informe o un proceso de extracción de datos extenso dentro de la empresa.

Es de vital importancia que el conocimiento adquirido a lo largo del proyecto sea organizado y presentado al cliente de manera clara y detallada para su uso, debido que algunas aplicaciones de estos modelos son utilizadas para personalizaciones en tiempo real o lecturas de bases de datos para extraer información. Por lo tanto, es importante que el cliente comprenda que acciones deben realizarse para el correcto uso de los modelos creados (Chapman et al., 2000).

### 4.2. Planteamiento del problema

La Plataforma *iBuyFlowers* (IBF) presenta fuga en los clientes que realizan compras habituales, esto se debe a diversos factores tales como:

1. No conocer la plataforma, su uso, sus funciones, sus ventajas, entre otros.
2. Problemas durante la compra debido a *bugs* del software.
3. No recibir orientación por parte de los vendedores asignados al usuario para usar la plataforma y posteriormente realizar una compra en ella.
4. No encontrar disponibilidad de un producto en una fecha dada y desconocer como buscar un producto en otra fecha con las mismas características.

Así mismo, para la retención de los clientes nuevos se utilizan estrategias de marketing, *emails* con información de descuentos en sus compras, combos, interacción directa con los usuarios, pero estos procesos hasta ahora no se logran que los clientes vuelvan recurrentemente a realizar compras y de esta manera aumentar las ventas. En algunas ocasiones, ocurre que los clientes deben hacer compras rápidas y no encuentran los productos en la página debido a la gran variedad del catálogo y suelen desertar de la plataforma.

Por consiguiente, la gerencia de la empresa se muestra preocupada debido que a pesar de que apuntan al crecimiento anual de la empresa reflejado en sus estadísticas internas, no alcanzan el umbral deseado en las ventas, el cual consiste en tomar el valor vendido en el mismo mes del año anterior, sumarle el 10% y al cumplir el mes actual ese debe ser el valor mínimo de ventas para obtener la meta esperada de cada mes (ver ecuación (2)).

$$\text{Meta Minima Mensual} = \text{Valor venta mes año anterior} * (1 + 0.1) \quad (2)$$

Por lo tanto, se optó por realizar el análisis de la competencia, observar que estrategias implementan para ofrecer más productos, analizar como lo hacen, enfocados en el sitio web, dado que las compras se realizan de manera online la plataforma *iBuyFlowers* es la insignia de la empresa, con la cual, siempre se busca de su continuo crecimiento.

Al realizar el análisis de otras páginas web tales como Amazon, mercado libre, Ebay, entre otras, son plataformas web de ventas en donde se observa en su sitio sus recomendaciones al entrar al detalle de cada producto, esto se puede observar en secciones tales como, “Con este artículo, también compran esto...”, “Antes de finalizar su compra, desea llevar algo más...”, “¿Un último antojo? ...” entre otros, llegando a la siguiente conclusión: En el sitio web de *iBuyFlowers* actualmente no existe una implementación que facilite a los usuarios tener una última sugerencia a la hora de realizar la compra.

Por lo tanto, se necesita un sistema de recomendación que esté en la capacidad de analizar el comportamiento de los usuarios a través de su histórico de compras, analizando las características o preferencias configuradas al momento del registro en el sitio web, permitiendo así realizar un análisis que ofrezca una experiencia personalizada y única para cada usuario según sus intereses, considerando que se tiene una base de datos de alrededor de 4 años con información relevante sobre los usuarios y las compras realizadas en ella, creando la necesidad de incluir una técnica de recomendaciones, tomando la información mencionada anteriormente almacenada en la base de datos y usarla como una herramienta poderosa que traiga finalmente un beneficio para ambas partes, el negocio y el usuario, otorgando así un mayor porcentaje de ventas exitosas.

### 4.3. Desarrollo del proyecto

Para el desarrollo del proyecto se han utilizado un conjunto de herramientas para la extracción, transformación, modelado y visualización de los datos. Teniendo como base el uso de la herramienta Jupyter Notebook y el lenguaje de programación Python.

Se escogió Jupyter notebook para el desarrollo del proyecto en lugar de otras herramientas como Google Colab, o cualquier herramienta en línea por la cantidad de datos a procesar debido que, al ser tan grandes, limitaba el uso gratuito de estas. Por lo tanto, se realizó la implementación en un computador local con las siguientes características:

- Maquina: MacBook Pro-14 Pulgadas (3024 × 1964)
- Año: 2021
- Chip: Apple M1 Pro
- Memoria Ram: 16GB
- Disco duro: 512 SSD
- Sistema Operativo: macOS – Ventura 13.4.1

Se usa el lenguaje Python porque es el que se utiliza para el desarrollo de los modelos y aplicaciones en ciencias de datos y es una herramienta que contiene un conjunto de librerías que permite manejar grandes conjuntos de datos de manera rápida y eficiente.

En este proyecto se emplean librerías como numpy, pandas y sklearn para el manejo y procesamiento de los datos, así mismo matplotlib y seaborn para la visualización de estos (ver Apéndice). A continuación, se muestra una descripción de cada librería:

- **Numpy:** Significa “*Numerical Python*”, es la librería principal para la informática científica, que proporciona grandes estructuras de datos. La versión utilizada de NumPy es v1.18.0 (Moreno, 2020). Tiene como características como:
  - Implementación de matrices N-dimensionales
  - Contiene: integrales, generadores de números totalmente aleatorios, transformadas de Fourier.
  - Interoperabilidad. Permite plataformas de hardware.
  - Fácil de usar. Alto nivel de sintaxis que facilita la accesibilidad a cualquier programador.
  - Código abierto. Mantenido por la comunidad diversa GitHub.
  - Muy utilizado en el desarrollo de algoritmos de *Machine Learning*.
- **Pandas:** es una librería de Python. Las estructuras de datos principales en pandas son Series para datos en una dimensión y *dataframe* para datos en dos dimensiones.  
Estas son las estructuras de datos usadas en campos tales como: finanzas, estadística, ciencias sociales y muchas áreas de ingeniería. Pandas destaca por

lo fácil y flexible que hace la manipulación de datos y el análisis de datos (Vilca, 2020).

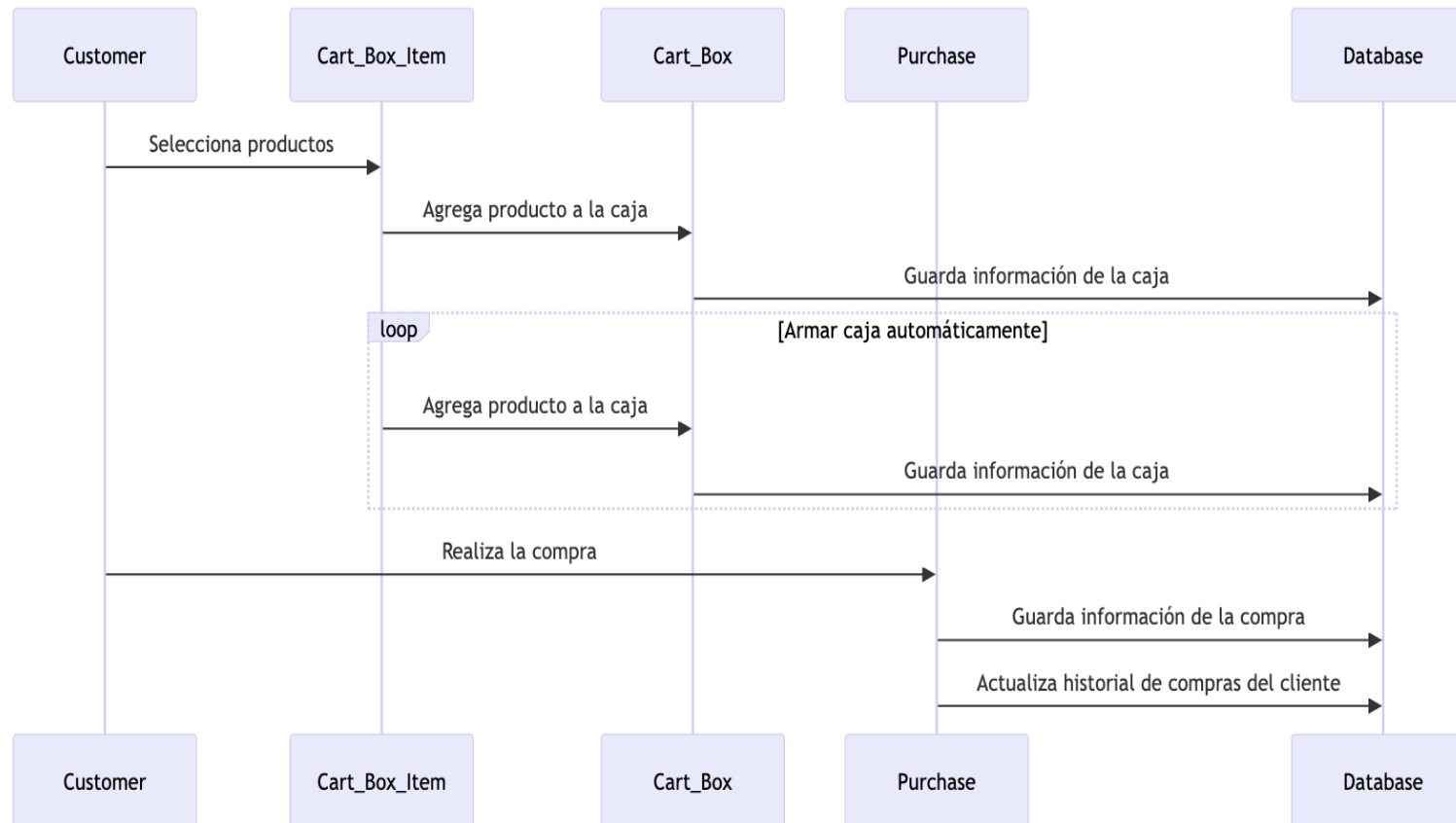
- **Matplotlib:** es una biblioteca para la generación de gráficos a partir de datos contenidos en listas o *arrays* en el lenguaje de programación Python (Vilca, 2020).
- **Seaborn:** es una librería gráfica basada en matplotlib, especializada en la visualización de datos estadísticos. Se caracteriza por ofrecer un interfaz de alto nivel para crear gráficos estadísticos visualmente atractivos e informativos. Seaborn considera la visualización como un aspecto fundamental a la hora de explorar y entender los datos. Se integra muy bien con la librería de manipulación de datos pandas (Vilca, 2020).
- **Sklearn:** es la librería de software libre la cual ofrece varios algoritmos, tanto de clasificación, regresión entre otros aplicados al análisis de datos y *Machine Learning*. Esta librería también tiene funciones para poder preparar los datos, esto permite limpiar el corpus de datos de entrenamiento y mejorar el clasificador. Está basado en NumPy, SciPy y Matplotlib. Su principal ventaja consiste la versatilidad de sus técnicas que implementa y la facilidad del uso de estas (Vilca, 2020).
- **Implicit:** Proporciona implementaciones rápidas en Python de varios algoritmos de recomendación populares diferentes para conjuntos de datos de retroalimentación implícita (Frederickson, n.d.).
- **Spicy:** SciPy provides algorithms for optimization, integration, interpolation, eigenvalue problems, algebraic equations, differential equations, statistics and many other classes of problems (SciPy, n.d.).

Luego de describir el entorno de trabajo y las herramientas computacionales necesarias, se muestra a continuación como se aplica cada uno de los pasos de la metodología planteada:

- **Comprensión del negocio:**

Al analizar los puntos clave que *iBuyFlowers* desea abarcar para la retención de clientes y satisfacción de estos en su proceso de compra. Se analizó los puntos como posicionamiento de la empresa, clientes, productos, conocer cómo es el proceso de registro, conversión de un usuario a la plataforma y el proceso de compra que estos realizan.

De esta manera, el proceso de compra se observa de la siguiente manera, (ver ilustración 3):



*Ilustración 3. Secuencia del proceso de compra de un usuario en iBuyFloers. Fuente: Elaboración propia*

Como muestra en la ilustración 3, en esta secuencia se visualiza como un usuario, al llegar al sitio web realiza la selección de productos, se crea una caja de los elementos seleccionados y finalmente realiza la compra, momento en el cual, se almacena la información en la base de datos, y se actualiza el historial de compras del usuario involucrado. La secuencia actual no incluye actualizaciones de inventarios ni otras operaciones internas que se realizan al momento de detectar una compra por la plataforma, por ejemplo, el envío de correos, conectarse con el transportador que recogerá y entregará los pedidos, entre otros.

Así mismo, se debe tomar en cuenta el atributo “tier”, el cual contiene un valor en el rango de 1 – 3, que permite clasificar a los usuarios para luego establecer que estrategia de marketing ofrecerles; además, se puede modificar los precios de los productos a través de otro atributo llamado “margin”, según el tipo de usuario, para lograr que adquieran otros productos y así obtener una mayor ganancia sobre sus compras.

Finalmente, luego de analizar lo anterior, se propone a la plataforma “iBuyFlowes” actualizar el sistema para captar mejor a los clientes y unirlos a sus procesos de ventas.

- **Comprensión de los datos:**

Para la compresión de los datos, se realiza el análisis de los datos a utilizar en el proyecto, por medio de SBXCloud tomando en cuenta las tablas involucradas:











- **Purchase:** Contiene el historial de compras de un usuario (Customer) (ver ilustración 4).

Ver	10	Registros	Modelo	purchase	Buscar en pagina			
<input type="checkbox"/>	KEY	customer_payment_option (REFERENCE-> customer_payment_option)	customer_date (DATE)	customer_user (REFERENCE-> user)	payment_status (STRING)	street (STRING)	state (REFERENCE-> state)	
<input type="checkbox"/>								
<input type="checkbox"/>								
<input type="checkbox"/>								
<input type="checkbox"/>								
<input type="checkbox"/>								
<input type="checkbox"/>								
<input type="checkbox"/>								
<input type="checkbox"/>								
<input type="checkbox"/>								
<input type="checkbox"/>								
<input type="checkbox"/>								
<input type="checkbox"/>								
<input type="checkbox"/>								
<input type="checkbox"/>								
<input type="checkbox"/>								
<input type="checkbox"/>								
<input type="checkbox"/>								
<input type="checkbox"/>								
<input type="checkbox"/>								
<input type="checkbox"/>								
<input type="checkbox"/>								
<input type="checkbox"/>								
<input type="checkbox"/>								
<input type="checkbox"/>								
<input type="checkbox"/>								
<input type="checkbox"/>								
<input type="checkbox"/>								
<input type="checkbox"/>								
<input type="checkbox"/>								
<input type="checkbox"/>								
<input type="checkbox"/>								
<input type="checkbox"/>								
<input type="checkbox"/>								
<input type="checkbox"/>								
<input type="checkbox"/>								
<input type="checkbox"/>								
<input type="checkbox"/>								
<input type="checkbox"/>								
<input type="checkbox"/>								
<input type="checkbox"/>								
<input type="checkbox"/>								
<input type="checkbox"/>								
<input type="checkbox"/>								
<input type="checkbox"/>								
<input type="checkbox"/>								
<input type="checkbox"/>								
<input type="checkbox"/>								
<input type="checkbox"/>								
<input type="checkbox"/>								
<input type="checkbox"/>								

*Ilustración 4. Previsualización de un fragmento de la tabla Purchase con algunas propiedades e información sensible oculta. Fuente: Elaboración propia*

- **State:** Guarda la información de cada estado como su nombre o ISO y su zona horaria. Por ejemplo: Texas - ISO -> (TX), zona horaria UTC (-6) (ver ilustración 5).

Ver 10 Registros Modelo state

<input type="checkbox"/>	KEY	state_iso (STRING)	state (STRING)	time_zone (STRING)
<input type="checkbox"/>		PR	Puerto Rico	UTC (-4)
<input type="checkbox"/>		VT	Vermont	UTC (-5)
<input type="checkbox"/>		MN	Minnesota	UTC (-6)
<input type="checkbox"/>		OR	Oregon	UTC (-8)
<input type="checkbox"/>		MT	Montana	UTC (-7)
<input type="checkbox"/>		CA	California	UTC (-8)
<input checked="" type="checkbox"/>		MS	Mississippi	UTC (-6)
<input type="checkbox"/>		TX	Texas	UTC (-6)
<input type="checkbox"/>		HI	Hawaii	UTC (-10)
<input type="checkbox"/>		GU	Guam	UTC (+10)

*Ilustración 5. Previsualización de la tabla State con algunas de sus propiedades. Fuente: Elaboración propia*

- **Cart\_box:** Contiene toda la información relacionada a la caja que incluye los productos que un usuario compra, el total, la cantidad, el agricultor, la compra a la que está relacionada, entre otros (ver ilustración 6).



Ver	10	Registros	Modelo	cart_box				Busc
<input type="checkbox"/>	KEY	product_group (REFERENCE-> product_group)	grade (REFERENCE-> grade)	grower (REFERENCE-> grower)	customer (REFERENCE-> customer)	purchase (REFERENCE-> purchase)	price (FLOAT)	
<input type="checkbox"/>							69.6	
<input type="checkbox"/>							100.48	
<input type="checkbox"/>							91.4	
<input type="checkbox"/>							173.68	
<input type="checkbox"/>							136.39	
<input type="checkbox"/>							141.6	
<input type="checkbox"/>							152.31	
<input type="checkbox"/>							102.8	
<input type="checkbox"/>							90.52	
<input type="checkbox"/>							120.32	

*Ilustración 6. Previsualización de un fragmento de la tabla Cart\_box con algunas propiedades. Fuente: Elaboración propia*

- **Cart\_box\_item:** Contiene toda la información de un producto comprado, por ejemplo: la variedad, el grupo al que pertenece un producto, el precio, la cantidad, la caja a la cual pertenece (ver ilustración 7).

Ver 10 Registros Modelo cart\_box\_item

		inventory (REFERENCE-> inventory)	product_group (REFERENCE-> product_group)	grower (REFERENCE-> grower)	variety (REFERENCE-> variety)	quantity (INT)	ci (R
<input type="checkbox"/>	KEY					4	
<input type="checkbox"/>						2	
<input type="checkbox"/>						3	
<input type="checkbox"/>						1	
<input type="checkbox"/>						2	
<input type="checkbox"/>						2	
<input type="checkbox"/>						2	
<input type="checkbox"/>						1	
<input type="checkbox"/>						5	
<input type="checkbox"/>						3	

Ilustración 7. Previsualización de un fragmento de la tabla *Cart\_box\_item* con algunas propiedades.  
Fuente: Elaboración propia

- **Customer:** Esta tabla almacena la información de los clientes, como el nombre de la empresa, dirección, detalles de contacto y preferencias. También incluye datos relacionados con la actividad comercial, como el número de órdenes, el total comprado y fechas importantes como la fecha de registro y la última compra (ver ilustración 8).

<input type="checkbox"/>	KEY	company_name (STRING)	street (STRING)	state (REFERENCE-> state)	zipcode (STRING)	city (STRING)	country (REFERENCE-> country)
<input type="checkbox"/>		Massonia					
<input type="checkbox"/>		Blooms By Nat					
<input type="checkbox"/>		Hello Daisy Flower Farm				Canton	
<input type="checkbox"/>		EnBloom Design				Jonesboro	
<input type="checkbox"/>		ramosbydulce					
<input type="checkbox"/>		Eternal Springs					
<input type="checkbox"/>							
<input type="checkbox"/>		9.15 Floral designs and gifts				Greenville	
<input type="checkbox"/>		Kathys Floral Designs					
<input type="checkbox"/>		Boutique Market					

*Ilustración 8. Previsualización de un fragmento de la tabla Customer con algunas propiedades e información sensible oculta. Fuente: Elaboración propia*

- **Variety:** Contiene información sobre las variedades de productos que se ofrecen, incluyendo el nombre de la variedad, su código, precio inicial, y detalles relacionados con la logística, como las semanas de disponibilidad y el número de tallos por manojo. Además, puede indicar si una variedad está inactiva o tiene una etiqueta de día festivo asociada (ver ilustración 9).

Ver	10	Registros	Modelo	variety				
			variety_name (STRING)	color (REFERENCE-> color)	product_group (REFERENCE-> product_group)	is_combo (BOOLEAN)	vbn_code (STRING)	cu (ST
<input type="checkbox"/>	KEY		Cumbia			<input type="checkbox"/>		
<input type="checkbox"/>			Gerbera Mini Assorted On Row Holstein			<input checked="" type="checkbox"/>		
<input type="checkbox"/>			Pink Jesolo			<input type="checkbox"/>		
<input type="checkbox"/>			Beltaard Red (tinted/dyed)			<input type="checkbox"/>		
<input type="checkbox"/>			Dried Flowers Mothers Day Assorted Box (7 Varieties - 360 Stems)			<input checked="" type="checkbox"/>		
<input type="checkbox"/>			Jungle Trial			<input type="checkbox"/>		
<input type="checkbox"/>			OASIS Rustic Wire , Brown, 18 ga, 70 ft. roll (10 pieces) 40-02642			<input type="checkbox"/>		
<input type="checkbox"/>			David Harum			<input type="checkbox"/>		
<input type="checkbox"/>			Seeded Brown (Tinted/ Dyed)			<input type="checkbox"/>		
<input type="checkbox"/>			Flower care Floralife CRYSTAL CLEAR Flower Food 300 Powder, 10 Lb. (1 piece) 82-03061			<input checked="" type="checkbox"/>		

Ilustración 9. Previsualización de un fragmento de la tabla variety con algunas propiedades. Fuente: Elaboración propia

- **Product\_Group:** Contiene los grupos de productos relacionados, como categorías de flores. Almacena el nombre común y botánico de cada grupo, así como información sobre su visibilidad y relevancia, como si está resaltado o si ha sido cancelado. También puede incluir detalles específicos de la logística, como el código HTS y el precio máximo de la caja (ver ilustración 10).

<div> <span>Ver</span> <span>10</span> <span>Registros</span> <span>Modelo</span> <span>product_group</span> <span>Buscar en</span> </div>								
<input type="checkbox"/>	KEY	category (REFERENCE-> category)	common_name (STRING)	botanical_name (STRING)	index_order (INT)	group_type (REFERENCE-> group_type)	is_combo (BOOLEAN)	hts_code (STRING)
<input type="checkbox"/>			Photinia	photinia	3		<input type="checkbox"/>	
<input type="checkbox"/>			Phylica	Phylica	3		<input type="checkbox"/>	
<input type="checkbox"/>			Herbs	herbs	3		<input type="checkbox"/>	
<input type="checkbox"/>			Other Containers	other containers	23		<input type="checkbox"/>	
<input type="checkbox"/>			Rudbeckia	Rudbeckia	2		<input type="checkbox"/>	
<input type="checkbox"/>			Oncidium	oncidium orchids	8		<input type="checkbox"/>	
<input type="checkbox"/>			Dill	Anethum	2		<input type="checkbox"/>	
<input type="checkbox"/>			Sleeves & wraps	Sleeves & wraps	3		<input type="checkbox"/>	
<input type="checkbox"/>			Cattails	cortaderia	3		<input type="checkbox"/>	
<input type="checkbox"/>			Rose Petals	rose petals	1		<input type="checkbox"/>	

*Ilustración 10. Previsualización de un fragmento de la tabla product\_group con algunas propiedades e información sensible oculta. Fuente: Elaboración propia*

En las ilustraciones anteriores, se encuentra la información relevante del usuario que hizo la compra, su dirección, sus preferencias, fechas de compras, registro, productos y que tipo compró, sus cantidades. A partir de estos datos, se puede comprender mejor la información que está involucrada en el proceso de compra de un cliente que es útil para realizar diferentes análisis posteriormente.

- **Preparación de los datos:**

Luego del paso de la comprensión de los datos, se realiza una extracción de los mismos a través de SBXCloud, lo cual permite a través de sus llaves foráneas relacionar los productos con las compras y los usuarios que realizaron estas. Para la extracción de los datos se tiene en cuenta el histórico de compras del año 2023 almacenados en un fichero CSV llamado 'all\_2023\_ibf\_data.csv' que es el resultado de la unión de las tablas mencionadas en el paso anterior (ver ilustración 11).

cart_box_item_cart_box_customer_company_name	cart_box_item_cart_box_customer_confirm_profile	cart_box_item_cart_box_customer_country
RoseDrops&Wishes	2021-07-27T21:33:52	
Christine Laurentius floral designs	2021-09-01T16:21:30	
A-1 Floral Design Studio	2021-06-21T19:56:45	
Lauras Garden	2021-06-21T15:08:42	
Mary Anns Floral and Gifts	2022-08-05T20:53:03	
Clementine Flowers	2021-06-25T00:24:48	
Sugarberry Blooms	2021-06-21T15:03:14	
Flower House KC	2022-01-24T20:37:41	

Ilustración 11. Previsualización de un fragmento dataset “all\_2023\_ibf\_data.csv” con algunas propiedades e información sensible oculta. Fuente: Elaboración propia

El conjunto de datos contiene inicialmente 155807 filas y 157 características. Luego, se procede al análisis de los datos con el uso de la librería Pandas en Python y se carga en un *dataframe* (ver Apéndice).

En primer lugar, se realiza el preprocesamiento de los datos, que incluye la selección, limpieza y transformación de los datos (ver Apéndice). En la fase de selección de los datos, se escogen las características o variables que estarán directamente involucradas en el desarrollo del proyecto, éstas son transversales para todos los usuarios de la plataforma, sin importar si es nuevo o antiguo (ver Apéndice). Las características seleccionadas se describen a continuación:

- **cart\_box\_item\_cart\_box\_purchase\_state\_state**: Estado en el que reside un usuario asociado a una compra para su dirección de entrega.
- **cart\_box\_item\_variety\_variety\_name**: Nombre de la variedad de un ítem en la caja.
- **cart\_box\_item\_variety\_KEY**: Identificador de la variedad de un ítem en la caja.
- **cart\_box\_item\_length**: Longitud de un ítem, por ejemplo: 60 → 60Cm.
- **cart\_box\_item\_product\_group\_common\_name**: Nombre común del grupo de productos al que pertenece el ítem.
- **cart\_box\_item\_cart\_box\_customer\_margin**: Margen del cliente en el negocio.
- **cart\_box\_item\_cart\_box\_customer\_tier\_sbx**: categorización del cliente en el negocio.
- **cart\_box\_item\_cart\_box\_customer\_KEY**: Identificador del cliente del en la base de datos.

- **`cart_box_item_cart_box_customer_business`**: Tipo de negocio del cliente.
- **`cart_box_item_cart_box_customer_events_per_year`**: Eventos por año del cliente.
- **`cart_box_item_cart_box_customer_stores_quantity`**: Cantidad de tiendas del cliente.
- **`cart_box_item_cart_box_customer_employees_quantity`**: Cantidad de empleados del cliente.
- **`cart_box_item_cart_box_customer_spend_per_week`**: Gasto por semana del cliente.

En la ilustración 12 se muestran las características en la tabla que contiene el conjunto de datos.

	<code>cart_box_item_cart_box_purchase_state_state</code>	<code>cart_box_item_variety_variety_name</code>	<code>cart_box_item_variety_KEY</code>	<code>cart_box_item_length</code>	<code>cart_box_item_product_g</code>
0	Pennsylvania	Dark X-Pression	469a7ea9-d706-4e90-adc4-f303e010ce0c	40.0	
1	Missouri	Pink Pigeon	4da78552-4952-4971-8ea5-f9ce645398e9	50.0	
2	Texas	Variegated	36b75d67-defd-4b5e-a3df-218a67d52d21	40.0	
3	New York	Deep Purple	7700981c-8639-4a6b-ac12-8f4577aac4ff	60.0	
4	Minnesota	Green	7477eac8-f207-41d6-b543-70c9845890b7	40.0	

*Ilustración 12. Previsualización de un fragmento del conjunto de datos con las características seleccionadas. Fuente: Elaboración propia.*

Se obtiene un *dataset* con 155807 filas, pero con las 13 características seleccionadas. A continuación, se realiza una transformación de los datos para que en los pasos siguientes se pueda realizar el análisis de los mismos, se hace de la siguiente manera:

1. Se realiza una copia del conjunto de datos inicial, para evitar su alteración para posteriores usos y así poder trabajar libremente con la copia.
2. Se renombraron las columnas iniciales para facilitar la comprensión:
  - a) La columna '`cart_box_item_cart_box_purchase_state_state`' indica el estado de compra y se renombra como '**`state`**'.

- b) La columna '**cart\_box\_item\_cart\_box\_customer\_margin**' indica el margen del cliente y se renombra como '**margin**'.
- c) La columna '**cart\_box\_item\_cart\_box\_customer\_tier\_sbx**' indica el nivel de cliente y se renombra como '**tier**'.
- d) La columna '**cart\_box\_item\_cart\_box\_customer\_\_KEY**' indica una clave del cliente y se renombra como '**customer\_key**'.
- e) La columna '**cart\_box\_item\_variety\_variety\_name**' indica el nombre de la variedad y se renombra como '**variety\_name**'.
- f) La columna '**cart\_box\_item\_variety\_\_KEY**' indica una clave de variedad y se renombra como '**variety\_key**'.
- g) La columna '**cart\_box\_item\_length**' indica la longitud y se mantiene igual.
- h) La columna '**cart\_box\_item\_product\_group\_common\_name**' indica el nombre común del grupo de productos y se renombra como '**product\_group**'.
- i) La columna '**cart\_box\_item\_cart\_box\_customer\_business**' indica el negocio del cliente y se renombra como '**business**'.
- j) La columna '**cart\_box\_item\_cart\_box\_customer\_events\_per\_year**' indica los eventos por año del cliente y se renombra como '**events\_per\_year**'.
- k) La columna '**cart\_box\_item\_cart\_box\_customer\_stores\_quantity**' indica la cantidad de tiendas del cliente y se renombra como '**stores\_quantity**'.
- l) La columna '**cart\_box\_item\_cart\_box\_customer\_employees\_quantity**' indica la cantidad de empleados del cliente y se renombra como '**employees\_quantity**'.
- m) La columna '**cart\_box\_item\_cart\_box\_customer\_spend\_per\_week**' indica el gasto por semana del cliente y se renombra como '**spend\_per\_week**'.

Luego de transformar las columnas, se realiza el análisis de valores faltantes (ver ilustración 13), se identifican cuantos valores faltantes existen y se obtiene el porcentaje.



```
state - 0%
variety - 0%
variety_key - 0%
length - 0%
product_group - 0%
margin - 0%
tier - 0%
customer_key - 0%
business - 0%
events_per_year - 0%
stores_quantity - 2%
employees_quantity - 0%
spend_per_week - 67%
```

*Ilustración 13. Porcentaje de datos faltantes en el conjunto de datos por columna. Fuente: Elaboración propia.*

Al observar la ilustración 13, se muestra que las columnas *stores\_quantity* y *spend\_per\_week* contiene 2% y 67% de datos faltantes respectivamente. Por lo tanto, se tratan estos dando el dar valor predeterminado a esas columnas para facilitar su procesamiento, eso se hace al sustituir el valor faltante por la palabra '*Unknown*' para las columnas de tipo categórico y 0.0 para las columnas de tipo numérico. Se obtiene como resultado, 0% de valores faltantes en todas las columnas del conjunto de datos, como se muestra en la ilustración 14.

```
state - 0%
variety - 0%
variety_key - 0%
length - 0%
product_group - 0%
margin - 0%
tier - 0%
customer_key - 0%
business - 0%
events_per_year - 0%
stores_quantity - 0%
employees_quantity - 0%
spend_per_week - 0%
```

*Ilustración 14. Porcentaje de datos faltantes en el conjunto de datos por columna después de la imputación por columna. Fuente: Elaboración propia.*

Finalmente, se procede a realizar un agrupamiento de los datos por usuario, esto se hace para tener un registro único de uno de ellos y así realizar un análisis de sus características y efectuar una comparación más precisa entre ellos. Durante la agrupación se eliminaron columnas como "*product\_group*", "*variety*", "*variety\_key*" y "*length*", porque no es relevante la información de los productos sino únicamente de los usuarios que realizaron compras en el año 2023 y sus características o atributos en común (ver figura 15).

customer_key	state	tier	business	events_per_year	spend_per_week	stores_quantity	employees_quantity	margin
000d1beb-2fa6-44cf-a68a-54c369090875	Mississippi	tier_2	Florist shop	10-24	100–499	1.0	4	36
007a32b6-b1a7-4c9d-9cb3-937ecc1f603d	South Carolina	tier_3	Other	< 10	< \$100	1.0	3	42
009a8305-205b-4337-a34a-8d423a8c7704	Arkansas	tier_1	Event planner (only), not a floral designer or...	> 60	Unknown	1.0	5	31
00a5424e-601d-486b-81b3-0f7db37b3a32	Wisconsin	tier_2	Florist shop	10-24	Unknown	1.0	1	36
01340b6f-7b0a-4e99-b088-efd37b549342	Virginia	tier_3	Wedding & event floral designer only (not a fl...	10-24	Unknown	0.0	2	42

Ilustración 15. Resultado del agrupamiento del conjunto de datos. Fuente: Elaboración propia

Como resultado de esta agrupación, el conjunto de datos pasa a tener 2238 filas que corresponden a la cantidad de usuarios que realizaron compras en el año 2023 y 8 características estas son:

- *Customer\_key*
- *State*
- *Tier*
- *Bussines*
- *Event\_per\_year*
- *Spend\_per\_week*
- *Stores\_Quantity*
- *Employees\_Quantity*
- *Margin*

Estas características son las preferencias de los usuarios registradas en la base de datos. Ahora la variable *customer\_key* paso a ser el índice de los datos, lo cual permite que no existan valores duplicados en los usuarios.

- **Modelado:**

Para el modelado, se basó en la retroalimentación implícita, teniendo en cuenta el comportamiento de los usuarios en el pasado con respecto al histórico de compras, se adoptó como característica implícita la frecuencia de compras de cada cliente por producto, evaluando así la confianza basada en este valor. Para ello se toma el conjunto de datos inicial, se realiza una copia y se crea un nuevo conjunto de datos llamado “*products\_by\_freq*”, en donde se crea una columna concatenada llamada “*product\_name*”, la cual estaba compuesta por “*product\_group*”, “*variety*”, y “*length*”,

está será usada como ID para identificar junto a la columna “*customer\_key*” cuantas veces fue comprado un producto (ver ilustración 16).

	<i>customer_key</i>	<i>product_name</i>	<i>count</i>
27811	[REDACTED]	Mini Carnation (Spray) Hamada 50cm	196
27812	[REDACTED]	Mini Carnation (Spray) Imagine 50cm	196
27813	[REDACTED]	Mini Carnation (Spray) Pigeon 50cm	196
27814	[REDACTED]	Mini Carnation (Spray) Pink Pigeon 50cm	196
27815	[REDACTED]	Mini Carnation (Spray) Rony 50cm	196

Ilustración 16. Previsualización de un fragmento del conjunto de datos con las frecuencias por compras por usuario e información sensible oculta. Fuente: Elaboración propia

Luego de obtener la frecuencia de compra de los productos por usuario, se aplica la técnica conocida como "Mínimos Cuadrados Alternos" o "*Alternating Least Squares algorithm*", esta técnica está basada en factorización matricial, lo cual permite descomponer una matriz que contiene la relación de usuario-artículo en dos matrices que tienen dimensiones inferiores, siendo una de ellas para usuarios y otra para los elementos. En esta implementación se obtendrá una nueva matriz que contiene los ID de los productos y clientes cuya multiplicación da como un resultado aproximado el puntaje de la interacción entre ellos (Loukili et al., 2023)

Inicialmente, para el proceso de creación de la matriz que tendrá la relación entre cliente y producto se crearon dos columnas de tipo numérica llamadas, "*customer\_id*" e "*item\_id*", en las cuales se asignara en valor numérico un ID para los productos y los clientes con ayuda de la función “*enumerate*” para obtener el id de cada cliente y luego usar la función “*map*” para agregar por cada fila el valor de cada id. Esto con el fin de crear nuestra matriz de factorización (ver ilustración 17).

	<i>customer_key</i>	<i>product_name</i>	<i>count</i>	<i>customer_id</i>	<i>item_id</i>
27811	[REDACTED]	Mini Carnation (Spray) Hamada 50cm	196	0	0
27812	[REDACTED]	Mini Carnation (Spray) Imagine 50cm	196	0	1
27813	[REDACTED]	Mini Carnation (Spray) Pigeon 50cm	196	0	2
27814	[REDACTED]	Mini Carnation (Spray) Pink Pigeon 50cm	196	0	3
27815	[REDACTED]	Mini Carnation (Spray) Rony 50cm	196	0	4

Ilustración 17. Previsualización de un fragmento del conjunto de datos con las frecuencias por compras por usuario con los ID de referencia para el usuario y producto con información sensible oculta. Fuente: Elaboración propia

Luego, se procede a crear la matriz dispersa que relaciona los clientes con los productos. En las cuales se asigna como valores las frecuencias de compras que tiene cada cliente por cada producto. Está matriz de dispersión se realizó con la función “*coo\_matrix*” de *scipy.sparse* para ser utilizada en el modelo con el algoritmo Mínimos cuadrados alternos provisto por la librería “*implicit*” a través de la función “*AlternatingLeastSquares*”

En este modelo, se tiene como objetivo conocer la interacción entre un usuario y un elemento, se utilizan los valores de una matriz dispersa que representan si un usuario

ha tenido interacción con un elemento o no la ha tenido. Los valores analizados en esas matrices son conocidos como confianza, en este caso la confianza será la frecuencia de compras de un usuario basado en un producto. Para ello se definen tres hiperparámetros:

1. *Factors*: Los factores o factores latentes son propiedades ocultas que están implícitas en el *dataset* que infieren en atributos "ocultos" que no son visibles, pero pueden influir en porque gusta más un producto o no. En este caso se inicializa con el valor en 20. Se define este valor para evitar sobre ajuste debido que se tiene en cuenta muchos atributos a la hora de analizar los productos y un mayor costo de cómputo.
2. *Regularization*: Este hiperparámetro permite mantener un equilibrio entre el sobreajuste y el sub-ajuste del modelo, ya que, si el valor es muy alto, hace el modelo más simple, pero puede que no se tenga en cuenta las relaciones y si es muy bajo puede haber riesgo de sobre ajuste y no realice una generalización entre los productos.
3. *Random\_state*: Permite colocar una semilla aleatoria para reproducir los experimentos en el mismo orden y mantener la consistencia en ellos

Para obtener el modelo con los mejores parámetros posibles, se realizó la implementación de un proceso de validación cruzada, en donde se utilizó la función *"train\_test\_split"* de *sklearn* para dividir el conjunto de datos utilizado para reservar una parte para el entrenamiento del modelo y la otra parte para la validación de este.

Finalmente, luego de definir estos parámetros para el modelo se hace el entrenamiento a través de la función *"fit"*, la cual recibió el conjunto de datos de entrenamiento mencionados anteriormente, y se procede a crear una función llamada *"getProductRecommendations"*, esta función recibe la llave (*key*) de un cliente (*customer*) y hace uso de la función *"recommend"* que provee el modelo, la cual genera un grupo de recomendaciones basándose en un usuario o un grupo de usuarios. En el cual, se pasa como parámetro el número de recomendaciones que se quiere obtener para el cliente, el cual es definido como  $K = 10$ , siendo  $K$  el número de recomendaciones.

Finalmente, se le solicitó al modelo la recomendación de productos para una llave (*key*) de cliente tomada del conjunto de datos "45193aab-887a-4590-be6e-6558691e63b1" donde se obtuvo el siguiente listado de recomendaciones (ver ilustración 18):

	product_id	score
0	Delphinium Planet Blush Pink (white few blush ...	0.332833
1	Hebes Green 40cm	0.275237
2	Anemone Bicolor White -Blush shade 35cm	0.269122
3	Delphinium Planet Blush Pink (white few blush ...	0.268780
4	Anemone Bayola (blue-purple) 30cm	0.263806
5	Dianthus Star Snow Tessino 50cm	0.254803
6	Dianthus Raffine Petit Faye 50cm	0.245794
7	Dianthus Solomio Fiorino Leo 50cm	0.225754
8	Scabiosa Bon Bon Vainilla 50cm	0.218752
9	Ranunculus Elegance White 30cm	0.208791

*Ilustración 18. Conjunto de datos que contiene las recomendaciones a un cliente con sus puntajes.  
Fuente: Elaboración propia*

Finalmente se obtiene las 10 recomendaciones para un usuario, basado en una retroalimentación implícita que no tuvo en cuenta ningún valor más allá de la frecuencia de compras de un usuario, en el cual, se puede evidenciar carencia de más contexto que dejan ver ciertas desventajas de este tipo de evaluación cuando no se cuenta con un soporte de valoraciones explícitas, es importante mencionar que se pueden manejar pesos en los valores números obtenidos para manejar situaciones donde los valores asignados como confianza son ruidosos, como por ejemplo, un cliente que realizó un encargo masivo de algunos productos en específico porque un familiar le pidió el favor que los comprara con su cuenta, entre otros.

Así mismo, otra desventaja con respecto al uso de este modelo y algoritmo es que solo tiene en cuenta los valores dentro del conjunto de datos propuesto, por lo cual un usuario nuevo, sin comportamiento de compras puede entrar en un punto frío y no obtener recomendaciones para ello, por lo tanto, se realizó un complemento para el modelo de recomendación, permitiendo así abarcar más clientes y asegurarle al negocio que todos los usuarios están obteniendo sugerencias para sus compras en el sitio web.

Para abordar este complemento, se utiliza la técnica de filtrado colaborativo basado en la selección del vecino más cercano. Esto consiste en la comparación de características elegidas previamente, en este punto son consideradas como las preferencias, entre un usuario objetivo y un conjunto de usuarios y en base a ello se toma el histórico de compras del usuario con mayor grado de similitud y a partir de las compras realizadas por este último se realiza la recomendación.

Para calcular la similitud de los usuarios existen diferentes métricas que dependen del conjunto de datos, tales como:

- Coeficiente de Sorensen-Dice.
- Índice de Jaccard.
- Distancia euclidiana.

- Coeficiente de correlación de Pearson.
- Similitud del coseno.

Cada métrica tiene sus ventajas y desventajas, ya que la similitud es una medida que permite entender el grado en el cual dos elementos son semejantes, generalmente se utiliza el intervalo 0 a 1, donde 0 significa que no hay similitud y 1 que la similitud es exacta (Baxla, 2014).

De esta manera, en la base de datos de *iBuyFlowers* no existe información relacionada a la valoración general de la experiencia de compras que permita medir niveles de satisfacción de los usuarios, como en otros casos, donde los *e-commerce* cuentan con comentarios, valoraciones de medición para los productos y las compras realizadas.

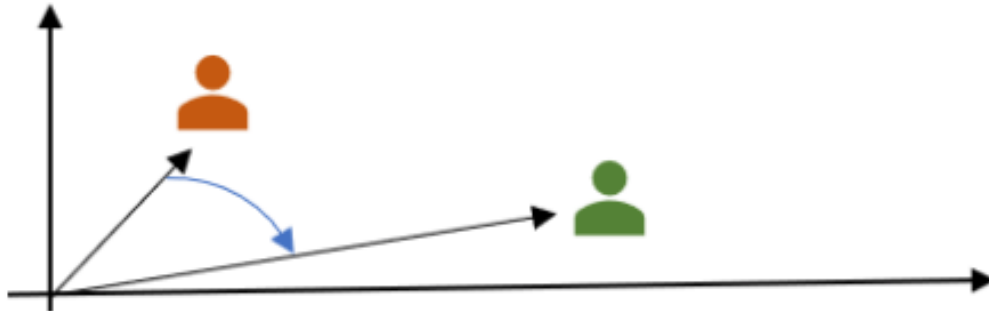
Por consiguiente, para el módulo de recomendación propuesto para *iBuyFlowers*, se utiliza como métrica la similitud de coseno, la escogencia se basó haciendo un análisis del conocimiento del negocio, la comprensión de los datos y el tiempo de ejecución (ver tabla 1).

Number of Users	Cosine based similarity(sec)	Extended Jaccard based similarity(sec)	Adjusted cosine based similarity(sec)	Correlation based similarity(sec)
1000	56.9473	57.089	290.110	300.733
4000	59.138	57.373	292.613	301.065
8000	62.916	59.069	293.769	303.166
12000	63.412	60.156	295.112	305.794
16000	64.533	62.826	296.996	306.835
20000	68.090	67.612	298.119	308.617
24000	68.209	67.882	299.357	309.234
24893	69.127	68.215	301.021	310.109

Tabla 1. Tiempo de ejecución para cada medida de similitud. Fuente: (Khatter et al. 2021)

Basado en esta tabla se puede observar que la similitud del coseno y el índice de Jaccard son los que menor tiempo de ejecución toman al momento de evaluar un gran número de usuarios, por ende, estas son las mejores métricas según (Khatter et al., 2021) para un sistema de recomendación.

Para realizar el cálculo de similitud del coseno entre dos usuarios se mide el ángulo entre sus vectores clasificados, indicando un mayor valor de afinidad cuando es más pequeño el ángulo y cuando es más grande, la afinidad es menor (ver ilustración 19).



*Ilustración 19. Cálculo de ejemplo de la similitud entre dos usuarios representado en el plano y mostrando el ángulo entre ellos. Fuente: (Munkholm et al., 2024)*

Para incluir la similitud del coseno en el módulo de recomendación propuesto, se tienen en cuenta los siguientes pasos:

1. Para realizar las recomendaciones de los productos a los usuarios, se toma en cuenta el conjunto de datos obtenidos en la fase anterior y se agrega un usuario nuevo (ver Apéndice). Este será el objetivo del análisis para obtener una recomendación, para ello se agrega al conjunto de datos existente con las siguientes características, en este caso se tiene:
  - a. *State*: "Texas".
  - b. *Tier*: "Tier\_2".
  - c. *Business*: "Wedding & evento floral designer only (not a florist)".
  - d. *events\_per\_year*: "25 - 59".
  - e. *spend\_per\_week*: "Unknown".
  - f. *stores\_quantity*: 1
  - g. *employees\_quantity*: 8
  - h. *margin*: 31
  - i. *customer\_key*: "c7792ee4-7303-4919-9521-1f221d5bdd96"
2. Luego de realizar la inserción del nuevo usuario y verificar que se encuentra en el *dataset* de *customers*, el conjunto de datos queda de la siguiente manera (ver ilustración 20).

	state	tier	business	events_per_year	spend_per_week	stores_quantity	employees_quantity	margin
000d1beb-2fa6-44cf-a68a-54c369090875	Mississippi	tier_2	Florist shop	10-24	100–499	1.0	4	36
007a32b6-b1a7-4c9d-9cb3-937ecc1f603d	South Carolina	tier_3	Other	< 10	< \$100	1.0	3	42
009a8305-205b-4337-a34a-8d423a8c7704	Arkansas	tier_1	Event planner (only), not a floral designer or...	> 60	Unknown	1.0	5	31
00a5424e-601d-486b-81b3-0f7db37b3a32	Wisconsin	tier_2	Florist shop	10-24	Unknown	1.0	1	36
01340b6f-7b0a-4e99-b088-efd37b549342	Virginia	tier_3	Wedding & event floral designer only (not a fl...	10-24	Unknown	0.0	2	42
...	...	...	...	...	...	...	...	...
ffb513af-c969-4616-97f3-428de2c37032	Minnesota	tier_1	Florist shop	> 60	Unknown	1.0	6	36
fff1017d-beeb-4b75-9bcf-ba957544ca6b	Tennessee	tier_2	Wedding & event floral designer only (not a fl...	10-24	Unknown	1.0	2	31
fff3ca77-b401-4329-a6c1-df4a2ed3462c	New York	tier_3	Wedding & event floral designer only (not a fl...	< 10	100–499	0.0	1	42
fff60109-7599-4cc1-8c42-7f44cc0bcd8	Nevada	tier_3	Florist shop	10 - 24	Unknown	0.0	2	42
c7792ee4-7303-4919-9521-1f221d5bdd96	Texas	Tier_2	Wedding & event floral designer only (not a fl...	25 - 59	Unknown	1	8	31

2239 rows x 8 columns

*Ilustración 20. Fragmento del conjunto de datos luego de la inserción del nuevo usuario visualizado al final de los datos. Fuente: Elaboración propia*

3. A continuación, se procede a convertir las columnas categóricas en binarias para aplicar la similitud de coseno, dado que se obtiene un conjunto de datos netamente numérico, obteniendo así, los vectores que serán utilizados para obtener los valores de semejanza, para conseguir esto se realiza la implementación de la función `get_dummies` de la librería `pandas` (ver Apéndice). Esta función, facilita la codificación *One Hot Encoding*, dando como resultado un nuevo *dataset* donde cada columna representa una categoría y un valor, donde 1 es equivalente a la presencia esa categoría y un 0 su ausencia (ver ilustración 21).



	stores_quantity	employees_quantity	margin	state_Alabama	state_Alaska	state_Arizona	state_Arkansas	state_California	state_Colorado	state
000d1beb-2fa6-44cf-a68a-54c369090875	1.0	4	36	0	0	0	0	0	0	
007a32b6-b1a7-4c9d-9cb3-937ecc1f603d	1.0	3	42	0	0	0	0	0	0	
009a8305-205b-4337-a34a-8d423a8c7704	1.0	5	31	0	0	0	1	0	0	
00a5424e-601d-486b-81b3-0f7db37b3a32	1.0	1	36	0	0	0	0	0	0	
01340b6f-7b0a-4e99-b088-efd37b549342	0.0	2	42	0	0	0	0	0	0	
...	...	...	...	...	...	...	...	...	...	...
ffb513af-c969-4616-97f3-428de2c37032	1.0	6	36	0	0	0	0	0	0	
fff1017d-becb-4b75-9bcf-ba957544ca6b	1.0	2	31	0	0	0	0	0	0	
fff3ca77-b401-4329-a6c1-df4a2ed3462c	0.0	1	42	0	0	0	0	0	0	
fff60109-7599-4cc1-8c42-7f44cc0bcd8	0.0	2	42	0	0	0	0	0	0	
c7792ee4-7303-4919-9521-1f221d5bdd96	1	8	31	0	0	0	0	0	0	

2239 rows x 84 columns

Ilustración 21. Visualización parcial del conjunto de datos luego de la aplicación de la función `get_dummies`. Fuente: Elaboración propia

Al aplicar la función `get_dummies`, se obtiene un nuevo *dataset* que contiene 2239 filas y 84 columnas, que representan los valores numéricos de todas las categorías involucradas en este proceso.

4. Se procede a aplicar la similitud del coseno, se usa la función `cosine_similarity` de Scikit-learn (ver Apéndice), esta calcula la similitud de coseno entre dos conjuntos de vectores. Para facilitar la comprensión de los datos y obtener mejores resultados se coloca como index y columnas los key o ID de los *customer* (usuarios) y así poder realizar un mejor filtrado de sus valores y obtener los resultados más cercanos entre sí (ver Ilustración 19).

	000d1beb- 2fa6-44cf- a68a- 54c369090875	007a32b6- b1a7-4c9d- 9cb3- 937ecc1f603d	009a8305- 205b-4337- a34a- 8d423a8c7704	00a5424e- 601d-486b- 81b3- 0f7db37b3a32	01340b6f- 7b0a-4e99- b088- efd37b549342	01b79024- b222-44d6- be1e- d59b39fca976	01db02a5- dd1f-414c- b986- 3ba5b5becf2c	01f4aa00- 2167-4b97- adad- 88be3aa656c2	0234cb76- 9bb1-48e5- 9534- b088bc64047f	026fc 14ad- 8f13c30e
000d1beb- 2fa6-44cf- a68a- 54c369090875	1.000000	0.995919	0.994368	0.995056	0.994986	0.995942	0.993339	0.997876	0.995641	0.995
007a32b6- b1a7-4c9d- 9cb3- 937ecc1f603d	0.995919	1.000000	0.992137	0.995724	0.997186	0.996193	0.995978	0.997477	0.997186	0.995
009a8305- 205b-4337- a34a- 8d423a8c7704	0.994368	0.992137	1.000000	0.987761	0.990044	0.993428	0.987829	0.994698	0.990044	0.995
00a5424e- 601d-486b- 81b3- 0f7db37b3a32	0.995056	0.995724	0.987761	1.000000	0.997407	0.994560	0.997259	0.996393	0.998065	0.995
01340b6f- 7b0a-4e99- b088- efd37b549342	0.994986	0.997186	0.990044	0.997407	1.000000	0.995596	0.996136	0.996354	0.998872	0.995
...	...	...	...	...	...	...	...	...	...	...
0c06aba0- 7225-451c- 85e8- 98114c2b6a36	0.998076	0.995683	0.993935	0.996711	0.995846	0.996412	0.994864	0.998274	0.996606	0.995
0c2581f1- 30a8-42bf- a75d- 56947d22094c	0.994986	0.997186	0.990044	0.997407	0.999436	0.995596	0.996899	0.996354	0.998872	0.995
0c368a87- 749b-4703- 9844- db374b8b6573	0.993193	0.995637	0.987190	0.997476	0.997928	0.994940	0.996886	0.995501	0.997320	0.995
0c3cb544- ff60-4f68- a4b2- da378335a39b	0.995431	0.997959	0.993935	0.994051	0.996606	0.996412	0.994864	0.997517	0.995846	0.995
0c7732a1- 2c93-44c6- a5e2- e8b713721878	0.992930	0.997186	0.987585	0.997312	0.998307	0.994636	0.997462	0.995792	0.997743	0.995
100 rows × 2239 columns										

Ilustración 22. Visualización parcial del conjunto de datos luego de la aplicación de la similitud de coseno con 100 Datos. Fuente: Elaboración propia

Finalmente, se obtiene la similitud de coseno, se produce como resultado un *dataframe* como se muestra en la ilustración 22, que está conformado por 2239 filas y 2239 columnas que es el valor total de los usuarios que realizaron compras en el año 2023. En los índices se ubica cada *customer key* (ID del usuario) y cada columna representa todos los usuarios (por *keys*) con los que se realizó la comparación y los valores de semejanza por cada uno de ellos.

## 4.4. Resultados

Como resultado del trabajo, se diseñó un modelo para un sistema de recomendaciones para una plataforma denominada “*iBuyFlowers*” basado en filtrado colaborativo enfocado en la retroalimentación implícita con el algoritmo de Mínimos cuadrados alterno y luego en la selección del vecino más cercado, para usuarios nuevos escogiendo como métrica la similitud del coseno.

Como se puede observar en la ilustración 18 se obtuvo un listado de 10 productos recomendados para el usuario basado en la frecuencia de compras que realizó a través de todo el año 2023 que fue tomado como valor de confianza, como resultado se muestra cómo el score o puntuación era bajo teniendo en cuenta que no se tuvo en cuenta nada adicional, debido a las limitantes de una retroalimentación implícita de no contar con interacción real medible de un cliente con el producto que permitiera mejorar las predicciones. Obteniendo como resultado el siguiente listado de productos previsualizados en un *dataframe* con sus imágenes (ver ilustración 23).




	product_name	image_preview
0	Delphinium Planet Blush Pink (white few blush touches) 50cm	
321	Hebes Green 40cm	
522	Anemone Bicolor White -Blush shade 35cm	

Ilustración 23. Previsualización de un fragmento del conjunto de datos resultante, conformada por nombre del producto y previsualización de la imagen. Fuente: Elaboración propia

Así mismo se obtuvo el resultado del complemento para los usuarios nuevos, donde basados en el cálculo de la similitud del coseno, se procede analizar el *dataframe* resultante con los valores. Para ello, se escoge un usuario del *dataset* como cliente, para conocer las compras que ha realizado en la plataforma.

Como referencia se toma el *customer\_key* del usuario objetivo para tomar de la tabla resultante de la ilustración 22, los usuarios con mayor similitud. Esto se realiza basado en los valores de similitud y los ID de los usuarios, se toma la fila representada por el índice del usuario objetivo y se extraen todos los valores asociados a él, excluyendo la comparación asociada a sí mismo, puesto que la similitud de coseno incluye la semejanza de todos vs todos, luego se ordenan los valores de mayor a menor y se obtendrán los 5 usuarios con mayor grado de similitud para luego comparar que tan cercanos o distantes son entre ellos y con el usuario objetivo, al realizar todo el proceso descrito anteriormente se obtienen los resultados que se observan en la ilustración 24.

c7792ee4-7303-4919-9521-1f221d5bdd96	
2f270530-43db-4e91-ad01-0eed173a68af	0.998109
aa702eff-11c6-4360-9cea-cb86e3a3ed75	0.998060
d743c173-4b89-41f2-ba35-be058cc6f847	0.997405
d15ee675-dc12-48e7-914c-d80974e2f003	0.997405
b57d0f5e-2177-416a-a919-85cbfe8aa8d7	0.997405

Ilustración 24. Valores de similitud con índice conformado por los ID de los usuarios y sus valores. Fuente: Elaboración propia

En esta ilustración se puede observar que tiene como índice los ID de los usuarios y la columna existente tiene como título el ID del usuario objetivo y los valores de sus filas son los valores de similitud, comparados con cada uno de los usuarios. Luego, se selecciona el primer usuario que tiene el mayor grado de similitud para hacer una comparación de características, tal como se muestra en la ilustración 25.

	state	tier	business	events_per_year	spend_per_week	stores_quantity	employees_quantity	margin
2f270530-43db-4e91-ad01-0eed173a68af	Washington	tier_2	Wedding & event floral designer only (not a fl...	25 - 59	Unknown	1.0	10	36
c7792ee4-7303-4919-9521-1f221d5bdd96	Texas	Tier_2	Wedding & event floral designer only (not a fl...	25 - 59	Unknown	1	8	31

Ilustración 25. Tabla conformada por Usuario objetivo y el usuario más similar a él. Fuente: Elaboración propia

En la ilustración 25 se observa que los valores de las características entre ambos usuarios son parecidos, pero se debe demostrar a través de la similitud del coseno entre ellos. Se toma en cuenta la ecuación siguiente:

$$\text{Cos}\theta = \frac{A \cdot B}{\|A\| \cdot \|B\|}$$

Donde A será el usuario identificado con el ID = "2f270530-43db-4e91-ad01-0eed173a68af".

Donde B será el usuario objetivo identificado con el ID = "c7792ee4-7303-4919-9521-1f221d5bdd96".

Para obtener los valores de la ecuación, primero se definen los vectores de A y B, estos son valores reales tomados de la aplicación de la similitud del coseno sobre el nuevo usuario y un usuario en el conjunto de datos:

$$\begin{aligned} A &= [1.0, 8.0, 31, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, \\ &\quad 0, \\ &\quad 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, \\ &\quad 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1] \\ B &= [1.0, 10, 36, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, \\ &\quad 0, \\ &\quad 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, \\ &\quad 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1] \end{aligned}$$

Luego, se calcula el producto escalar de los dos vectores A y B:

$$\begin{aligned} A \cdot B &= (1) \cdot (1) + (8) \cdot (10) + (31) \cdot (36) + (0) \cdot (0) + (0) \cdot (0) + (0) \cdot (0) + (0) \cdot (0) \\ &\quad + (0) \cdot (0) + (0) \cdot (0) + (0) \cdot (0) + (0) \cdot (0) + (0) \cdot (0) + (0) \cdot (0) + (0) \cdot (0) + (0) \cdot (0) \\ &\quad + (0) \cdot (0) + (0) \cdot (0) + (0) \cdot (0) + (0) \cdot (0) + (0) \cdot (0) + (0) \cdot (0) + (0) \cdot (0) + (0) \cdot (0) \\ &\quad + (0) \cdot (0) + (0) \cdot (0) + (0) \cdot (0) + (0) \cdot (0) + (0) \cdot (0) + (0) \cdot (0) + (0) \cdot (0) + (0) \cdot (0) \\ &\quad + (0) \cdot (0) + (0) \cdot (0) + (0) \cdot (0) + (0) \cdot (0) + (0) \cdot (0) + (0) \cdot (0) + (0) \cdot (0) + (0) \cdot (0) \\ &\quad + (1) \cdot (0) + (0) \cdot (0) + (0) \cdot (0) + (0) \cdot (0) + (0) \cdot (0) + (0) \cdot (1) + (0) \cdot (0) + (0) \cdot (0) \\ &\quad + (0) \cdot (0) + (0) \cdot (0) + (1) \cdot (0) + (0) \cdot (0) + (0) \cdot (1) + (0) \cdot (0) + (0) \cdot (0) + (0) \cdot (0) \\ &\quad + (0) \cdot (0) + (0) \cdot (0) + (0) \cdot (0) + (0) \cdot (0) + (0) \cdot (0) + (0) \cdot (0) + (0) \cdot (0) + (0) \cdot (0) \\ &\quad + (0) \cdot (0) + (0) \cdot (0) + (0) \cdot (0) + (0) \cdot (0) + (0) \cdot (0) + (0) \cdot (0) + (0) \cdot (0) + (0) \cdot (0) \\ &\quad + (1) \cdot (1) + (0) \cdot (0) + (0) \cdot (0) + (1) \cdot (1) + (0) \cdot (0) + (0) \cdot (0) + (0) \cdot (0) + (0) \cdot (0) \\ &\quad + (0) \cdot (0) + (0) \cdot (0) + (0) \cdot (0) + (0) \cdot (0) + (0) \cdot (0) + (0) \cdot (0) + (1) \cdot (1) = 1200. \end{aligned}$$

Ahora se calcula las magnitudes de los vectores  $\|A\|$  y  $\|B\|$ :

[illegible]

$$\|A\| = \sqrt{1031}$$

[illegible]

$$\| B \| = \sqrt{1042}$$

Se reemplaza los resultados obtenidos en la fórmula, de la siguiente manera:

$$\cos\theta = \frac{A.B}{\|A\|.\|B\|} = \frac{1200}{\sqrt{1031}.\sqrt{1042}} = \frac{1200}{\sqrt{1074302}} = \frac{1200}{1.036,485407519083}$$

$$\cos\theta = 1,157758702$$

Finalmente, se calcula el coseno del valor obtenido para obtener la similitud:

$$similarity(A, B) = \cos \theta = 0,999795851557893$$

Como se aprecia es un valor cercano al resultado calculado a través de la función *Cosine Similarity* de *sklearn* (ver Apéndice). El cual fue = 0.998109 con respecto al usuario más semejante comparado con el usuario objetivo. Permitiendo así visualizar los productos recomendados para este usuario nuevo basado en su vecino más cercano (ver ilustración 26).





	product_name	image_preview
0	Queen Annes Lace White QAL 70cm	
434	Roses Playa Blanca (Sometimes have subtle blush hints) 40cm	
1243	Ranunculus White 30cm	
1580	Ruscus Israeli 60cm	

Ilustración 26. Previsualización de un fragmento del conjunto de datos resultante para el nuevo usuario, conformada por nombre del producto y previsualización de la imagen. Fuente: Elaboración propia

## 4.5. Evaluación del modelo

Para la evaluación de modelo se utilizó la métrica *Mean Average Precision at k* ( $MAP@k$ ), a cuál es una medida utilizada para evaluar la precisión de un sistema de recomendación o un modelo de recuperación de información, tomando en cuenta la posición de los elementos relevantes en las recomendaciones. Se tomo en cuenta el ajuste de varios hiperparámetros mediante la técnica de *Grid search*, la cual se encarga de probar todas las combinaciones posibles dado un conjunto de valores de los hiperparámetros y seleccionar los que reflejan un mejor resultado, en este caso los mejores hiperparámetros fueron:

- *Factors* = 20
- *Iterations* = 35
- *Regularization* = 1

Está evaluación se hizo con el método “*mean\_average\_precision\_at\_k*” provisto por la librería “*implicit*”. En el cual se aplicó la técnica de validación cruzada para reservar el 20% del conjunto de dos para pruebas y el 80% para entrenamiento, teniendo en cuenta que se está validando un modelo para  $K = 10$  recomendaciones, se obtuvo como resultado un  $MAP@10 = 0.12196745108693767$ . Lo cual quiere decir que



aproximadamente el 12% de los primeros  $K(10)$  productos recomendados son relevantes para el usuario, es importante recordar que solo se están teniendo en cuenta valoraciones implícitas y que a futuro se pueden tener en cuenta otras valoraciones para mejorar el modelo y además de ello se realizó un filtro de los usuarios que no tuvieran muchas interacciones con los productos con un valor de mínimo 125 iteraciones teniendo en cuenta que puede traer pérdida de datos valiosos, reducir la cobertura de ciertos usuarios y cierto sesgo, se llegó a ese valor teniendo en cuenta que nos permitió obtener un modelo más preciso con datos más consistentes y reduciendo el uso de recursos y carga computacional.

## 5. Conclusión y trabajos futuros

### 5.1. Conclusión

- El módulo de recomendación propuesto utiliza la técnica de filtrado colaborativo basado en retroalimentación implícita con el método de mínimos cuadrados alternos, permitió el desarrollo para poder ofrecer recomendaciones de los productos disponibles teniendo en cuenta que la plataforma no cuenta en su sitio web con valoraciones explícitas para evaluar la interacción real de los productos con los clientes.
- El complemento módulo de recomendación propuesto se basó en la selección del vecino más cercano, y la métrica de similitud del coseno. El cual se apoyó en las características de los usuarios que se almacenan al momento de acceder a la plataforma “iBuyFlowers” y realizar las compras, lo cual facilita hacer una comparación de un usuario objetivo vs un usuario que ya ha realizado compras, y poder ofrecer recomendaciones de los productos disponibles con el modelo propuesto.
- Uno de los puntos importantes abordar es la precisión del modelo, el cual fue considerado para el negocio aceptable siendo una primera versión y teniendo en cuenta que todos los datos son implícitos y que el comportamiento de los clientes en el negocio es impredecible, en el cual teniendo en cuenta el *Mean average precision 10K*, se obtuvo que el 12% de los primeros K(10) productos recomendados son relevantes para el usuario, lo cual deja mucho por mejorar en el modelo para hacer mejor la valoración.
- Uno de los puntos importantes es definir las tablas y datos involucrados en el proceso de compra de un usuario, para así obtener el *dataset* que permita el diseño del módulo de recomendación para la plataforma “iBuyFlowers”, para ello es necesario realizar reuniones con el grupo de trabajo responsable del manejo de esta información dentro de la organización.
- El conjunto de datos utilizado para el desarrollo del módulo de recomendación es muy completo, la información faltante era muy poca y permite realizar el preprocesamiento de los datos para obtener los datos de interés para la fase del desarrollo.
- En el módulo de recomendación propuesto se calcula matriz de similitud entre usuarios semejantes a partir de sus características, y se ordenaron los clientes de mayor a menor según el valor de semejanza obtenido, para a partir de allí realizar las sugerencias de los productos por parte del sistema al usuario final para que sean tomados en cuenta al momento de realizar la compra.



- El uso de herramientas como Python y Jupyter Notebook fueron clave para el desarrollo del módulo, debido al fácil acceso a librerías para manejo de datos y uso para ciencia de datos por parte de Python, así mismo poder combinar código con *markdown* por parte de Jupyter.
- El uso de recursos computacionales para procesar conjuntos de datos grandes fue un factor clave para el desarrollo del proyecto, ya que se utilizó un conjunto inicial con más de 155.000 filas y 157 columnas, que se hacía inmanejable con las plataformas en línea de uso gratuito.

## 5.2. Trabajos futuros

- Analizar la base de datos con el histórico de compras que ha tenido la empresa cuándo comenzó a realizar ventas hasta la actualidad, para brindar recomendaciones basadas en información almacenada de todos los usuarios que han utilizado la plataforma “*iBuyFlowers*”.
- Se necesita la implementación de métricas que permitan agregar más significancia a los valores implícitos y así obtener mejores resultados y mejorar la precisión del módulo propuesto. Utilizar diferentes métricas de similitud que se puedan adaptar a los datos almacenados de la plataforma “*iBuyFlowers*”, ya que este trabajo se enfocó en la retroalimentación implícita limitando las recomendaciones a inferencias en el comportamiento de los usuarios en el pasado.
- Analizar otros escenarios de recomendación, por ejemplo: el enfoque basado en contenido, para evaluar el comportamiento directo que tienen los usuarios con respecto a los productos.
- Sugerir a “*iBuyFlowers*” implementar interacciones con usuarios sobre los productos que adquieren para así obtener información y realizar evaluaciones, tener comentarios, calificaciones, además se pueden hacer encuestas sencillas, que permitan usar esa información para otros tipos de análisis diferentes a una recomendación, como, por ejemplo, medir la satisfacción de un producto en la plataforma y conocer si se debe retirarse o de lo contrario es una estrella y no ha sido promocionado lo suficiente para obtener más ventas.
- Mejorar el modelo que permita incrementar sus métricas de evaluación y darle más confianza a los clientes cuando compran en la plataforma.
- Realizar la integración del módulo de recomendación desarrollado a la plataforma de “*iBuyFlowers*” para recomendar a los clientes según sus intereses o preferencias y conocer el rendimiento con casos reales.

## Apéndice: Código del Proyecto

El código del proyecto desarrollado se encuentra alojado en GitHub, accesible a través del siguiente enlace: <https://github.com/mzuleta6/TFM>.

En este repositorio se podrá detallar el código desarrollado para el sistema de recomendación en un archivo “RecommendationSystem.ipynb” (Archivo de jupyter notebook) que contiene todo el contenido del Notebook con la descripción paso a paso del desarrollo realizado.

El código se organizó de la siguiente manera:

1. Se realiza la importación de las librerías a utilizar, con el comando import (ver ilustración 27), estas son:
  - a. numpy (NumPy)
  - b. seaborn (Seaborn)
  - c. pandas (Pandas)
  - d. cosine\_similarity: con el comando *from* para especificar que viene de (*Scikit-learn*)
  - e. matplotlib.pyplot (Matplotlib).

```
import numpy as np
import seaborn as sns
import pandas as pd
from sklearn.metrics.pairwise import cosine_similarity
import matplotlib.pyplot as plt
```

*Ilustración 27. Importación de las librerías a utilizar. Fuente: Elaboración propia.*

2. Luego, se procede a la selección e importación de los datos mediante el uso de la función “read\_csv” (ver ilustración 28), la cual permite crear un *dataframe* basado en un archivo, en este caso “all\_2023\_ibf\_data.csv” y se imprime la forma de este para obtener la información respectiva al número de filas y columnas de la tabla de datos.

```
df = pd.read_csv('all_2023_ibf_data.csv', low_memory=False)
df.shape
```

*Ilustración 28. Código para la importación de los datos. Fuente: Elaboración propia.*

- Posteriormente, se inicia la etapa del preprocesamiento de los datos, se comienza por la limpieza y preparación de estos para el análisis, se crea un listado llamado “*columns\_selected*” (ver ilustración 29), el cual incluye los nombres de las columnas que serán preseleccionadas, esto permite crear un nuevo *dataset* llamado “*df\_selected*” que es un nuevo conjunto únicamente con las columnas escogidas.

```
# Columnas a trabajar en nuestro dataset
columns_selected = ['cart_box_item_cart_box_purchase_state_state', 'cart_box_item_variety_variety_name', 'cart_box_item_product_group_common_name', 'cart_box_item_cart_box_customer_margin', 'cart_box_item_cart_box_customer_tier_sbx', 'cart_box_item_cart_box_customer_KEY', 'cart_box_item_cart_box_customer_business', 'cart_box_item_cart_box_customer_events_per_year', 'cart_box_item_cart_box_customer_stores_quantity', 'cart_box_item_cart_box_customer_employees_quantity', 'cart_box_item_cart_box_customer_spend_per_week']
df_selected = df[columns_selected]
df_selected.head()
```

*Ilustración 29. Previsualización del código para la selección de columnas. Fuente: Elaboración propia.*

- Para la transformación de los datos se usa de la función “*copy*” del *dataframe* para crear una copia del conjunto de datos “*all\_2023\_ibf\_data.csv*” (ver ilustración 30), esto se hace con el fin de alterar libremente y aplicar las modificaciones a la tabla, como, por ejemplo, el renombrar las columnas a través de la función “*rename*”.

```
copy_selected = df_selected.copy()
copy_selected.rename(columns={
    'cart_box_item_cart_box_purchase_state_state': 'state',
    'cart_box_item_cart_box_customer_margin': 'margin',
    'cart_box_item_cart_box_customer_tier_sbx': 'tier',
    'cart_box_item_cart_box_customer_KEY': 'customer_key',
    'cart_box_item_variety_variety_name': 'variety',
    'cart_box_item_variety_KEY': 'variety_key',
    'cart_box_item_length': 'length',
    'cart_box_item_product_group_common_name': 'product_group',
    'cart_box_item_cart_box_customer_business': 'business',
    'cart_box_item_cart_box_customer_events_per_year': 'events_per_year',
    'cart_box_item_cart_box_customer_stores_quantity': 'stores_quantity',
    'cart_box_item_cart_box_customer_employees_quantity': 'employees_quantity',
    'cart_box_item_cart_box_customer_spend_per_week': 'spend_per_week'
}, inplace=True)
```

*Ilustración 30. Código para realizar una copia del conjunto de datos anterior y renombrar las columnas. Fuente: Elaboración propia.*

- Se efectúa la verificación de valores faltantes mediante un recorrido por las columnas utilizando el ciclo “*for*” y se verifica cada columna a través del método “*isnull*”, se muestra así el porcentaje de filas vacías por cada columna (ver ilustración 31).

```
for col in copy_selected.columns:
    pct_missing = np.mean(copy_selected[col].isnull())
    print('{} - {}'.format(col, round(pct_missing*100)))
```

Ilustración 31. Código para verificar el porcentaje de valores null en las filas por cada columna. Fuente: Elaboración propia.

6. Después, se hace un tratamiento a los datos mediante la técnica de imputación, se asignan valores a esas filas basados en reglas del negocio, se coloca como 'Unknown' todo valor faltante de las columnas de tipo categórico y 0.0 para todo valor faltante de las columnas de tipo numérico (ver ilustración 32).

```
copy_selected.loc[copy_selected['spend_per_week'] == "", ['spend_per_week']] = 'Unknown'
copy_selected.loc[copy_selected['spend_per_week'] != copy_selected['spend_per_week'], ['spend_per_week']] = 'Unkr
copy_selected.loc[copy_selected['events_per_year'] == "", ['events_per_year']] = 'Unknown'
copy_selected.loc[copy_selected['stores_quantity'] == "", ['stores_quantity']] = 0.0
copy_selected.loc[copy_selected['employees_quantity'] == "", ['employees_quantity']] = 0.0
copy_selected.loc[:, ['stores_quantity']] = copy_selected['stores_quantity'].fillna(0).astype(float)
```

Ilustración 32. Código para realizar la asignación de valores por defectos en las filas cuyas columnas tienen datos faltantes. Fuente: Elaboración propia.

7. Ahora pasamos a la fase de creación del modelo, primeramente, se va a crear un nuevo dataframe llamado "products\_by\_freq" en el cual se van agrupar los productos por la cantidad de veces que un cliente lo ha comprado a través de la función "group by" (ver ilustración 33)

```
products_by_freq = copy_selected.copy()

products_by_freq['length'].fillna(0, inplace=True)
products_by_freq['product_name'] = products_by_freq.apply(lambda x: str(x.product_group) + " " + str(x.variety) + "
products_by_freq = products_by_freq.drop(['variety_key', 'variety', 'length',
                                           'product_group', 'margin', 'tier',
                                           'state', 'business',
                                           'events_per_year',
                                           'stores_quantity',
                                           'employees_quantity', 'spend_per_week'], axis='columns')

columns = ['product_name']
aggs = {key: ['count'] for key in columns}
products_by_freq = products_by_freq.groupby(by=["customer_key", 'product_name']).agg(aggs).droplevel(0, axis=1).reset
products_by_freq.head(5)
```

Ilustración 33. Código para realizar el agrupamiento de los usuarios basados en la frecuencia de compra por producto. Fuente: Elaboración propia.

8. Luego de la agrupación procedemos a crear dos columnas nuevas en nuestro conjunto de datos que tendrán como valores IDs numéricos que servirán para la creación de la matriz de dispersión que utilizara el modelo (ver ilustración 34).

```
unique_customers = products_by_freq['customer_key'].unique()
unique_items = products_by_freq['product_name'].unique()

customer_ids = {customer: idx for idx, customer in enumerate(unique_customers)}
item_ids = {item: idx for idx, item in enumerate(unique_items)}

products_by_freq['customer_id'] = products_by_freq['customer_key'].map(customer_ids)
products_by_freq['item_id'] = products_by_freq['product_name'].map(item_ids)

n_users = products_by_freq.customer_key.unique().shape[0]
n_items = products_by_freq.product_name.unique().shape[0]

print('Number of users: {}'.format(n_users))
print('Number of products: {}'.format(n_items))
products_by_freq.head()
```

Ilustración 34. Código para realizar la asignación de columnas con id numéricas para los productos y usuarios. Fuente: Elaboración propia.

9. Para la implementación del modelo se tuvo en cuenta el algoritmo de mínimos cuadrados alternos, el cual utilizaremos a través de la librería “*implicit*” con la función “*AlternatingLeastSquares*” (ver ilustración 35).

```
# initialize a model
model = implicit.als.AlternatingLeastSquares(factors=50, regularization=1, random_state=123)
model.fit(coo_train)
```

Ilustración 35. Código para realizar la inicialización del modelo. Fuente: Elaboración propia.

10. Luego para la creación y validación del modelo se procede a realizar la técnica de *Grid search*, en la cual se establecieron unos valores predefinidos para los hiperparámetros para conseguir el mejor modelo posible, en el cual por medio de la función “*train\_test\_split*” de *sklearn* se procedió a la evaluación del modelo a partir de una matriz dispersa con el 80% de los datos y el 20% reservado para la validación del modelo (ver ilustración 36).



```
def sparse_customer_item(df):
    row = df['customer_id'].values
    col = df['item_id'].values
    data = np.ones(df.shape[0])
    coo = coo_matrix((data, (row, col)), shape=(len(unique_customers.tolist()), len(unique_items.tolist())))
    return coo

def split_data(df):
    df_train, df_val = train_test_split(df, test_size=0.2)
    return df_train, df_val

def get_sparse_matrix(df):
    df_train, df_val = split_data(df)
    coo_train = sparse_customer_item(df_train)
    coo_val = sparse_customer_item(df_val)

    csr_train = coo_train.tocsr()
    csr_val = coo_val.tocsr()

    return {'coo_train': coo_train,
            'csr_train': csr_train,
            'csr_val': csr_val}

def validate(matrix, factors=200, iterations=20, regularization=0.01, show_progress=True):

    coo_train, csr_train, csr_val = matrix['coo_train'], matrix['csr_train'], matrix['csr_val']

    model = implicit.als.AlternatingLeastSquares(factors=factors,
                                                  iterations=iterations,
                                                  regularization=regularization,
                                                  random_state=42)

    model.fit(coo_train, show_progress=show_progress)

    map10 = mean_average_precision_at_k(model, csr_train, csr_val, K=10, show_progress=show_progress)
    # print(f"Factors: {factors:>3} - Iterations: {iterations:>2} - Regularization: {regularization:4.3f} ==> \nMAP@
    # print('-----\n\n\n')
    return map10

matrix = get_sparse_matrix(products_by_freq)

best_map10 = 0

for factors in [20, 50, 70, 100]:
    for iterations in [12, 15, 20, 35]:
        for regularization in [0.01, 0.05, 0.1, 1]:
            map10 = validate(matrix, factors, iterations, regularization, show_progress=False)
            if map10 > best_map10:
                best_map10 = map10
                best_params = {'factors': factors, 'iterations': iterations, 'regularization': regularization}
            # print(f"Best MAP@10. Updating: {best_params}")
```

Ilustración 36. Código para realizar la validación del modelo y obtención de los mejores hiper parámetros. Fuente: Elaboración propia.

11. Luego se procede a la creación y entrenamiento del modelo, dados los hiperparámetros previamente seleccionados (ver ilustración 37)

```
coo_train = sparse_customer_item(products_by_freq)
csr_train = coo_train.tocsr()

def train(coo_train, factors=100, iterations=15, regularization=0.01, show_progress=True):
    model = implicit.als.AlternatingLeastSquares(factors=factors,
                                                  iterations=iterations,
                                                  regularization=regularization,
                                                  random_state=42)

    model.fit(coo_train, show_progress=show_progress)
    return model

model = train(coo_train, **best_params)
```

Ilustración 37. Código para el entrenamiento de los datos. Fuente: Elaboración propia.

12. Finalmente, se crea una función que permita dado un *customer* obtener sus recomendaciones, lo cual se hará a través de la función “*recommend*” del modelo previamente entrenado obteniendo el siguiente resultado (ver ilustración 38).

```
items_by_id = dict(list(enumerate(unique_items.tolist())))

def getProductRecommendations(customer):
    userId=customer_ids[customer]
    ids, scores = model.recommend(userId, csr_train[userId], N=10)
    product_id = [items_by_id[x] for x in ids]
    result_table = pd.DataFrame({"product_id": product_id, "score": scores})
    return result_table

basedRecommendations = getProductRecommendations('45193aab-887a-4590-be6e-6558691e63b1')
basedRecommendations
```

Ilustración 38. Código para la generación de recomendaciones de los datos. Fuente: Elaboración propia.

13. Luego de realizar la generación de las recomendaciones, en la ilustración 39 se puede observar una función de validación de parámetros en la cual tiene en cuenta el *Mean Average Precision at k (MAP@k)*, el cual tomaremos para evaluar el modelo y reflejar la capacidad de generar elementos relevantes arrojando como resultado que solo el 12% de los datos son relevantes para un usuario (ver ilustración 39).

```
print(best_map10)
perc = round(best_map10 * 100)
print(f' Precision: {perc}%')

0.12196745108693767
Precision: 12%
```

Ilustración 39. Resultado al aplicar el *Mean Average Precision*. Fuente: Elaboración propia.

14. Luego de finalizar la limpieza de los datos y obtención del modelo para recomendar productos, se toma el dataset, para encontrar el vecino más cercano al usuario objetivo y realizar las recomendaciones, para ello se utiliza las funciones “*drop*” para eliminar columnas del *dataset* y dejar únicamente la información relacionada al usuario, además, se realiza un agrupamiento de los usuarios teniendo en cuenta el “*customer\_key*” como identificador único, el cual se asigna como índice del conjunto de datos a través de la función “*set\_index*”, se imprime la información relacionada al conjunto de datos para conocer la cantidad de información que ahora te tiene, esto se realiza mediante la función “*shape*” (ver ilustración 40).

```
copy_selected_group = copy_selected.drop(['product_group', 'variety', 'variety_key', 'length'], axis='columns')
columns = ['state', 'tier', 'business', 'events_per_year', 'spend_per_week', 'stores_quantity',
           'employees_quantity', 'margin']
aggs = {key: ['first'] for key in columns}
copy_selected_group = copy_selected_group.groupby(by=["customer_key"]).agg(aggs).droplevel(0,axis=1)
.reset_index().set_index('customer_key')
copy_selected_group.columns = columns

print(copy_selected_group.shape)
copy_selected_group.head()
```

(2238, 8)

*Ilustración 40. Código para realizar el agrupamiento de los usuarios basados en sus características.  
Fuente: Elaboración propia.*

15. En el análisis, se realiza la inserción de un nuevo usuario (ver ilustración 41), para ello se crea un diccionario que es adjuntado a un *dataset* nuevo por medio de la función de pandas “*dataframe*”, que recibe como parámetros, el diccionario que incluye la información y el *index* que es el identificador del usuario. Igualmente, se procede hacer el tratamiento en caso de que presente algún valor faltante y se realiza la concatenación de los dos conjuntos de datos por medio de la función “*concat*” que está en la librería de pandas:

```
customer_key = "c7792ee4-7303-4919-9521-1f221d5bdd96"

new_customer = {
    "state": "Texas",
    "tier": "Tier_2",
    "business": "Wedding & event floral designer only (not a florist)",
    "events_per_year": "25 - 59",
    "spend_per_week": "Unknown",
    "stores_quantity": "1",
    "employees_quantity": "8",
    "margin": 31
}

new_df = pd.DataFrame(new_customer, index=[customer_key])

copy_new_df = new_df.copy()

copy_new_df.loc[copy_new_df['spend_per_week'] == "", ['spend_per_week']] = 'Unknown'
copy_new_df.loc[copy_new_df['spend_per_week'] != copy_new_df['spend_per_week'], ['spend_per_week']] = 'Unknown'
copy_new_df.loc[copy_new_df['events_per_year'] == "", ['events_per_year']] = 'Unknown'
copy_new_df.loc[copy_new_df['stores_quantity'] == "", ['stores_quantity']] = 0.0
copy_new_df.loc[copy_new_df['employees_quantity'] == "", ['employees_quantity']] = 0.0
copy_new_df.loc[:, ['stores_quantity']] = copy_new_df['stores_quantity'].fillna(0).astype(float)
copy_new_df.loc[:, ['employees_quantity']] = copy_new_df['employees_quantity'].fillna(0).astype(float)

copy_selected_group = pd.concat([copy_selected_group, copy_new_df])
copy_selected_group
```

*Ilustración 41. Código la creación de un nuevo dataset que contiene al nuevo usuario registrado, tratamiento de datos faltantes y concatenación con el conjunto de datos existentes. Fuente: Elaboración propia.*

16. A continuación, se convierten las columnas con valores categóricos en valores numéricos con el uso de la función “get\_dummies” de la librería pandas (ver ilustración 41). Esta función provee un nuevo dataset donde cada columna representa una categoría y un valor, donde 1 es equivalente a la presencia esa categoría y un 0 su ausencia.

```
data = pd.get_dummies(copy_selected_group, columns=['state', 'tier', 'business', 'events_per_year', 'spend_per_week'])
```

*Ilustración 42. Previsualización del código para convertir valores categóricos en numéricos. Fuente: Elaboración propia.*

17. Luego de obtener el conjunto de datos únicamente con valores numéricos, se aplica la similitud de coseno a través de la función “cosine\_similarity” de Scikit-learn (ver ilustración 42), ésta calcula la semejanza entre dos conjuntos de vectores y se muestra una previsualización de los 100 primeros datos para ver el resultado de la aplicación del proceso.

```
cos_sim = pd.DataFrame(cosine_similarity(data), columns=data.index, index=data.index)
cos_sim.head(100)
```

*Ilustración 43. Código para la aplicación de la similitud del coseno. Fuente: Elaboración propia.*

18. A continuación, se obtienen los resultados de la aplicación de la métrica seleccionada, para ello se utilizó la función “iloc” de pandas aplicado al *dataframe* (ver ilustración 43), eso permite conseguir los clientes basados en el índice filtrados por la siguiente condición:

- Obtener de todas las filas cuyo índice sea diferente al usuario ingresado en el paso 15, el cual es el usuario objetivo.

Así mismo, se ordena los valores de forma descendente para obtener los que tengan mayor puntuación de similitud, se toman los 5 primeros para visualizarlos y se guarda el índice del que tiene más semejanza el cual se usa para obtener los productos que van a ser recomendados.

```
print(f"Customer key: {customer_key}")
sim_customers = cos_sim[customer_key].iloc[lambda x: x.index != customer_key].sort_values(ascending=False).head(5)
df_sim_customers = pd.DataFrame(sim_customers)

sim_customer = sim_customers.index[0]

print(f"\nMore similar customer: {sim_customer}")

df_sim_customers.head()
```

*Ilustración 44. Código para obtener los usuarios más semejantes ordenados por puntuación y guardar el más cercano. Fuente: Elaboración propia.*

19. Cuando se tiene el cliente con mayor similitud con el usuario objetivo, se crea una previsualización del conjunto de datos de estos por medio de la función de pandas “dataframe” (ver ilustración 44).

```
: pd.DataFrame([copy_selected_group.loc[sim_customer], copy_selected_group.loc[customer_key]])
```

*Ilustración 45. Código creación de un dataframe comparativo. Fuente: Elaboración propia.*

20. Finalmente se obtienen las recomendaciones para el nuevo usuario (ver ilustración 45).

```
newUserRecommendations = getProductRecommendations(sim_customer)
newUserRecommendations
```

*Ilustración 46. Código para la generación de recomendaciones para un nuevo usuario. Fuente: Elaboración propia.*

21. Para previsualizar las recomendaciones se crea una función, dadas las recomendaciones y el dataset inicial, se extraen las imágenes y posteriormente se visualizan en un dataframe (ver ilustración 46)

```
def getRecommenderListPreview(recommendations):
    copy_rec = recommendations.copy()
    copy_rec.rename(columns={'product_id': 'product_name'}, inplace=True)

    df_merge = pd.merge(recommendations, copy_products[['product_name', 'variety_key']], on='product_name', how='left')
    df_merge = df_merge.drop_duplicates()
    df_merge['image'] = df_merge.apply(lambda x: "https://sbxcloud.com/www/ibuyflowers/varieties/variety_" + x.variety_key + ".jpg", axis=1)
    df_merge['image_preview'] = df_merge['image'].apply(lambda f: path_to_image_html(f))
    return df_merge[['product_name', 'image_preview']]

# Example
getRecommenderListPreview(basedRecommendations)

newUserRecommendations = getProductRecommendations(sim_customer)
newUserRecommendations
```

*Ilustración 47. Código previsualizar las imágenes de los productos recomendados. Fuente: Elaboración propia.*

## Referencias

- Baxla, M. A. (2014). *Comparative study of similarity measures for item based top n recommendation*. <https://core.ac.uk/reader/53190130>
- Bellogín, A., Castells, P., & Cantador, I. (2014). Neighbor selection and weighting in user-based collaborative filtering: A performance prediction approach. *ACM Transactions on the Web*, 8(2), 1–30. <https://doi.org/10.1145/2579993>
- Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., & Wirth, R. (2000). *University of Kassel*. Retrieved from University of Kassel: <https://www.kde.cs.uni-kassel.de/wp-content/uploads/lehre/ws2012-13/kdd/files/CRISPWP-0800.pdf>
- Bellogín, A., Cantador, I., & Castells, P. (2014). Neighbor Selection and Weighting in User-Based Collaborative Filtering: A Performance Prediction Approach. *Research Gate*, 31.
- Bach Munkholm, N., A. Alphinias, R., & Tambo, T. (2024). Collaborative filtering, K-nearest neighbor and cosine similarity in Home Décor recommender systems – a case study . *arXiv*, 8.
- Baxla, M. A. (2014). Comparative study of similarity measures for item based top n recommendation. *CONnecting REpositories*, 25.
- Fan, B., & Hu, J. (2023). Application Research of Collaborative Filtering Algorithm in Catering Recommendation System . *IEEE*.
- Frederickson, B. (n.d.). *benfred.github.io*. Retrieved from benfred.github.io: <https://benfred.github.io/implicit/index.html>
- iBuyFlowers. (2017). *iBuyFlowers*. Retrieved from iBuyFlowers: <https://www.ibuyflowers.com/>
- Han, J., Kamber, M., & Pei, J. (2012). In *Data Mining: Concepts and Techniques* (p. 703). Morgan Kaufmann.
- Hu, Y. (2015). *yifanhu*. Retrieved from yifanhu: <http://yifanhu.net/index.html>
- Khatteer, H., Goel, N., Gupta, N., & Gulati, M. (2021). Movie Recommendation System using Cosine Similarity with Sentiment Analysis. *IEEE*, 7.
- Kumar Singh, M., Prakash Rishi, O., Kumar Singh, A., Singh, P., & Pushpa, C. (2021). Implementation of Knowledge based Collaborative Filtering and Machine Learning for E-Commerce Recommendation System. *IOP Science*.
- Li, L. (2024). Research on Personalized Recommendation System for E-Commerce Products Based on Collaborative Filtering Algorithm. *IEEE*.

- Li, X. (2021). Research on the Application of Collaborative Filtering Algorithm in Mobile E-Commerce Recommendation System. *IEEE*.
- Loukili, M., Messaoudi, F., & El Ghazi, M. (2023). Personalizing Product Recommendations using Collaborative Filtering in Online Retail: A Machine Learning Approach. *IEEE*, 6.
- Moreno Fernandez, S. (2020). *Herramienta de Reconocimiento de Imágenes en Python*. Retrieved from Universidad de sevilla: <https://biblus.us.es/bibing/proyectos/abreproy/92877/fichero/TFG-2877+MORENO+FERN%C3%81NDEZ%2C+SAMUEL.pdf>
- Ricci, F., Rokach, L., & Shapira, B. (2011). *Recommender Systems Handbook*. Springer.
- SBXCloud. (n.d.). *SBXCloud*. Retrieved from SBXcloud: <https://sbxcloud.com/#!/terms>
- SciPy. (n.d.). *SciPy*. Retrieved from <https://scipy.org/>
- Vilca Paredes, J. S. (2020). Análisis de riesgo para préstamos bancarios. *Revistas Bolivianas*.
- Vullam, N., Vellela, S., Reddy B , V., Rao, M., Basha SK, K., & D, R. (2016). Multi-Agent Personalized Recommendation System in E-Commerce based on User . *IEEE*.
- Yadav, R., Choorasiya, A., Singh, U., Khare, P., & Pahade, P. (2018). A Recommendation System for E-Commerce Base on Client Profile. *IEEE*.
- Yang, Y., & Zhang, H. (2011). An E-Commerce Personalized Recommendation System Based on Customer Feedback . *IEEE*.
- Zan, Z., & Zhang, J. (2010). Item-based collaborative filtering with fuzzy vector cosine and item directional similarity . *IEEE*.