



GRADO EN INGENIERÍA EN TECNOLOGÍAS DE TELECOMUNICACIÓN

TRABAJO FIN DE GRADO

Desarrollo de un sistema recomendador para la
compra de ropa online

Autor: Gonzalo Fuster Olivera

Director: Carlos Miguel Vallez Fernández

Madrid

Declaro, bajo mi responsabilidad, que el Proyecto presentado con el título
Desarrollo de un sistema recomendador para la compra de ropa online
en la ETS de Ingeniería - ICAI de la Universidad Pontificia Comillas en el
curso académico 2023/07 es de mi autoría, original e inédito y
no ha sido presentado con anterioridad a otros efectos.

El Proyecto no es plagio de otro, ni total ni parcialmente y la información que ha sido
tomada de otros documentos está debidamente referenciada.



Fdo.: Gonzalo Fuster Olivera Fecha: 05/07/ 2023

Autorizada la entrega del proyecto

EL DIRECTOR DEL PROYECTO



Fdo.: Carlos Miguel Vallez Fernández Fecha: 05/07/ 2023



COMILLAS

UNIVERSIDAD PONTIFICIA

ICAI

GRADO EN INGENIERÍA EN TECNOLOGÍAS DE TELECOMUNICACIÓN

TRABAJO FIN DE GRADO

Desarrollo de un sistema recomendador para la
compra de ropa online

A handwritten signature in black ink, appearing to read 'Gonzalo Fuster Olivera', with a long, sweeping underline.

Autor: Gonzalo Fuster Olivera

Director: Carlos Miguel Vallez Fernández

Madrid

Agradecimientos

A mis padres, profesores y compañeros: por su guía, motivación y dedicación, sin los cuales no habría alcanzado este logro y terminar esta carrera.

A mi director Carlos Miguel Vallez Fernández por aceptar este proyecto independiente y adaptarse a mis tiempos y trabajos.

DESARROLLO DE UN SISTEMA RECOMENDADOR PARA LA COMPRA DE ROPA ONLINE

Autor: Fuster Olivera, Gonzalo

Director: Vallez Fernández, Carlos Miguel.

Entidad Colaboradora: ICAI – Universidad Pontificia Comillas

RESUMEN DEL PROYECTO

Se ha desarrollado un sistema recomendador para la compra de ropa online adaptado a las necesidades y tendencias actuales. Se ha conseguido diseñar un sistema que recomiende conjuntos de dos prendas de ropa en función del histórico de compras y la composición en color forma de ambas prendas.

Palabras clave: sistema recomendador, matrix factorization, autoencoder, redes neuronales, image embedding, ropa, moda, web scraping.

1. Introducción

La compra de ropa online ha crecido de manera durante los últimos años y se prevé que siga creciendo en los próximos años. El formato de comprar ropa online no ha evolucionado acorde con su desarrollo y crecimiento. La innovación se ha concentrado en la parte visual de la compra online, pero ha habido muy poco desarrollo sobre cómo se navega a través de esta ropa y sobre las recomendaciones que reciben los clientes. Estas recomendaciones se limitan al mismo tipo de recomendaciones y formatos que tenían las páginas de compra online hace 9 años.

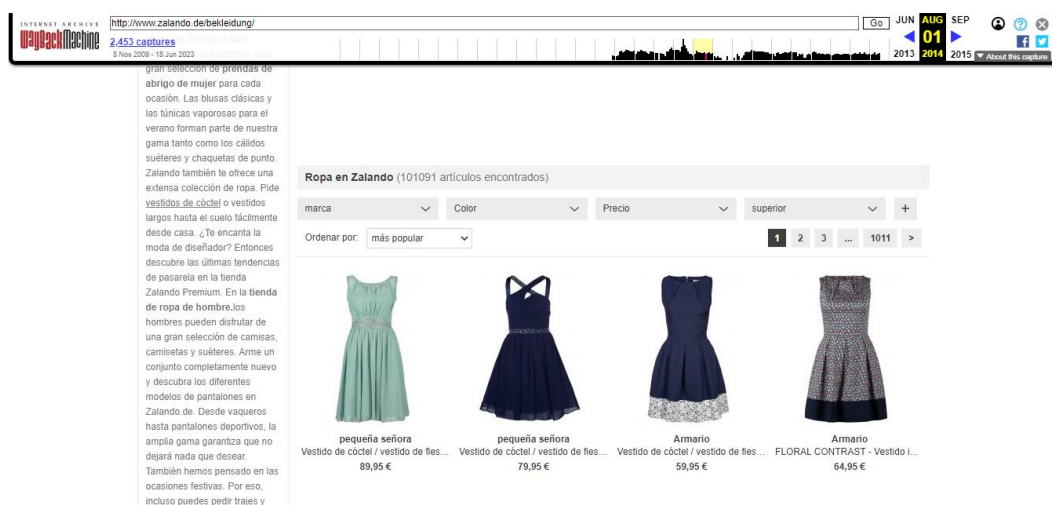


Ilustración 1 – Web de Zalando en 2014

La compra online tiene que evolucionar y tiene que empezar por un sistema recomendador que se ajuste a la realidad de hoy en día (Peña-Fernández, Larrondo-Ureta, & Morales-i-Gras, 2022). Se requiere un sistema capaz de ajustarse a los intereses individuales de los usuarios y utilizar la retroalimentación implícita, es decir, la información generada a partir de las interacciones y el comportamiento de los usuarios dentro de la aplicación. Esta información incluye métricas como el tiempo de

visualización, los clicks o las ventas, para combinarlas y generar una única métrica que refleje el uso que las empresas quieren de los usuarios.

2. Definición del proyecto

El proyecto consiste en desarrollar una base de datos con imágenes e información sobre prendas de ropa para crear un recomendador acorde a las tendencias y tecnologías actuales. Se espera que el recomendador se pueda adaptar a las nuevas tendencias en aplicaciones, TikTok o Tinder. De esta manera, se podrá navegar a través del catálogo de manera más eficiente, viendo prendas que se ajusten a las preferencias de los clientes.

Con el recomendador, se pretende limitar el impacto de una primera mala página en los usuarios, (Çano & Morisio, 2017) haciendo que encuentren prendas que sean de su agrado. El formato de aplicación fomenta el uso de la herramienta, aumentando la capacidad de los minoristas de ropa para generar datos de calidad.

3. Descripción del sistema recomendador

El sistema recomendador consta de 4 partes conectadas entre sí, que terminan con la recomendación de un atuendo con una parte de arriba y otra de abajo. En la figura se pueden apreciar todas las partes involucradas.

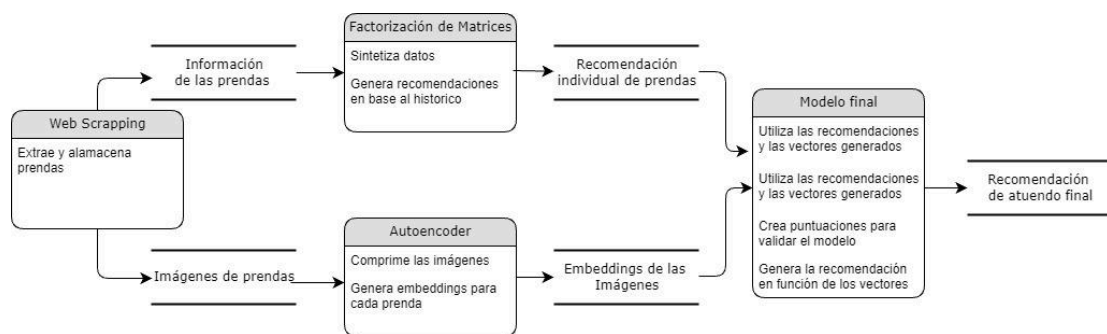


Ilustración 2 – Esquema de los módulos del proyecto

En primera instancia, *scrapeando* la página web de Zalando se extrae una base de prendas sobre la que se desarrolla todo el proyecto. Las imágenes se comprimen utilizando un *autoencoder* generando un *embedding* para cada imagen. La información de las prendas se complementa sintetizando compras para generar recomendaciones de en función a un histórico.

Las recomendaciones de prendas se juntan generando todas las combinaciones posibles, que serán utilizadas en el modelo como entradas. Al no contar con datos empíricos, se generan combinaciones fácilmente reconocibles. Utilizando el *clustering* se categorizan las prendas de arriba, mientras que para las de abajo se utiliza su categoría. Una vez aleatorizado las puntuaciones en función a las combinaciones, los *embeddings* de las imágenes se introducen una red convolucional profunda para generar las mejores combinaciones

4. Resultados

Los atuendos generados por el sistema completo consiguen recoger los dos aspectos fundamentales presentes en los datos introducidos. Por un lado, prendas poco compradas en los datos sintetizados como las blazers, no aparecen en las recomendaciones. Los clientes no compran ese tipo de prendas y, por lo tanto, aunque la combinación tenga éxito en el modelo, no pasan el primer punto.

Por otro lado, el modelo final es capaz de utilizar la información comprimida de las imágenes para identificar patrones de colores y la forma de las prendas. El modelo identifica las prendas de arriba con colores rojizos, rosas y naranjas, combinan con faldas. El modelo también identifica la otra combinación. Esta se compone de azul marino o negro por la parte de arriba con pantalones vaqueros. En este caso al haberla puesto como una combinación peor que la primera pero mejor que el resto de las combinaciones, la identifica como segunda opción. Así se vería una recomendación en la aplicación.



Ilustración 3 – Ejemplo de la aplicación

5. Conclusiones

El sistema recomendador completo es capaz de ofrecer una recomendación adaptada a las tecnologías y necesidad de hoy en día. Fomenta el uso de una posible aplicación dinámica, similar a las que hoy en día se utiliza habitualmente. El sistema ha sido capaz de basar sus recomendaciones en el histórico de compras y de identificar patrones, formas y prendas, para ofrecer un atuendo que le guste al cliente y quiera seguir viendo más.

6. Referencias

- [1] Peña-Fernández, S., Larrondo-Ureta, A., & Morales-i-Gras, J. (2022). *Current affairs on TikTok. Virality and entertainment for digital natives*. Çano, E., &
- [2] Morisio, M. (2017). *Hybrid recommender systems: A systematic literature review*.
- [3] Kiela, D., & Bottou, L. (2014). Learning image embeddings using convolutional neural networks for improved multi-modal semantics. *Proceedings of the 2014 Conference on empirical methods in natural language processing (EMNLP)*, 36-45.

Development of a recommender system to purchase garments online

Autor: Fuster Olivera, Gonzalo

Director: Vallez Fernández, Carlos Miguel.

Entidad Colaboradora: ICAI – Universidad Pontificia Comillas

ABSTRACT

A recommendation system for online clothes shopping has been developed, tailored to address present needs and trends. A system has been successfully designed to provide recommendations for the combination of two garments, taking into account the purchase history as well as the colour and form of both articles.

Keywords: recommender system, matrix factorization, autoencoder, neural networks, image embedding, clothing, fashion, web scraping

1. Introduction

Online shopping for clothes has grown in a big way over the last few years and is expected to continue to grow in the coming years. The format of buying clothes online has not evolved in line with its development and growth. All development has concentrated on the visual part of online shopping, but there has been very little development on how to navigate through the clothes and the recommendations that customers receive. These recommendations are limited to the same type of recommendations and formats that online shopping sites had 9 years ago.

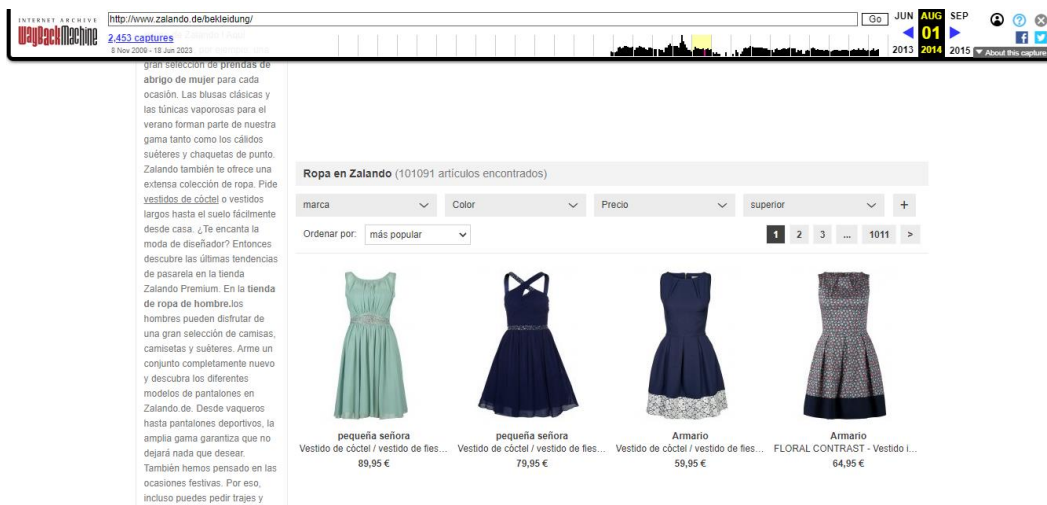


Illustration 1 – Zalando Web in 2014

Online shopping has to evolve, and it has to start with a recommender system that fits today's reality (Peña-Fernández, Larrondo-Ureta, & Morales-i-Gras, 2022). It requires a system capable of adjusting to users' individual interests and using implicit feedback, i.e., information generated from users' interactions and behaviour within the application. This information includes metrics such as viewing time, clicks, or sales, to combine them to generate a single metric that reflects the usage that companies want from users.

2. Project definition

The project consists of developing a database with images and information about clothes to create a recommender according to current trends and technologies. It is hoped that the recommender can be adapted to new trends in apps, TikTok or Tinder. In this way, it will be possible to browse through the catalog in a more efficient way, seeing garments that fit the preferences of the customers.

With the recommender, it is intended to limit the impact of a bad first page on users, (Çano & Morisio, 2017) by making them find clothes that are to their liking. The app format encourages the use of the app, increasing the ability of clothing retailers to generate quality data.

3. Description of the recommender system

The recommender system consists of 4 interlinked parts, ending with the recommendation of an outfit with a top and a bottom part. In the figure, you can see all the parts involved.

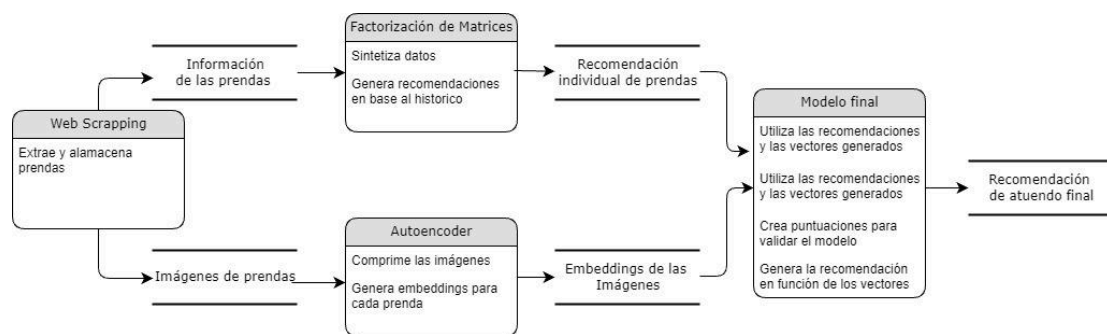


Illustration 2 - Diagram of the functioning of the recommender system

In the first instance, by scraping the Zalando website, a base of garments is extracted on which the whole project is developed. The images are compressed using an autoencoder, generating an embedding for each image. The garment information is complemented by synthesizing purchases to generate recommendations based on history.

The garment recommendations are put together generating all possible combinations, which will be used in the model as inputs. In the absence of empirical data, easily recognizable combinations are generated. Using clustering, the top garments are categorized, while the bottom garments are categorized by their category. Once the scores have been randomized according to the combinations, the embeddings of the images are introduced into a deep convolutional network to generate the best combinations.

4. Results

The outfits generated by the complete system manage to pick up the two fundamental aspects present in the input data. On the one hand, garments that are rarely purchased in the synthesized data, such as blazers, do not appear in the recommendations. Customers

do not buy this type of garment and therefore, even if the combination is successful in the model, it does not pass the first point.

On the other hand, the final model is able to use the compressed information from the images to identify colour patterns and the shape of the garments. The model identifies the top garments with reddish, pink and orange colours, matched with skirts. The model also identifies the other combination. This consists of navy blue or black on the top with jeans. In this case, having put it as a worse combination than the first but better than the rest of the combinations, the model identifies it as the second option. That is how a recommendation would look in the app.



Illustration 3 – Example of the application

5. Conclusions

The complete recommender system is able to offer a recommendation adapted to today's technologies and needs. It encourages the use of a possible dynamic application like those commonly used today. The system has been able to base its recommendations on purchase history and identify patterns, shapes, and garments, to offer an outfit that the customer likes and wants to see more of.

6. Bibliography

[1] Peña-Fernández, S., Larrondo-Ureta, A., & Morales-i-Gras, J. (2022). *Current affairs on TikTok. Virality and entertainment for digital natives*. Çano, E., &

[2] Morisio, M. (2017). *Hybrid recommender systems: A systematic literature review*.

[3] Kiela, D., & Bottou, L. (2014). Learning image embeddings using convolutional neural networks for improved multi-modal semantics. *Proceedings of the 2014 Conference on empirical methods in natural language processing (EMNLP)*, 36-45.

Índice

Índice de figuras	III
Capítulo 1. INTRODUCCIÓN.....	5
1.1 Motivación	7
1.1.1 Objetivos principales.....	7
1.2 Metodología.....	8
1.2.1 Módulos del proyecto	8
1.2.2 Agile.....	10
Capítulo 2. ESTADO DE LA CUESTION.....	12
2.1 Modelos basados en filtros colaborativos.....	14
2.1.1 Recomendaciones basadas en la memoria	15
2.1.2 Recomendaciones basadas en modelos	15
2.2 Modelos basados en contenidos	16
2.3 Modelos Híbridos	18
2.4 Tendencias actuales.....	19
Capítulo 3. desarrollo del sistema recomendador	22
3.1 Extracción y almacenamiento de las imágenes y características de las prendas.	23
3.1.1 Selección de la página web	23
3.1.2 Web Scraping.....	26
3.1.3 Almacenamiento	29
3.2 CODIFICAR LAS IMÁGENES DE LAS PRENDAS A TRAVÉS DE UN AUTOENCODER	32
3.2.1 Fundamentos teóricos	32
3.2.2 Arquitectura del Autoencoder	34
3.2.3 Entrenamiento del modelo.....	37
3.3 DESARROLLAR UN SISTEMA DE RECOMENDACIÓN DE PRENDAS BASADO EN FACTORIZACIÓN DE MATRICES.....	38
3.3.1 Extracción de datos	38
3.3.2 Fundamentos teóricos	39
3.3.3 Generación de recomendaciones.....	41
3.4 DESARROLLO DEL MODELO FINAL	42

3.4.1 Randomización de los ratings	42
3.4.2 Creación del modelo.....	44
3.4.3 Entrenamiento del modelo.....	46
Capítulo 4. Análisis de Resultados.....	48
4.1 Web Scraping	48
4.2 Autoencoder	49
4.3 Factorización de matrices.....	51
4.4 Modelo final	52
Capítulo 5. Conclusiones.....	56
5.1 futuros posibles desarrollos del proyecto	57
Capítulo 6. Bibliografía.....	59
ANEXO I: ALINEACIÓN DEL PROYECTO CON LOS ODS	61

Índice de figuras

Ilustración 1. Web de Zalando en 2014	5
Ilustración 2. Matrix Usuario-Ítem	13
Ilustración 3. Agrupación sistemas recomendadores	13
Ilustración 4. Fundamentos filtros colaborativos	14
Ilustración 5. Modelos basados en contenido	17
Ilustración 6. Esquema de los módulos del proyecto	22
Ilustración 7. Web de Zara actualmente	25
Ilustración 8. Prenda en la web de Zalando	28
Ilustración 9. Búsqueda de prendas en Zalando	29
Ilustración 10. Arquitectura del Encoder	35
Ilustración 11. Imágenes pertenecientes al cluster 1	43
Ilustración 12. Imágenes pertenecientes al cluster 2	43
Ilustración 13. Arquitectura modelo final (1)	44
Ilustración 14. Arquitectura modelo final (2)	45
Ilustración 15. Ejemplo de la aplicación	46
Ilustración 16. Distribución de las prendas por categoría	48
Ilustración 17. Error del Autoencoder durante el entrenamiento	49
Ilustración 18. Reconstrucción prendas inferiores	50
Ilustración 19. Reconstrucción prendas superiores	51
Ilustración 20. Explicación de la varianza en función del número de componentes	51
Ilustración 21. Distribución de las puntuaciones generadas	52
Ilustración 22. Atuendos recomendados para un cliente	53
Ilustración 23. Atuendos generados para un cliente	54
Ilustración 24. Objetivos de Desarrollo Sostenible	61

Capítulo 1. INTRODUCCIÓN

La compra de ropa online ha crecido de manera durante los últimos años y se prevé que siga creciendo en los próximos años. Sin embargo, la forma de comprar esta ropa online apenas ha evolucionado en los años recientes. La mayoría de los cambios en la compra online se han centrado en cambios de la Interfaz de Usuario o (UI) o de la Experiencia de Usuario (UX). Se han centrado principalmente en el desarrollo de aplicaciones móviles y webs responsive que se adapten a dispositivos móviles, otras formas de presentar la ropa ... Incluso algunas empresas como Zalando, no ha cambiado su diseño de su página web desde hace al menos 10 años.

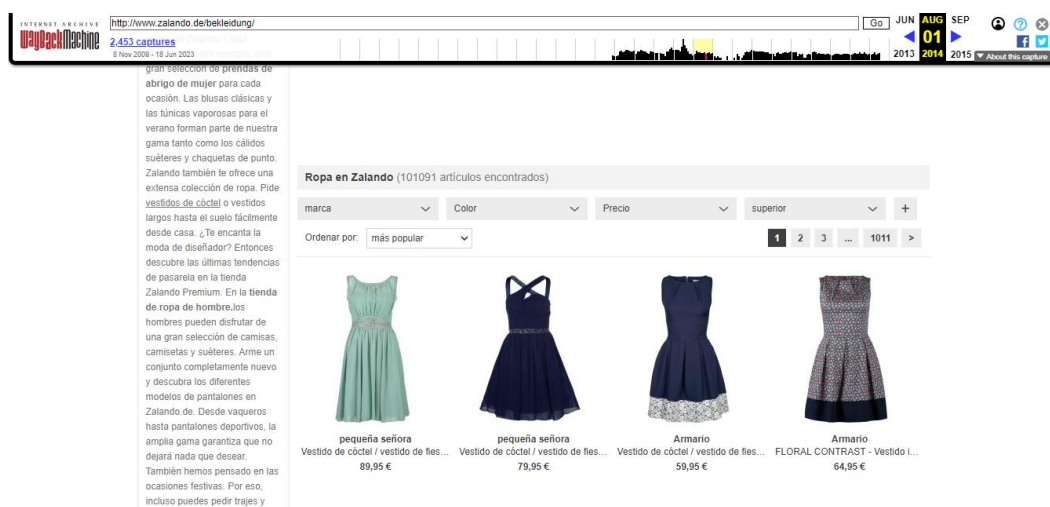


Ilustración 1. Web de Zalando en 2014

Todo el desarrollo disponible en la actualidad se ha concentrado en la parte visual de la compra online, pero ha habido muy poca innovación sobre como navegamos a través de esta ropa. Todos los sitios de compra online nos hacen una categorización en función del tipo prenda, la temporada, las ofertas, nuevas prendas ... Las recomendaciones de compra se basan en las prendas más vendidas, pero no en los gustos de los clientes. Las tiendas online

tienen decenas de miles de prendas, separadas en decenas de páginas. Las personas con un uso habitual no llegan más allá de la 2 o tercera página, es decir, ven un pequeño porcentaje de las prendas que una tienda puede ofrecer. Las personas se limitan a ver aquello que más se ha comprado en la tienda. Pero ¿y si el gusto de alguien no coincide con el de la mayoría? ¿Y si la prenda está en la página 15, a la cual nunca se va a llegar?

Se requiere un sistema de recomendación capaz de ajustarse a los intereses individuales de los usuarios y utilizar la retroalimentación implícita, es decir, la información generada a partir de las interacciones y el comportamiento de los usuarios dentro de la aplicación (Jannach, Lerche, & Zanker, *Recommending Based on Implicit Feedback*, 2018). Esta información incluye métricas como el tiempo de visualización, los *clicks* o las ventas, para combinarlas y generar una única métrica a optimizar. Esta métrica resume como de bien retienen las recomendaciones al posible comprador o a que compre en la aplicación.

Por otro lado, se espera que el recomendador se pueda adaptar a las nuevas interfaces de usuario más extendidas hoy en día, como pueden ser la TikTok o Tinder (Peña-Fernández, Larrondo-Ureta, & Morales-i-Gras, 2022). De esta manera se podrá navegar a través del catálogo online de una manera mucho más eficiente, viendo prendas que se ajusten más a los gustos y preferencias de los clientes. Las tiendas consiguen que su amplia gama de productos sea vista en su totalidad y los clientes consiguen encontrar prendas que se ajusten más a sus gustos. Considerando un uso casual de los usuarios, las únicas prendas que se la mayoría de los compradores ven son las más vendidas o aquellas promocionadas por la página. Hay muchos usuarios que, si no han visto nada que les guste en la primera página, abandonan la tienda en búsqueda de una mejor “primera plana”. Con el recomendador, se pretende limitar el impacto de una primera mala impresión en los usuarios, haciendo que encuentren prendas que sean de su agrado.

1.1 MOTIVACIÓN

El proyecto nace con el fin de adaptar los sistemas recomendadores existentes en la actualidad a un caso de uso en auge, la compra de ropa online. La nueva manera de acceder al contenido permite que se creen algoritmos de personalización mucho más potentes, ya que la retroalimentación es constante y dinámica. Los algoritmos de recomendación recibirán información sobre todas las prendas y de una forma que se ha comprobado que funciona con TikTok o Tinder, haciendolos mejores y más personalizados (Menon, 2022).

Esto desemboca en una mejor experiencia de compra, con sugerencias que se ciñen más a los gustos de cada uno. Incentiva que los usuarios se creen cuentas, ya que es la única forma de acceder a esta forma de ver ropa. Para los compradores que utilicen la aplicación, aumenta el espectro de ropa sugerida y no se ciñe solo a lo más vendido. Convierte en un pasatiempo la actividad de búsqueda de ropa, aumentando así las posibilidades de venta.

El modelo del sistema recomendador tiene como finalidad diferenciarse de otros sistemas actuales en dos cosas. La primera, utilizar retroalimentación implícita para recomendar ropa y la segunda adaptarse a las realidades cambiantes de consumo de información por parte de las personas (Ma, Feng, Feng, & Wang, 2019).

1.1.1 OBJETIVOS PRINCIPALES

El objetivo principal que engloba a todo el proyecto es desarrollar un sistema recomendador adaptado a las necesidades de los usuarios de hoy en día. Este modelo debería recomendar un conjunto de prendas que se ajusten a los gustos de los clientes. Para crear este modelo de recomendación, es fundamental conseguir que el sistema recomendador cumpla los siguientes requisitos.

1. El modelo esté basado en retroalimentación implícita
2. Adaptar las recomendaciones al nuevo contexto de interfaz de usuario

1.2 METODOLOGÍA

El sistema recomendador se va a desarrollar a través de la realización de un número de etapas divisibles e independientes unas de otras. Además, todo el proyecto está realizado bajo una metodología *Agile*, que es una forma consensuada de desarrollar software (Beck, y otros, 2001).

1.2.1 MÓDULOS DEL PROYECTO

MÓDULO 1 EXTRACCIÓN Y ALMACENAMIENTO DE LAS IMÁGENES Y CARACTERÍSTICAS DE LAS PRENDAS.

En la fase 1 del proyecto, el objetivo es extraer imágenes y características de prendas de vestir de un sitio web mediante técnicas de *web scraping* (Lawson, 2015). Para ello hay que identificar el sitio web de destino, seleccionar las herramientas de *web scraping* y analizar su estructura. Se extraen de cada prenda las imágenes y los detalles escritos de las prendas, como el tipo, el color, la talla, la marca, el precio y la descripción. Los datos extraídos se almacenan en una base de datos para su uso posterior. Al completar con éxito la fase 1, se obtiene un conjunto de datos de imágenes de prendas de vestir y sus características, que se utilizará en las fases posteriores del proyecto.

MÓDULO 2 CODIFICAR LAS IMÁGENES DE LAS PRENDAS A TRAVÉS DE UN AUTOENCODER

En la fase 2 del proyecto, la atención se centra en la codificación de imágenes de prendas mediante *autoencoders* (Kiela & Bottou, 2014). Las imágenes se dividen en prendas para la parte superior del cuerpo y prendas para la parte inferior. Se crean dos autocodificadores utilizando redes neuronales convolucionales. El objetivo es entrenar estos *autoencoders* para que codifiquen eficazmente las imágenes de entrada y generen representaciones comprimidas de las prendas de la parte superior e inferior del cuerpo. En esta fase se implementan los autocodificadores y se entrenan utilizando el conjunto de datos de imágenes

de prendas obtenido en la Fase 1. Las imágenes codificadas serán utilizadas para el desarrollo del modelo final.

MÓDULO 3 DESARROLLAR UN SISTEMA DE RECOMENDACIÓN DE PRENDAS BASADO EN FACTORIZACIÓN DE MATRICES.

En la fase 3 del proyecto, el objetivo es desarrollar un sistema de recomendación de prendas de vestir basado en la factorización matricial (Koren, Bell, & Volinsky, 2009). Se trata de implementar un sistema de recomendación clásico utilizando técnicas de factorización matricial. El sistema de recomendación tiene en cuenta únicamente los datos históricos. La factorización matricial se aplica para descomponer los datos codificados de las prendas en factores latentes, que representan patrones o características subyacentes. A continuación, estos factores latentes se utilizan para generar recomendaciones de prendas para cada usuario. Esta fase se centra en crear candidatos para la recomendación hecha por el modelo final.

MÓDULO 4 DESARROLLO DEL MODELO FINAL

En la Fase 4 del proyecto, la atención se centra en el desarrollo de un sistema de recomendación basado en un modelo de Deep Learning que incorpora una retroalimentación implícita y recomendaciones anteriores. El objetivo es crear un modelo propuesto que utilice los candidatos generados en la Fase 3 para las partes superior e inferior del cuerpo. Considerando todas las combinaciones posibles entre estas prendas, el modelo selecciona las mejores combinaciones basándose en las imágenes codificadas obtenidas en la Fase 2. El modelo propuesto tiene como objetivo proporcionar a los usuarios sugerencias de ropa combinadas, la recomendación final del proyecto.

Con estas 4 fases se pretende diseñar un sistema recomendador completo, que genere recomendaciones de dos prendas, una parte de arriba y una parte de abajo, que combinen entre sí. Esto pretende simular a un generador de outfits, y, el modelo con un futuro desarrollo podría evolucionar a introducir más prendas a la recomendación.

El sistema final recomendará un *outfit* de dos prendas, una de la parte de arriba y otra de la parte de abajo. Esto mejora la cantidad de contenido que la aplicación puede generar. La combinación de prendas aumenta exponencialmente las posibles recomendaciones de la aplicación. Si hay 10 prendas de arriba y 10 prendas de abajo, con los modelos habituales habría 20 posibles prendas. Con este modelo de recomendación las posibilidades aumentan a las 100 combinaciones. El número total de opciones pasará de ser de $M + N$ a ser $M \times N$, siendo M el número prendas de la parte superior y N el número de prendas del tronco inferior. Este número crece de manera exponencial, y puede continuar haciéndolo si se añadiesen más prendas.

1.2.2 AGILE

La metodología Agile se aplica a proyectos de desarrollo de software. Agile proporciona un *framework* que ayuda en el desarrollo del proyecto a priorizar, planificar y ejecutar su trabajo de forma eficiente e iterativa. Aunque este pensado para el desarrollo dentro de un equipo y con cliente, hay puntos que también aplican al desarrollo individual de un proyecto.

Agile te permite adaptarte y crear un proyecto flexible para obtener un producto de calidad al terminar. La naturaleza iterativa de Agile le permite dividir su proyecto en incrementos más pequeños y manejables. Cada incremento representa una parte significativa de la funcionalidad que puede implementarse, probarse y, potencialmente, lanzarse de forma independiente. Al trabajar en iteraciones cortas, cada funcionalidad desarrollada se puede luego conectar para llegar al objetivo final. De esta manera es más sencillo encontrar fallos en el código y solucionarlos, ya que las distintas partes funcionan de manera independiente.

Agile organiza el trabajo y prioriza las tareas en función de las necesidades y los objetivos del proyecto. Al dividir su trabajo en tareas más pequeñas y manejables, que se pueden priorizar y desarrollar en paralelo, se puede mantener un enfoque claro y asegurarse de que está aportando valor de forma constante. Además, se fomenta la mejora continua. Se reflexiona periódicamente sobre el desarrollo, se identifican áreas de mejora e implementan

cambios. Esto permite perfeccionar el desarrollo, aumentar la eficiencia y mejorar la calidad de su trabajo.

La comunicación eficaz es crucial incluso cuando se trabaja solo. En este caso, es con el director del proyecto, con el que se comunican con regularidad los avances, se comparten actualizaciones y se solicitan comentarios ayuda. Esto ayuda a generar confianza y a mantener un sentido de la responsabilidad, incluso siendo un único desarrollador.

Capítulo 2. ESTADO DE LA CUESTION

Los sistemas de recomendación se han convertido en parte integral de las experiencias de compra en línea, ayudando a los usuarios a descubrir productos relevantes en función de sus preferencias y comportamiento. El desarrollo de los sistemas de recomendación se remonta a los inicios del comercio electrónico, cuando los sitios web empezaron a darse cuenta de la necesidad de personalizar las recomendaciones de productos para cada usuario.

A finales de los 90, sitios web de comercio electrónico como Amazon empezaron a aplicar algoritmos de filtrado colaborativo para ofrecer recomendaciones personalizadas a sus usuarios. Este fue el precursor de “las personas que compraron esto también compraron” de Amazon, en el que te recomienda otros artículos que te pueden interesar, hasta los “relacionados” de YouTube.

Los desarrollos se centraban en las tecnologías de la época es decir el email y filtrar información (Goldberg, Nichols, Oki, & Terry, 1992). Esto supuso un importante paso adelante para mejorar la experiencia del usuario y aumentar las ventas. Las recomendaciones se basaban en el análisis de grandes cantidades de datos de los usuarios, lo que permitía identificar patrones y correlaciones que no eran evidentes para los comerciantes humanos.

Los agentes que participan en el sistema recomendador son los usuarios, que utilizan el sistema recomendador y los ítems, los elementos recomendados (Gunawardana & Meek, 2009). Estos generan interacciones que moldean las futuras recomendaciones. Los ítems varían en función de la aplicación a la que se aplique el sistema y van desde artículos de venta online, películas, libros o tweets.

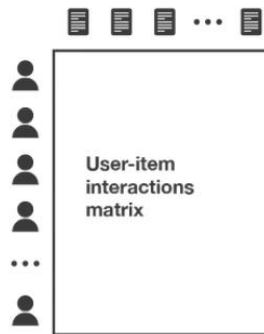


Ilustración 2. Matrix Usuario-Ítem

Durante estos 30 años ha sido una de las ramas del Machine Learning que más se ha desarrollado y más aplicaciones tienen en el día a día. Las recomendaciones de Amazon, Netflix, Spotify, YouTube ... (Jannach, Pu, Ricci, & Zanker, 2021) Los enfoques iniciales de los sistemas de recomendación eran relativamente sencillos y han ido evolucionando a lo largo del tiempo haciéndolos más complejos. Todos ellos se basan en la idea en que los usuarios con gustos y preferencias similares en el pasado tendrán preferencias similares en el futuro. En este enfoque, las interacciones usuario-artículo (como las valoraciones o el historial de compras) se utilizan para identificar patrones y recomendar artículos que gustan a usuarios similares.

Hoy en día se pueden diferenciar los sistemas recomendadores se pueden agrupar como en la siguiente figura (Rocca, 2019).

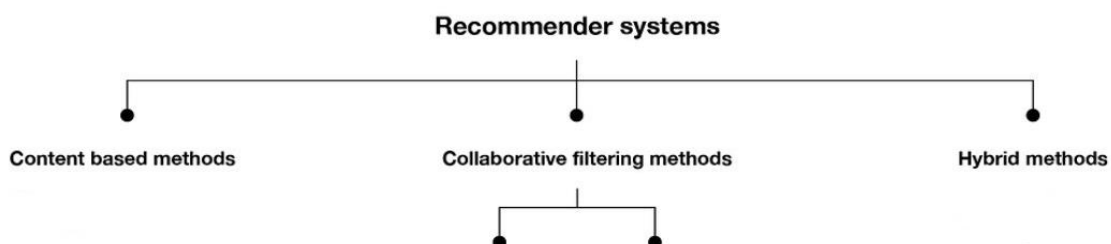


Ilustración 3. Agrupación sistemas recomendadores

En esta se identifican tres grupos, los filtros colaborativos, los modelos basados en contenido y los métodos híbridos. Los filtros fueron los primeros que se empezaron a utilizar, y se han dividido en dos, los basados en memoria y los basados en un modelo. A continuación de estos se desarrollaron los basados en contenido y seguidos por los modelos híbridos, una combinación de los dos tipos de modelo.

2.1 *MODELOS BASADOS EN FILTROS COLABORATIVOS*

El filtrado colaborativo es una de las técnicas más utilizadas en los sistemas de recomendación. Se basa en el principio de que los usuarios que han coincidido en sus preferencias en el pasado probablemente coincidirán también en el futuro (Hu, Koren, & Volinsky, 2008). En el filtrado colaborativo, las recomendaciones se generan basándose en la inteligencia colectiva de un grupo de usuarios. Este enfoque no requiere un conocimiento explícito de los atributos de los artículos ni de las preferencias de los usuarios; en su lugar, identifica patrones y similitudes en las interacciones usuario-artículo. Esta técnica ha ido evolucionando para satisfacer el crecimiento del comercio online, tanto en usuarios como en ítems recomendados.

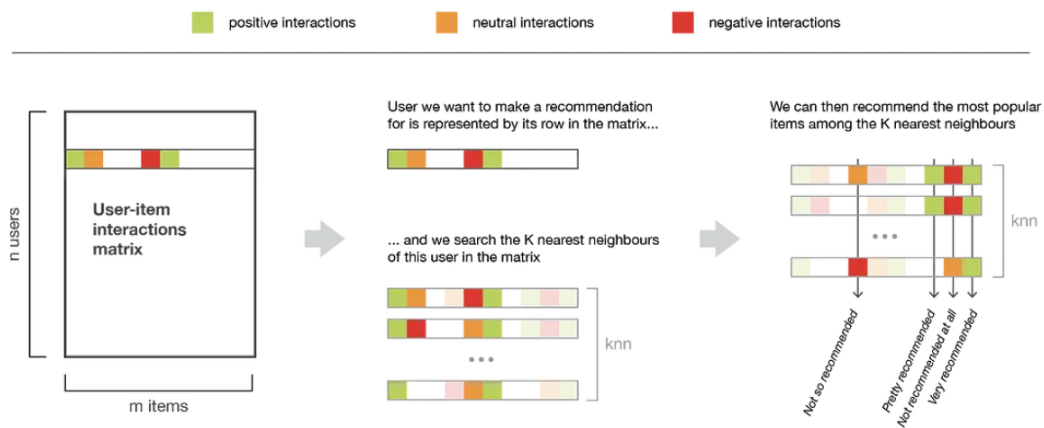


Ilustración 4. Fundamentos filtros colaborativos

2.1.1 RECOMENDACIONES BASADAS EN LA MEMORIA

Las recomendaciones basadas en la memoria, también conocidas como métodos basados en la vecindad, son la primera implementación del filtrado colaborativo. Este enfoque consiste en analizar el comportamiento histórico de los usuarios y encontrar usuarios o artículos similares basándose en sus interacciones pasadas. A continuación, las recomendaciones se generan agregando las preferencias de usuarios o artículos similares.

En las recomendaciones basadas en la memoria, hay dos tipos principales: filtrado colaborativo basado en el usuario y filtrado colaborativo basado en el artículo. El filtrado colaborativo basado en el usuario encuentra usuarios que tienen preferencias similares a las del usuario objetivo y recomienda artículos que han gustado a estos usuarios similares. El fin es recomendar ítems a usuarios, basados en la similitud entre usuarios. Por otro lado, el filtrado colaborativo basado en ítems identifica elementos similares basándose en sus patrones de ocurrencia en las preferencias del usuario y recomienda elementos similares a los que ya han gustado al usuario objetivo. Es decir, el fin es recomendar artículos en función a artículos similares a los que ya ha interactuado el usuario.

Las recomendaciones basadas en la memoria son relativamente sencillas de implementar y han resultado eficaces en muchas aplicaciones. Sin embargo, adolecen de problemas de escalabilidad a medida que aumenta el número de usuarios y artículos. Además, pueden ser propensas al "problema de la escasez" cuando las interacciones usuario-elemento disponibles para realizar recomendaciones precisas son limitadas.

2.1.2 RECOMENDACIONES BASADAS EN MODELOS

Las recomendaciones basadas en modelos, también conocidas como modelos de factores latentes, abordan algunas de las limitaciones de los enfoques basados en la memoria. El mayor problema de los sistemas basados en memoria es la escalabilidad, ya que la técnica

se vuelve impracticable cuando se encuentra un caso de uso del tamaño de Amazon o de los portales de compra online actuales como Shein. El objetivo de esta técnica es reducir el tamaño la matriz, encontrando un espacio latente más pequeño que explique estas interacciones. La factorización de matrices es una popular técnica basada en modelos que aprende factores latentes que representan las preferencias de los usuarios y las características de los artículos. Descompone la matriz de interacción usuario-artículo en representaciones de menor dimensión, capturando la estructura subyacente de los datos.

Los modelos de factorización de matrices suponen que las preferencias de los usuarios y las características de los artículos pueden representarse mediante un conjunto de factores latentes (Koren, Bell, & Volinsky, 2009). Al aprender estos factores latentes, los modelos pueden hacer predicciones para las interacciones usuario-artículo que aún no se han producido. Esto permite realizar recomendaciones personalizadas incluso cuando se dispone de datos limitados o escasos.

La factorización de matrices se puede implementar utilizando varios algoritmos, como la descomposición del valor singular (SVD), la factorización no negativa de matrices o técnicas más avanzadas como las máquinas de factorización y modelos basados en el aprendizaje profundo. Estos modelos pueden manejar conjuntos de datos a gran escala, proporcionar recomendaciones precisas y han tenido éxito en muchos sistemas de recomendación del mundo real.

2.2 MODELOS BASADOS EN CONTENIDOS

Con el tiempo, los sistemas de recomendación evolucionaron e incorporaron técnicas más sofisticadas para mejorar su precisión y pertinencia. El filtrado basado en el contenido se introdujo como enfoque alternativo al filtrado colaborativo. El filtrado basado en el contenido consiste en analizar las características de los artículos y crear perfiles de usuario basados en sus preferencias. Las recomendaciones se hacen comparando los atributos de los artículos con los perfiles de los usuarios.

Los usuarios aportan la edad, sexo o la zona geográfica, entre unas de las muchas características que se pueden obtener de estos. De los ítems, se extraen características en función de lo que se esté recomendando. En Amazon el tipo de producto, tamaño, precio, o categoría serían algunas de las características de esos productos. Toda esta información es la que se recoge y utiliza para realizar la recomendación

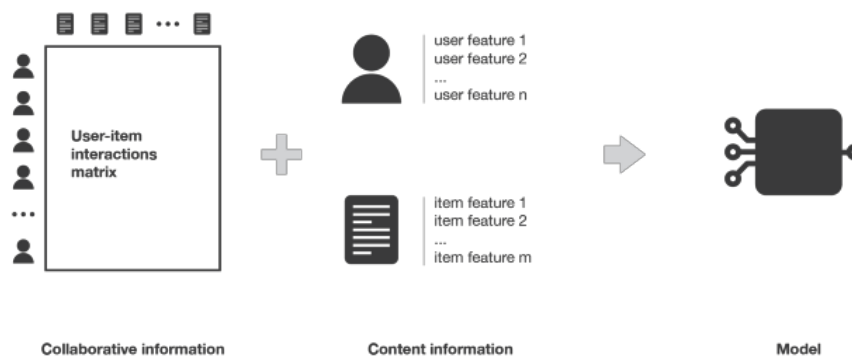


Ilustración 5. Modelos basados en contenido

Estos métodos son más complejos y requieren un mayor trabajo para generar recomendaciones. Comienzan representando los elementos por sus características, como la información textual o los metadatos. Se aplican técnicas de extracción de características, como minería de textos o procesamiento de imágenes, para convertir los datos en una representación numérica que pueda ser utilizada en la recomendación. Se crean perfiles de usuario basados en sus preferencias y se utilizan medidas de similitud para calcular las semejanzas entre artículos o perfiles de usuario. A continuación, se generan recomendaciones sugiriendo artículos similares al perfil del usuario o artículos que le han gustado anteriormente. Los métodos basados en el contenido resuelven el problema del arranque en frío y ofrecen transparencia e interpretabilidad. Proporcionan recomendaciones

personalizadas basadas en atributos explícitos de los artículos, lo que los convierte en un enfoque eficaz en los sistemas de recomendación.

Los métodos basados en el contenido tienen varias ventajas. Pueden resolver el problema del "arranque en frío", cuando los datos del usuario son limitados o cuando hay artículos nuevos sin interacciones históricas. Estos métodos ofrecen transparencia e interpretabilidad, ya que las recomendaciones se basan en atributos explícitos de los artículos.

En general, los métodos basados en el contenido ofrecen una forma eficaz de recomendar artículos en función de sus atributos. Al aprovechar las características de los artículos y los perfiles de los usuarios, los enfoques basados en el contenido pueden ofrecer recomendaciones personalizadas que se ajusten a las preferencias individuales de los usuarios y aborden los retos del problema del arranque en frío.

2.3 MODELOS HÍBRIDOS

Los modelos híbridos combinan múltiples técnicas y metodologías para mejorar la precisión y eficacia de las recomendaciones. Estos modelos aprovechan los puntos fuertes de distintos métodos de recomendación, como el filtrado colaborativo, el filtrado basado en contenidos u otras técnicas avanzadas, para superar las limitaciones y ofrecer recomendaciones más completas y personalizadas.

La idea principal de los modelos híbridos es aprovechar la diversidad de enfoques para mejorar la calidad de las recomendaciones. Al combinar múltiples técnicas de recomendación, estos modelos pueden compensar los puntos débiles de los métodos individuales y ofrecer sugerencias más sólidas y precisas a los usuarios.

La integración de distintas técnicas de recomendación puede lograrse de varias maneras. Algunos modelos híbridos emplean un enfoque en dos fases, en el que las recomendaciones de distintos métodos se generan de forma independiente y luego se combinan utilizando reglas o algoritmos predefinidos. Otros incorporan la hibridación directamente en el

algoritmo de recomendación, combinando los puntos fuertes de los distintos métodos durante el proceso de generación de recomendaciones.

La combinación de filtrado colaborativo y filtrado basado en el contenido es un enfoque habitual en los modelos híbridos. Al incorporar tanto las interacciones usuario-artículo como los atributos de los artículos, estos modelos pueden ofrecer recomendaciones más personalizadas y conscientes del contexto. Por ejemplo, el filtrado colaborativo puede captar las preferencias del usuario basándose en usuarios similares, mientras que el filtrado basado en el contenido tiene en cuenta las características del artículo para sugerir artículos que coincidan con las preferencias del usuario de forma más precisa.

Los modelos híbridos también pueden incorporar técnicas adicionales. En los últimos años, el auge del aprendizaje profundo y las redes neuronales ha impulsado aún más las capacidades de los sistemas de recomendación. Se han aplicado técnicas como las redes neuronales profundas, las redes neuronales recurrentes (RNN) y los modelos transformadores para aprender patrones intrincados a partir del comportamiento de los usuarios y los atributos de los artículos. Estos modelos pueden captar relaciones y dependencias complejas, dando lugar a recomendaciones muy precisas y conscientes del contexto.

2.4 TENDENCIAS ACTUALES

En los últimos años, TikTok se ha convertido en una la red social dominante cautivando a millones de usuarios de todo el mundo con sus vídeos cortos. Detrás de su éxito masivo se encuentra un sofisticado sistema de recomendación que aprovecha el contenido, la retroalimentación implícita y una cantidad interminable de recomendaciones que entretienen a los usuarios.

Aunque TikTok no ha cambiado directamente el diseño de otras redes sociales como Twitter o Instagram, su éxito ha influido en estas plataformas para que se adapten e incorporen esta nueva forma de recomendar contenido. El algoritmo de TikTok recoge la popularidad y el

atractivo de los contenidos de vídeo cortos, las recomendaciones personalizadas, la interactividad y una experiencia de usuario más atractiva. El impacto de TikTok puede apreciarse en la forma en que estas plataformas dan prioridad al vídeo, mejoran el descubrimiento de contenidos, fomentan la interactividad y perfeccionan sus elementos de diseño visual.

Entre todas las cosas que han hecho que TikTok determine el paradigma actual, hay que destacar las siguientes.

- El algoritmo de recomendación de TikTok y su interminable cronología de contenidos han establecido un nuevo estándar para el descubrimiento personalizado de contenidos. En respuesta, tanto Twitter como Instagram han mejorado sus secciones Explorar o Descubrir para dar prioridad al contenido curado algorítmicamente en función de las preferencias e intereses de los usuarios. Este cambio tiene como objetivo proporcionar a los usuarios una experiencia más personalizada y atractiva, similar al enfoque de TikTok en las recomendaciones individualizadas.
- El éxito de TikTok se atribuye en gran medida a su experiencia de usuario altamente interactiva y atractiva. Los usuarios pueden colaborar fácilmente con otros, crear duetos, participar en retos y utilizar una amplia gama de herramientas de edición creativa. Para seguir esta tendencia, Twitter e Instagram han introducido funciones como Twitter Spaces (conversaciones basadas en audio) y la función Remix de Instagram (para colaborar en vídeo). Estas incorporaciones pretenden mejorar la interactividad, fomentar el compromiso y mantener a los usuarios activos en sus respectivas plataformas.
- La interfaz y el diseño visual de TikTok, caracterizados por sus vídeos envolventes a pantalla completa y una navegación basada en el deslizamiento, han tenido una sutil influencia en Twitter e Instagram. Aunque la estructura general y el diseño de las plataformas permanecen prácticamente inalterados, se han introducido ciertos elementos de diseño, como previsualizaciones de vídeo más grandes, una mejor

integración multimedia y una experiencia de navegación más ágil, para que el consumo de contenidos visuales sea más fluido y cautivador.

Estas mejoras que todavía no se han implementado en el comercio electrónico, pretenden ser solventadas con el modelo propuesto. La generación de *outfits* hace que se potencien los tres puntos mencionados. La cantidad de posibilidades que se generan con el modelo propuesto hace que sea factible el carácter continuo de las recomendaciones. Además, fomentan la interacción introduciendo nuevas funcionalidades. Un ejemplo puede ser el bloquear una prenda en concreto y ver distintas posibilidades para generar un conjunto. Por último, el formato de aplicación favorece la recopilación de información para alimentar un modelo basado en la retroalimentación implícita.

Capítulo 3. DESARROLLO DEL SISTEMA

RECOMENDADOR

El sistema recomendar consta de cuatro módulos relacionados entre sí que juntos crean las recomendaciones completas. El sistema trata de recomendar una pareja de prendas o atuendo, formado por una parte de arriba y otra de abajo. La recomendación de conjunto en concreto se hace en base a las compras históricas de un cliente y a la combinación de las dos prendas entre sí.

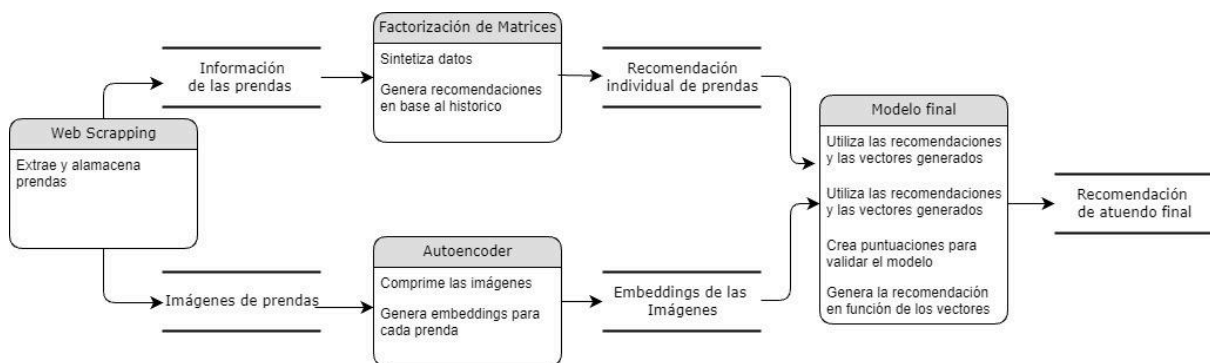


Ilustración 6. Esquema de los módulos del proyecto

El primer módulo genera la base para los demás modelos. En esta fase se recopilan las imágenes y la información de las prendas para los demás módulos. Paralelamente se desarrolla la compresión de las imágenes en *embeddings* y las recomendaciones basadas en un hipotético histórico de compras. Acto seguido, se utilizan las recomendaciones para generar todos los posibles atuendos para cada usuario y se juntan con los *embeddings* generados por el otro módulo. Por último, se introducen los conjuntos en un último modelo que puntúa los atuendos en función de los gustos del usuario.

A continuación, se describen los módulos con más detalle.

3.1 EXTRACCIÓN Y ALMACENAMIENTO DE LAS IMÁGENES Y CARACTERÍSTICAS DE LAS PRENDAS.

Un requisito previo esencial para crear un sistema de recomendación eficaz es disponer de un conjunto de datos completo que abarque una amplia gama de imágenes de prendas y sus características asociadas. Este conjunto de datos sirve de base para entrenar los modelos del sistema recomendador y generar recomendaciones precisas. Adquirir este tipo de datos a gran escala una tarea que requiere el uso de técnicas de *web scraping*.

El *web scraping*, un proceso de extracción de datos de sitios web, ofrece un medio automatizado y eficiente de recopilar información de páginas web. Gracias a esta técnica, podemos acceder a catálogos de prendas y sus atributos correspondientes, como el tipo, el color, la talla, la marca, y la descripción. Estos datos permiten crear un conjunto de datos rico y diverso.

El objetivo principal de la esta fase es extraer imágenes y características de la ropa de un sitio web objetivo. Los datos adquiridos se almacenarán en un formato estructurado dentro de una base de datos, que se accederá durante las demás fases del proyecto.

3.1.1 SELECCIÓN DE LA PÁGINA WEB

La selección de un sitio web adecuado es un paso fundamental en el proceso de *web scraping*. Hay que tener en cuenta varios factores para garantizar que la información extraída es útil para el desarrollo del proyecto. La disponibilidad de una gama diversa de prendas, que las prendas estén bien categorizadas y que la página web permita accesibilidad a la información, son los puntos de decisión para seleccionar una página web u otra.

En primer lugar, se llevó a cabo una investigación exhaustiva para identificar sitios web populares de comercio electrónico de moda que ofrecieran una amplia variedad de prendas.

Se dio prioridad a los sitios web conocidos con amplias colecciones de prendas, interfaces fáciles de usar y disponibilidad de descripciones detalladas de las prendas.

Las páginas seleccionadas han sido de moda femenina. La decisión de centrarse en las páginas web de ropa de mujer para el sistema de recomendación de prendas se basó en tres razones fundamentales. En primer lugar, la ropa de mujer suele ofrecer una mayor variedad de prendas en comparación con ropa de hombre. Esta variedad incluye distintos tipos de prendas, diversos estilos y estampados, lo que proporciona un conjunto de datos más completo para el sistema de recomendación. La ropa de mujer está compuesta de prendas comunes a los hombres como pantalones, camisas o camisetas, a los que se añaden otros tipos de prendas como faldas, blusas o tops. Además, incluso la variedad dentro de las categorías que comparten hombres y mujeres es mayor en favor a la moda femenina. Al incorporar una amplia gama de opciones, el sistema puede identificar mejores patrones de prendas que combinen entre sí, facilitando la identificación de conclusiones

En segundo lugar, el volumen de prendas disponibles para las mujeres es considerablemente mayor que el de las disponibles para los hombres. Tradicionalmente, la industria de la moda ha hecho mayor hincapié en la moda femenina, lo que se traduce en un mercado más amplio y una mayor selección de opciones de ropa para mujeres. Al seleccionar páginas web de ropa de mujer, podemos aprovechar esta gran reserva de datos y garantizar un conjunto de datos más amplio para el sistema de recomendación. Esta abundancia de datos permite desarrollar modelos más robustos.

Por último, es importante reconocer que el comportamiento y las preferencias de compra de hombres y mujeres suelen ser diferentes. Los hombres y las mujeres suelen tener estilos, tendencias de moda y necesidades de ropa distintos. Al centrarnos en las páginas web de ropa femenina, reconocemos la necesidad de un modelo independiente que pueda atender específicamente a las preferencias de las mujeres y ofrecer recomendaciones a medida. Este enfoque permite una experiencia de usuario más personalizada y pertinente, atendiendo a las necesidades y deseos únicos de la población femenina.

Aunque la incorporación de páginas web de ropa masculina podría ampliar el alcance del sistema de recomendación para incluir una base de usuarios más amplia, es crucial mantener un enfoque especializado y específico para garantizar la eficacia del sistema. Al centrarse inicialmente en la ropa de mujer, podemos establecer una base sólida y comprender las características y matices únicos de la moda femenina, lo que permitirá crear un sistema de recomendación más refinado y preciso.

Una vez identificados los sitios web potenciales, se llevó a cabo una evaluación sobre su estructura y sus políticas. Se analizó la estructura HTML del sitio web, y se decidió eliminar aquellos sitios web con generación dinámica de contenidos. Esto significa que páginas web como Zara, no tienen páginas y requieren las prendas se van generando de manera dinámica a la vez que se desliza hacia abajo. Como se puede ver en la figura, la estructura de la página web no facilita encontrar y categorizar la ropa. Estos sitios web quedaron eliminados por la dificultad de sacar información de ellas.

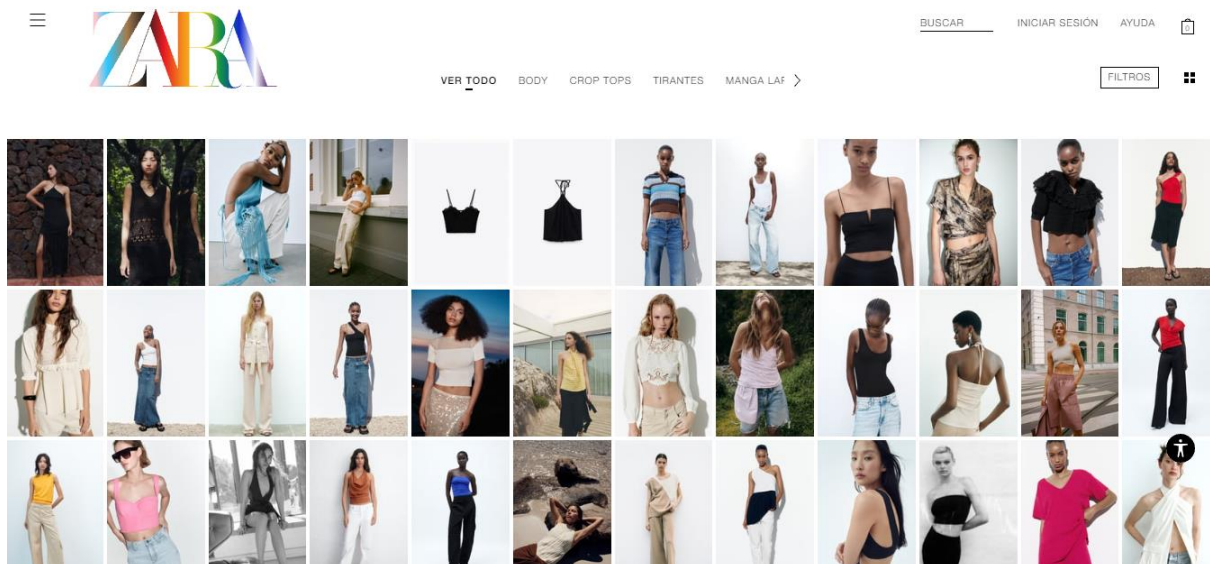


Ilustración 7. Web de Zara actualmente

Además, se revisaron las condiciones de servicio del sitio web y el archivo robots.txt para garantizar el cumplimiento de las directrices sobre el acceso automatizado a su página web.

Finalmente se decidió seleccionar Zalando por cumplir con todos los requisitos anteriores. Es una web de compra online que acumula prendas de una gran variedad de marcas asegurando una base de datos grandes y diversa. Además, la página web está bien organizada por categorías, las prendas están descritas en profundidad, y sus políticas de acceso a robots se permite hacer peticiones a través de software automatizado.

3.1.2 WEB SCRAPING

El *web scraping* es una técnica utilizada para extraer datos de sitios web de forma automática. Consiste en recuperar información de páginas web analizando el código HTML o XML y extrayendo los elementos de datos deseados. El web scraping permite recopilar grandes cantidades de datos de forma rápida y eficaz, lo que proporciona información valiosa para diversas aplicaciones, como la investigación, el análisis y el desarrollo de sistemas.

Los programadores los suelen utilizar para usos diversos como extraer datos específicos de una web, hacer seguimientos de precios, investigación de mercado o para agregar contenidos de diferentes webs. En este proyecto se utiliza el caso de uso más sencillo que es el de únicamente extraer información de las prendas de Zalando.

Para realizar extraer esta información, existen varias librerías y herramientas, cada una con sus propios puntos fuertes y funcionalidades. En este proyecto, se eligieron dos bibliotecas populares, Selenium y BeautifulSoup, para facilitar el proceso.

Selenium es una potente herramienta de automatización web que permite interactuar con sitios web simulando el comportamiento del usuario. Puede automatizar tareas como pulsar botones, rellenar formularios o navegar por páginas web. Selenium es particularmente útil cuando se trata de sitios web dinámicos que dependen en gran medida de JavaScript para la representación de contenidos y la interacción. Al ser el contenido dinámico algo habitual en este tipo de páginas web, se decide incluirlo para pulsar botones o deslizar en algunas de las prendas.

Beautiful Soup, por su parte, es una biblioteca de Python que proporciona una forma cómoda de analizar y extraer datos de documentos HTML o XML. Permite navegar y manipular fácilmente la estructura HTML, facilitando la localización y extracción de elementos de datos específicos. Beautiful Soup se utiliza normalmente para sitios web estáticos o cuando el análisis de HTML es el objetivo principal.

Al combinar las capacidades de Selenium y Beautiful Soup, este proyecto aprovecha las ventajas de ambas librerías. Selenium permite interactuar con sitios web dinámicos, mientras que Beautiful Soup facilita la extracción y el análisis sintáctico de los elementos de datos relevantes de las páginas HTML recuperadas.

Continuando con el método de trabajo *Agile*, esta fase tiene tres módulos independientes. El primero consiste en extraer la información de una prenda de manera correcta y guardarla en un diccionario de Python. En este caso, se utiliza Selenium para acceder a información a la información de los botones, que solo aparece al hacer clic en ellos. Beautiful Soup ayuda a seleccionar las imágenes para los análisis posteriores.

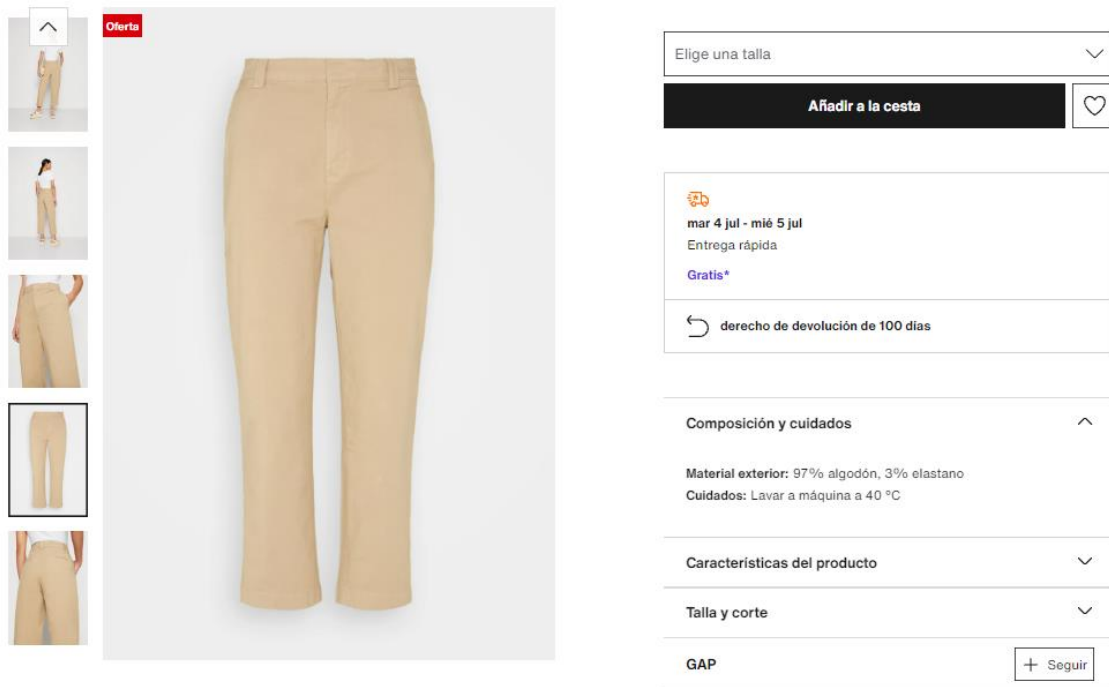
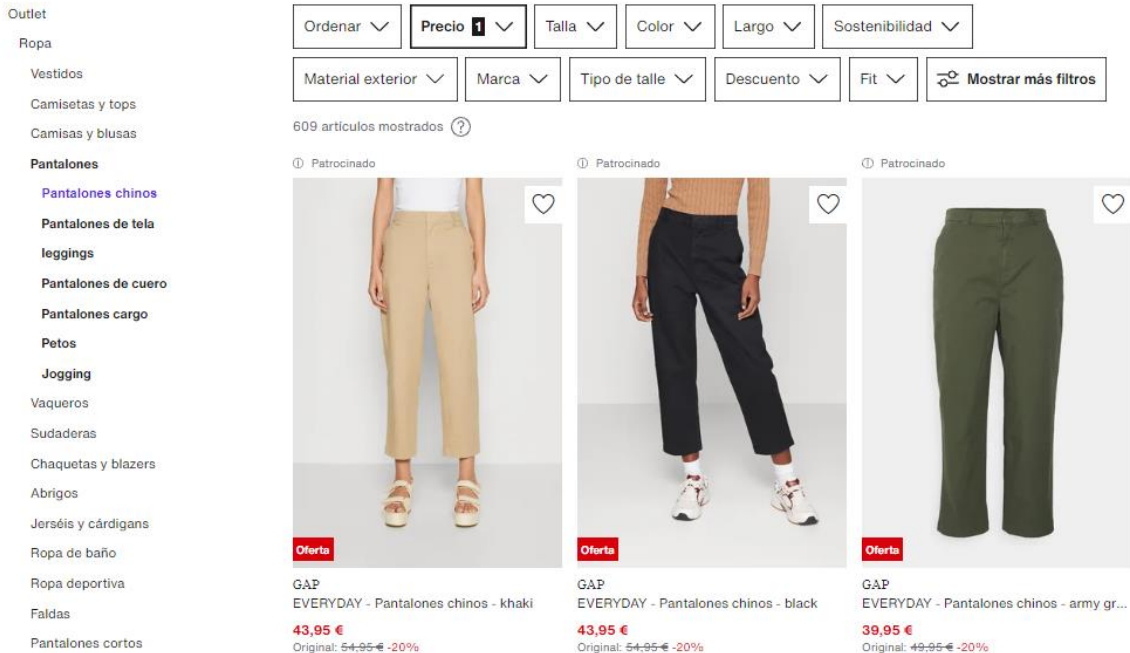


Ilustración 8. Prenda en la web de Zalando

En la imagen, se pueden ver muchas imágenes, algunas con modelo de cuerpo entero y otras con la prenda de cerca. De todas estas imágenes se selecciona únicamente aquella en la que la prenda está sola. Para los procesos posteriores, en los que se quiere comparar si la combinación de las dos prendas le va a gustar al usuario, se tienen que seleccionar ese tipo de imágenes. Por suerte para la selección, este tipo de imágenes tenían una palabra al final que las identificaba y ha facilitado su selección

El segundo módulo es corto y sencillo. Este simplemente extrae los enlaces de todas las prendas posibles con *Beautiful Soup*. Manualmente se copian los enlaces de las categorías que se pueden ver en la figura y la extracción se hace para cada una de ellas. Como se puede ver, hay categorías que no tiene sentido extraer como la de vestidos o petos, ya que estos outfits son de una sola prenda y no se ajustarían al modelo base.

Ofertas en pantalones chinos de mujer





Outlet

Ropa

- Vestidos
- Camisetas y tops
- Camisas y blusas
- Pantalones**
 - Pantalones chinos**
 - Pantalones de tela
 - leggings
 - Pantalones de cuero
 - Pantalones cargo
 - Petos
 - Jogging
 - Vaqueros
 - Sudaderas
 - Chaquetas y blazers
 - Abrigos
 - Jerséis y cárdigans
 - Ropa de baño
 - Ropa deportiva
 - Faldas
 - Pantalones cortos

Ordenar ▾ Precio 1 ▾ Talla ▾ Color ▾ Largo ▾ Sostenibilidad ▾

Material exterior ▾ Marca ▾ Tipo de talla ▾ Descuento ▾ Fit ▾  Mostrar más filtros

609 artículos mostrados 

Patrocinado

Oferta

GAP
EVERYDAY - Pantalones chinos - khaki
43,95 €
Original: 54,95 € -20%

Patrocinado

Oferta

GAP
EVERYDAY - Pantalones chinos - black
43,95 €
Original: 54,95 € -20%

Patrocinado

Oferta

GAP
EVERYDAY - Pantalones chinos - army gr...
39,95 €
Original: 49,95 € -20%

Ilustración 9. Búsqueda de prendas en Zalando

El tercer y último módulo consiste en la paralelización y extracción de la información de todas las prendas extraídas en con el módulo anterior. Este ha sido el módulo más complejo de todos y tedioso de los tres, ya que se entrelaza con la fase de almacenamiento. Al tratarse de muchas prendas, el ordenador se congelaba y el programa dejaba de funcionar. Se realizó con una paralelización de 6 módulos, por las limitaciones de la CPU. Algunos de los mayores problemas han sido la interrupción inesperada de los ejecutores de Chrome que causaba el colapso del programa o el almacenamiento de todas las imágenes.

3.1.3 ALMACENAMIENTO

El almacenamiento se combinaba con el primer y tercer módulo de la parte anterior, para almacenar todos los datos necesarios para completar el proyecto. Tras extraer toda la información de una prenda, esta se almacenaba con la función correspondiente al

almacenamiento. La función transformaba los datos en formato JSON para almacenarlo en MongoDB posteriormente.

Para elegir cual era el mejor sistema, se utilizó el teorema de Eric Brewer, también conocido como teorema CAP, para tomar la decisión. Se trata de un principio fundamental en los sistemas distribuidos que describe el equilibrio entre tres propiedades deseables: Consistencia, Disponibilidad y Tolerancia a Particiones.

- **Consistencia:** La consistencia se refiere a la propiedad por la que todos los nodos de un sistema distribuido ven los mismos datos simultáneamente. En el caso del sistema de recomendación de prendas de vestir, la consistencia estricta no es un requisito estricto. Este tipo de requisitos es común en el sistema bancario, donde no puede haber incoherencias en el sistema. A la hora de recomendar ropa, es menos crítico que los otros dos puntos.
- **Disponibilidad:** La disponibilidad se refiere a la propiedad por la que el sistema acceda a los datos en un tiempo limitado. Para un sistema de recomendación, la disponibilidad es crucial para garantizar una experiencia de usuario rápida e ininterrumpida. Los usuarios deben poder acceder a la aplicación, buscar prendas y recibir recomendaciones con la mayor brevedad posible.
- **Tolerancia a las particiones:** se refiere a la capacidad del sistema para seguir funcionando y ofreciendo respuestas coherentes, pese a estar estos particionados en distintos nodos. Aunque la ropa pueda parecer limitada y no necesitada de datos masivos, los datos de los clientes no.

Teniendo en cuenta los requisitos del sistema de recomendación de prendas, la disponibilidad y la tolerancia a las particiones tienen prioridad sobre la coherencia estricta.

Por ello, el sistema de bases de datos elegido para realizar esta tarea fue MongoDB. Se trata de un sistema de gestión de bases de datos NoSQL que ofrece flexibilidad y escalabilidad para almacenar datos no estructurados. A diferencia de las bases de datos relacionales tradicionales, MongoDB está diseñada para manejar grandes volúmenes de datos diversos y dinámicos sin necesidad de un esquema predefinido.

Esta parte es fundamental para el proyecto, ya que cada las prendas pueden tener peculiaridades, como colores diferentes o distinto número de imágenes. La arquitectura orientada a documentos de MongoDB permite almacenar datos en documentos tipo JSON, lo que lo hace idóneo para manejar datos no estructurados. La sencillez de su almacenamiento también fue un factor para tener en cuenta a la hora de elegir este sistema almacenamiento. Las unidades fundamentales de datos en MongoDB son las colecciones y los documentos.

Una colección en MongoDB es una agrupación de documentos, similar a las tablas de las bases de datos relacionales tradicionales. Las colecciones no tienen esquema, lo que significa que cada documento de una colección puede tener estructuras y campos diferentes. Esta flexibilidad permite almacenar diversos tipos de datos dentro de una misma colección. Las colecciones sirven como contenedores de documentos relacionados y se utilizan para organizar lógicamente los datos en función de sus características o relaciones.

Los documentos son las unidades básicas de almacenamiento de datos en MongoDB. Son análogos a las filas o registros de las bases de datos tradicionales. Un documento en MongoDB es una estructura de tipo JSON formada por pares clave-valor. Estos pares clave-valor representan los campos y los valores correspondientes que componen los datos. Los campos pueden contener varios tipos de datos, como cadenas, números, fechas, matrices o incluso documentos anidados.

Esta base de datos cuenta con tres colecciones una para las prendas con sus características guardadas, otra para las prendas que dan error al guardar o no tienen las imágenes necesarias y otra únicamente para las imágenes. Todas ellas tienen como identificador la subcategoría de la prenda más un número incremental, que será utilizado durante todo el proceso para identificar las prendas.

3.2 CODIFICAR LAS IMÁGENES DE LAS PRENDAS A TRAVÉS DE UN AUTOENCODER

Codificar las imágenes o *image embedding* en inglés, se refiere al proceso de convertir imágenes de alta dimensión en representaciones numéricas de baja dimensión, también conocidas como vectores. Estos vectores capturan las características esenciales de las imágenes, lo que nos permite analizarlas, compararlas y utilizarlas en el modelo final del recomendador. Al comprimir la información presente en las imágenes, podemos reducir la complejidad computacional y mejorar la escalabilidad del sistema de recomendación. Es un proceso bastante habitual cuando se trabaja con imágenes en redes neuronales.

El objetivo principal de vectorizar las imágenes es permitir el análisis de similitudes y la generación de recomendaciones basadas en la compatibilidad visual entre prendas. Los vectores recogen combinaciones de colores, patrones, estilos y formas, pero en un formato mucho más reducido.

3.2.1 FUNDAMENTOS TEÓRICOS

Las redes neuronales son una clase de modelos de aprendizaje automático inspirados en la estructura y el funcionamiento del cerebro humano. Compuestas por neuronas artificiales interconectadas, o nodos, las redes neuronales destacan en la resolución de problemas complejos mediante el aprendizaje de patrones y relaciones a partir de los datos. Estas redes están formadas por capas, cada una de las cuales contiene varios nodos. Las capas suelen estar conectadas completamente conectadas unas detrás de otras y realizan cálculos a partir de la salida de la capa anterior. Existen tres tipos de capas. La capa de entrada recibe los datos iniciales multiplicados por un peso, estos luego son agregados y pasados a una función de activación. La salida se procesa a través de una o varias capas ocultas, que sufren el mismo peso de multiplicación, agregación y función de activación. Finalmente llegan a la capa de salida, donde se realiza la predicción o clasificación final, en función del problema.

La fortaleza de las redes neuronales reside en su capacidad para aprender patrones subyacentes complejos a partir de los datos. Se trata de un proceso iterativo, en el que los pesos se van ajustando en función del error que tenga la predicción. Este proceso de adaptación de los pesos se conoce como entrenamiento. Durante el entrenamiento, la red ajusta los pesos, para minimizar la diferencia entre su predicción de salida y los valores que se quieren predecir. Este proceso de optimización, se realiza habitualmente mediante algoritmos de descenso gradiente, en el que los pesos se ajustan minimizando el error. Con el paso del entrenamiento, se va aumentando la capacidad de la red de generalizar y hacer predicciones precisas sobre datos nuevos no vistos. Las redes neuronales han demostrado un éxito notable en diversos ámbitos, como el reconocimiento de imágenes y del habla, el procesamiento del lenguaje natural y los sistemas de recomendación, lo que las convierte en una herramienta versátil y ampliamente adoptada en el campo de la inteligencia artificial.

Las redes neuronales también son muy flexibles y pueden diseñarse para manejar distintos tipos de datos, como los numéricos, los categóricos e incluso los secuenciales o espaciales. Se han desarrollado distintos tipos de arquitecturas de redes neuronales, como las redes neuronales convolucionales (CNN) para el procesamiento de imágenes o las redes neuronales recurrentes (RNN) para datos secuenciales, con el fin de abordar estructuras de datos y tareas específicas.

Las CNN utilizan filtros para extraer patrones locales y relaciones espaciales de las imágenes, que se conocen como mapas de características. Al encadenar varias capas convolucionales, las CNN aprenden representaciones de las características y captan información visual cada vez más compleja, según van avanzando en la red. Las capas de *pooling*, reducen el tamaño de los mapas de características, preservando los detalles esenciales. Las CNN destacan en tareas como la clasificación de imágenes, la detección de objetos y la segmentación de imágenes, lo que permite avances en campos como la visión por ordenador, los vehículos autónomos y el análisis de imágenes médicas.

Los *autoencoders* son un tipo de arquitectura de dentro de esta tecnología. Constan de una red codificadora y otra decodificadora que trabajan juntas para aprender una representación compacta de las imágenes de entrada. El codificador comprime la información al tamaño del vector que se haya decidido, extrayendo las características visuales más relevantes, mientras que el decodificador reconstruye la imagen original a partir de la representación codificada. Si la imagen tras ser reconstruida por el decodificador es similar o muy parecida a la imagen original, se puede decir que el vector recoge toda la información de la imagen.

Las ventajas del *image embedding* mediante son múltiples. En primer lugar, permite almacenar y recuperar información visual de forma eficiente. Los vectores comprimidos requieren mucha menos memoria que el almacenamiento de las imágenes completas, lo que agiliza el procesamiento y la recuperación. El vector simplificado proporciona agrupa toda la información significativa en menos espacio, lo que nos permite medir la similitud entre prendas utilizando diversas métricas de distancia, como la similitud coseno o la distancia euclídea. Además, un vector más pequeño con la misma información permite la creación de modelos más sencillos con menos parámetros con el uso de estos vectores comprimidos. Esta característica puede ser fundamental para no saturar a los modelos con ruido, lo que puede hacer que no converjan.

3.2.2 ARQUITECTURA DEL *AUTOENCODER*

La diferencia de forma entre las prendas de arriba y las prendas de abajo empuja a que se creen dos modelos para codificar y comprimir las imágenes de entrada. Las imágenes de entrada tienen un tamaño de 288x200 y este será la partida del modelo.

Al ser modelos diferentes, se podría haber diseñado una arquitectura distinta para cada uno. Si embargo, se empieza utilizando la misma arquitectura para los dos modelos y al dar esta arquitectura buenos resultados en ambos modelos se decidió continuar con la misma para ambos. La arquitectura que aparece en la siguiente figura y hace únicamente referencia al codificador, ya que el decodificador es igual.

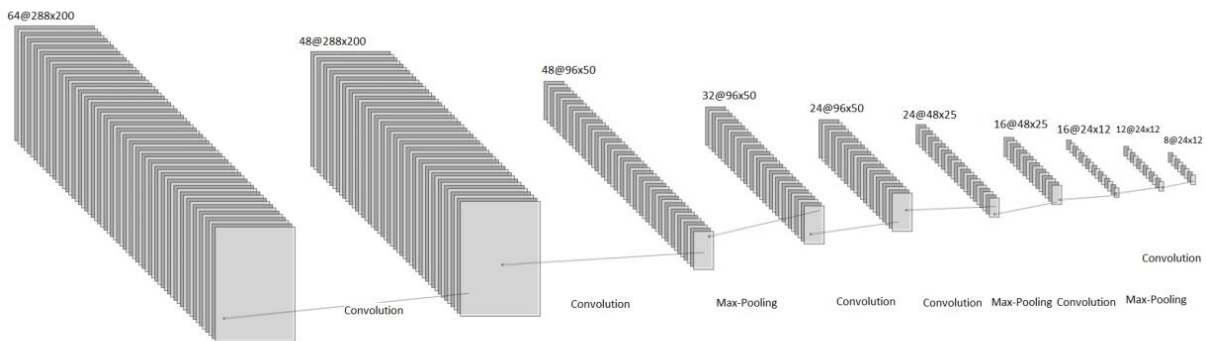


Ilustración 10. Arquitectura del Encoder

El codificador y el decodificador tienen la misma estructura con las mismas capas, pero de forma invertida. La entrada en formato (288, 200, 3) representa las imágenes RGB. Estas imágenes van a través de capas convolucionales y de max-pooling hasta llegar a 8 mapas de características de 24x12.

La función de activación en las convoluciones es Leaky ReLU. ReLU (*Rectified Linear Unit*) es una función de activación muy utilizada en redes neuronales. Aplica la función $f(x) = \max(0, x)$ a una entrada x . En otras palabras, pone a cero todos los valores negativos, mientras que los positivos permanecen inalterados. ReLU es el método preferido por su sencillez y eficiencia computacional. Ayuda a resolver el problema del *vanishing gradient*, es decir que los pesos del principio de la red no se adapten, e introduce la no linealidad en la red. Sin embargo, ReLU tiene una limitación conocida como el problema del *dying ReLU*. En algunos casos, las neuronas pueden quedarse atascadas en un estado de activación negativo

y no recuperarse durante el entrenamiento, lo que las hace ineficaces para el aprendizaje. Esto ocurre porque la parte negativa de la función genera un 0.

Para superar las limitaciones de ReLU, se introdujo una variación llamada Leaky ReLU. Leaky ReLU conserva la estructura básica de ReLU, pero introduce una pequeña pendiente para los valores negativos en lugar de ponerlos a cero. En concreto, Leaky ReLU aplica la función $f(x) = \max(0, 0.01x, x)$ (o una pendiente pequeña similar) a la entrada x . Al utilizar una pendiente pequeña para los valores negativos, Leaky ReLU permite el flujo de gradientes pequeños, evitando el problema de las neuronas muertas.

Entre las capas de las convoluciones se introduce una capa de olvido llamada *Dropout*. La capa de abandono es una técnica de regularización utilizada en redes neuronales para evitar el *overfitting*. Pone a cero aleatoriamente una fracción de las unidades de entrada durante el entrenamiento, lo que ayuda a reducir la dependencia de la red de unidades específicas y fomenta el aprendizaje de características más robustas e independientes. El parámetro de tasa de abandono determina la fracción de unidades a abandonar, y un valor común es 0,5. El abandono mejora la capacidad de generalización del modelo introduciendo ruido y actuando como una técnica de aprendizaje de *ensemble*. Durante la predicción, el abandono se desactiva y se utilizan todas las unidades.

La capa de *Max-Pooling* es una técnica de muestreo descendente muy utilizada en las redes neuronales convolucionales. Divide la entrada en pequeñas regiones que no se solapan y escoge el valor máximo de cada región. La agrupación máxima ayuda a reducir las dimensiones espaciales de la entrada, lo que da como resultado una representación comprimida. Hay otras opciones a esta agrupación como por ejemplo el *Average-Pooling*, aunque son menos utilizados.

Al extraer el valor máximo, se conservan las características más destacadas y descarta los detalles menos importantes. De este modo se consigue la invariancia de traslación, en la que pequeñas variaciones locales en la entrada no afectan significativamente a la salida agrupada. Esta técnica también ayuda a reducir la complejidad computacional de las capas posteriores al disminuir el tamaño de los filtros, y por lo tanto la dimensionalidad de los datos.

Max pooling es una forma eficaz de reducir la muestra de entrada y extraer características esenciales, por lo que es una operación muy utilizada en redes neuronales convolucionales para tareas como la clasificación de imágenes y la extracción de características.

Tras comprimir las imágenes en los 8 mapas, se recorre el camino inverso. Para hacer este camino se utilizan dos tipos de capas, *Conv2DTranspose* y *Upsampling*. *Conv2DTranspose* es una capa que realiza la operación opuesta a Conv2D. Aumenta las dimensiones espaciales de la entrada aplicando una operación de convolución transpuesta mediante filtros *learnable*. *Conv2DTranspose* se utiliza habitualmente en tareas como la generación de imágenes y el aumento de la resolución para recuperar detalles espaciales y generar resultados de mayor resolución.

El *Upsampling* es una técnica utilizada para aumentar las dimensiones espaciales de la entrada. Suele emplearse en combinación con las capas *Conv2DTranspose* para ampliar el tamaño de los datos. El *Upsampling* duplica valores o utiliza métodos de interpolación para rellenar nuevas filas y columnas, ampliando así los datos y pudiendo recuperar la imagen inicial.

A pesar de ser un modelo complejo, al no incluir capas densas, el número de parámetros del modelo es pequeño para los valores habituales en este tipo de problemas. Alexnet, un modelo para la clasificación de imágenes muy utilizado en la industria cuenta con 62.3 millones de parámetros (Krizhevsky, Sutskever, & Hinton, 2012). El modelo diseñado, que es distinto ya que su tarea no es clasificar, pero sirve para tener un punto de partida del que comparar, tiene únicamente 122,723 parámetros.

3.2.3 ENTRENAMIENTO DEL MODELO

El modelo se entrena con el optimizador Adam. Es un algoritmo alternativo al precursor inicial, SGD o Descenso de Gradiente Estocástico. Se caracteriza por cambiar el factor de aprendizaje de manera dinámica según avanza el entrenamiento. El factor de aprendizaje o *learning rate* es un hiperparámetro que regula lo rápido que se actualizan los pesos con los errores del modelo. Se utiliza un batch de 256 y 200 épocas, aunque hay programado un

early stopping para que el modelo deje de entrenar si no mejora con respecto a los datos de validación.

El entrenamiento se externaliza en *Google Collaboratory* que es una herramienta de computación para proyectos de *data science* alojado en la nube. La web te permite ejecutar *jupyter notebooks*, un tipo de archivo que ejecuta código separado por celdas manteniendo una memoria común para todo el documento, alojados en los servidores de Google con sus recursos. Al tratarse de un modelo complejo, que necesita mucha memoria RAM por trabajar con imágenes y gran capacidad de cálculo por las operaciones, los recursos necesarios para entrenar este tipo de modelos no están al alcance de los ordenadores.

Los recursos utilizados para entrenar este modelo han sido 84 GB de RAM, que, al haber solo 8000 imágenes, no generaba problema. También se utiliza una GPU para realizar los cálculos más rápido y el modelo tarda poco en entrenar, que se trata de una Nvidia A100, especialmente diseñada para los cálculos en las redes neuronales.

3.3 DESARROLLAR UN SISTEMA DE RECOMENDACIÓN DE PRENDAS BASADO EN FACTORIZACIÓN DE MATRICES

La primera fase del sistema recomendador en sí consiste en generar los posibles candidatos con las técnicas habituales de recomendación. Estas técnicas están muy desarrolladas y son fácilmente aplicables a través de librerías de Python. Tras la generación de estos candidatos, se pasan a el modelo final para hacer la recomendación última.

3.3.1 EXTRACCIÓN DE DATOS

La consecución de datos para realizar este sistema es compleja ya que requeriría trabajo de campo de recopilación y validación de un sistema de compra online. Las prendas han sido sacadas de Zalando, pero no se ha obtenido ningún dato sobre las compras o la base de

clientes. Al no tener esta información referente a las prendas que se han obtenido de la página web, se ha decidido sintetizar estos datos en función a otra base de datos.

Se han utilizado los datos de una competición lanzada por HyM el año pasado. El objetivo de esa competición no es relevante para este estudio, pero sus datos sí que lo han sido. HyM cedió unos datos de sus productos y las transacciones que habían tenido. Esta información contenía el tipo de prenda y el cliente que la compraba. Lo que se ha hecho ha sido seleccionar los productos más populares de cada categoría, para no tener matrices vacías o poco representativas. Luego se han traducido las categorías de HyM a las extraídas en Zalando y se ha simulado que las transacciones pertenecían a las prendas ya extraídas.

De esta manera, se puede generar una la matriz de usuarios-ítems, necesaria para el proceso de recomendación de prendas.

3.3.2 FUNDAMENTOS TEÓRICOS

La factorización de matrices es una técnica muy utilizada en los sistemas de recomendación para predecir y generar recomendaciones. Consiste en descomponer una matriz de interacción usuario-artículo en dos matrices de rango inferior, que capturan factores latentes o características de los usuarios y los artículos. Estos factores latentes ayudan a representar las características o preferencias subyacentes de usuarios y artículos, lo que permite al sistema hacer recomendaciones personalizadas.

En general, la factorización de matrices pretende aproximar la matriz usuario-artículo original multiplicando dos matrices de rango inferior. Consideremos un ejemplo con una matriz usuario-artículo en la que cada entrada representa la valoración o interacción de un usuario con un artículo. La matriz tiene "m" filas que representan usuarios y "n" columnas que representan elementos. Cada celda de la matriz corresponde a una interacción usuario-elemento, en este caso esta interacción representa la compra de un usuario de ese artículo.

La factorización matricial representa la matriz usuario-elemento R como el producto de dos matrices de rango inferior:

$$R \approx P \times QQ^T$$

Aquí, P es una matriz ' $m \times k$ ' que representa a los usuarios, donde cada fila corresponde a un usuario y cada columna representa un factor latente. Q es una matriz " $n \times k$ " que representa los elementos, donde cada fila corresponde a un elemento y cada columna representa un factor latente. k indica el número de factores latentes, que suele ser mucho menor que el número de usuarios y elementos.

La factorización de matrices tiene como objetivo aprender los valores de P y Q de forma que su producto se aproxime a la matriz original usuario-artículo R . Al factorizar la matriz, podemos rellenar los valores que faltan o predecir las valoraciones de nuevos pares usuario-artículo. Existen múltiples formas de factorizar matrices que van desde su estudio algebraico hasta utilizar redes neuronales similares a los *autoencoders*.

Para este problema se ha decidido elegir la factorización de matrices no negativas (NMF). La factorización de matrices no negativas es una variante de la factorización de matrices que impone la restricción de que todas las entradas de P y Q deben ser no negativas. Esta restricción hace que NMF sea adecuada para escenarios en los que se espera que los factores sean cantidades no negativas.

En el NMF, descomponemos la matriz usuario-elemento R en dos matrices iguales que en la explicación general. Aquí, P y Q son matrices no negativas con las mismas dimensiones que en el caso general de factorización matricial. La restricción de no negatividad garantiza que los factores latentes no sean negativos y permite la interpretación de los factores como componentes aditivos.

El NMF es especialmente útil cuando se trabaja con datos que tienen valores no negativos, como calificaciones en una escala positiva o recuentos de ocurrencias, como es en el caso de las compras. Se ha aplicado con éxito en diversos ámbitos, como la minería de textos, el tratamiento de imágenes y los sistemas de recomendación.

El proceso de factorización en NMF consiste en actualizar iterativamente los valores de P y Q para minimizar la diferencia entre la matriz original R y el producto de P y Q. Se pueden utilizar varios algoritmos, como actualizaciones multiplicativas o mínimos cuadrados alternos, para optimizar la factorización.

Al descomponer la matriz usuario-elemento en factores no negativos, el NMF extrae patrones y representaciones significativos de los datos, lo que permite mejorar las recomendaciones en diversas aplicaciones.

3.3.3 GENERACIÓN DE RECOMENDACIONES

Para la creación del modelo se parte de los datos de las transacciones. En estos se recoge el producto, el cliente y una columna de que ese cliente ha comprado el producto. En primer lugar, se crea una tabla pivote a partir del conjunto de datos. El resultado es una matriz de interacción usuario-producto en la que cada celda representa la compra por un cliente de un producto específico. Los valores que faltan en la matriz original se rellenan con ceros.

A continuación, la tabla pivote se convierte en una representación matricial dispersa utilizando el formato Compressed Sparse Row (CSR). Las matrices dispersas se emplean para almacenar de forma eficiente matrices de gran tamaño con numerosos valores cero, como suele ocurrir en los sistemas de recomendación, en los que los usuarios suelen interactuar sólo con un pequeño subconjunto de elementos.

Para realizar la factorización de matrices, se utiliza la librería sklearn, habitual en este tipo de problemas. El modelo NMF se instanciará con un número especificado de componentes, que determinará la dimensionalidad de los factores latentes. Cuanto mayor sea el número de componentes, más detallada será la representación de los factores subyacentes. En este caso particular, tras analizar la varianza explicada se deciden utilizar 80 componentes, un valor habitual en este tipo de problemas.

El proceso de factorización de la matriz descompone la matriz dispersa en dos matrices no negativas, normalmente denominadas W y H. La matriz W representa la matriz transformada de los datos originales, donde cada fila corresponde a un producto y cada columna representa

un factor latente. Los valores de W indican la fuerza de la asociación entre cada producto y los factores latentes, revelando patrones y relaciones subyacentes que pueden aprovecharse para recomendaciones personalizadas. La matriz H representa lo mismo para los usuarios.

Para generar recomendaciones para un usuario específico, se define la función que funciona de manera independiente. Esta función toma como entradas el ID de usuario, la matriz base, la matriz de coeficientes y el número deseado de recomendaciones (N).

Dentro de la función, los coeficientes del usuario se extraen de la matriz de coeficientes utilizando el ID de usuario. Estos coeficientes representan la asociación del usuario con cada factor latente. A continuación, se calcula el producto punto de la matriz de bases y los coeficientes del usuario, lo que da como resultado las valoraciones previstas para todos los ítems.

Los índices de los elementos mejor valorados se obtienen ordenando las valoraciones previstas en orden descendente, y se seleccionan los N elementos con las valoraciones más altas. Por último, la función devuelve los índices superiores, que representan los elementos recomendados para el usuario. El proceso se repite para los dos modelos, generando N recomendaciones para las prendas de arriba y otras N para las prendas de abajo.

3.4 DESARROLLO DEL MODELO FINAL

El modelo final pretende cumplir con los objetivos planteados en el proyecto, generar una combinación de prendas que le guste al usuario basado en el histórico y en su uso de la aplicación. Esta última fase es una extensión de las recomendaciones generadas en la fase anterior. Recoge las recomendaciones y crea todas las combinaciones posibles.

3.4.1 RANDOMIZACIÓN DE LOS RATINGS

El modelo tiene que estar preparado para tener una retroalimentación implícita. Igual que en el caso anterior, no es factible recoger datos reales sobre el modelo. En este caso se van a

aleatorizar los ratings con combinaciones que arrojen a la luz si el modelo es capaz de identificar formas y colores y combinarlos en base a un gusto.

La idea es ponerle un buen rating a una combinación que sea fácilmente identificable para validar que el modelo aprende de los datos que introducimos y que las recomendaciones no son aleatorias. En este caso se ha decidido utilizar clustering en la parte de arriba para identificar prendas del mismo estilo o color. Se han cogido 4 clústers para ver distintos estilos. Se ha decidido utilizar los segmentos que mejor se diferencian para crear los *outfits*.



Ilustración 11. Imágenes pertenecientes al cluster 1

En las imágenes se presentan patrones marcados, chaquetas de colores rojos, rosas y naranjas, en el primer grupo, y de colores oscuros en el segundo. Como vemos que el color va a ser representado por la parte de arriba, se intenta ver si también identifica patrones de forma como vaqueros o faldas. Por lo tanto, las combinaciones serán prendas coloridas con faldas, y las oscuras con vaqueros. Las primeras tendrán buenos ratings, entre 0.9 y 1 y las segundas entre 0.7 y 0.9, para que no se solapen y no hay confusión. El resto de las

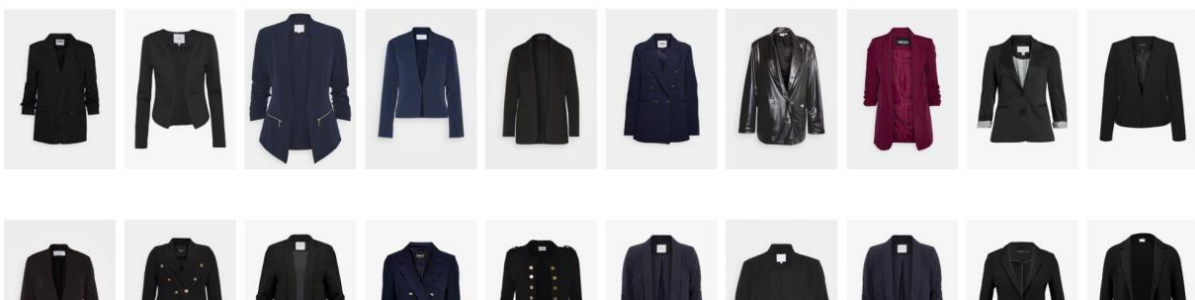


Ilustración 12. Imágenes pertenecientes al cluster 2

combinaciones tendrán un rating entre 0.1 y 0.5. De esta manera, el modelo debería ser capaz de identificar estos patrones y que las recomendaciones incluyan estos dos estilos.

3.4.2 CREACIÓN DEL MODELO

El modelo se entrena con el 25 por ciento de todas las combinaciones posibles. Al tratarse de 4000 prendas de la parte de arriba por 4000 prendas de la parte de abajo, aproximadamente, esto darían unos 16 millones de combinaciones. Al no necesitar el modelo tantos datos para entrenarse, se coge solo una parte.

El modelo consta de dos entradas, los *embeddings* de la parte de arriba y la parte de abajo. La métrica para predecir es el rating, del que hay que tener en cuenta que los valores son aleatorios y además están en rangos entre, por lo que, aunque el error del modelo puede parecer grande, no tiene por qué ser un mal resultado. El resultado del modelo se analizará viendo si las predicciones se ajustan a las suposiciones hechas en el apartado anterior.

La estructura del modelo es la siguiente, aunque está separado en dos imágenes.

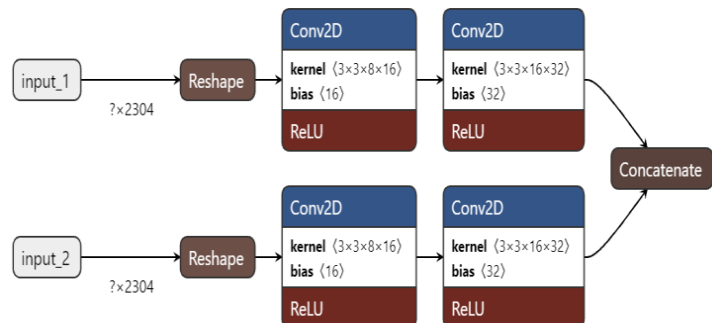


Ilustración 13. Arquitectura modelo final (1)

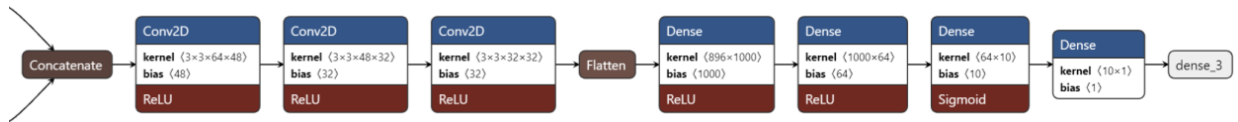


Ilustración 14. Arquitectura modelo final (2)

En primer lugar, se convolucionan las prendas por separado, para posteriormente juntarlas. Siguen avanzando por 3 fases de convoluciones hasta dejar 32 filtros. De esta manera se pretende que el modelo recoja todas las relaciones entre las dos imágenes. El modelo termina en tres capas densas hasta hacer la predicción. Todas las capas tienen ReLU, ya que al ser el modelo más sencillo no tienen los problemas con el gradiente, menos la última que tiene la función sigmoide. Esta función es habitual en las redes neuronales en las que hay que predecir un valor numérico.

El flujo de los datos hasta llegar al modelo final es el siguiente. Primero se generan los candidatos a las recomendaciones, generando 25 prendas para la parte de arriba y otras 25 para abajo. Estas recomendaciones están basadas en la factorización de matrices y en modelo NMF creado. Tras obtener estas recomendaciones, se recuperan los *embeddings* de las imágenes, que por cuestión de rendimiento ya están guardados en una variable.

Posteriormente se crean todas las combinaciones posibles, 625, y se pasan por el modelo final. El modelo les da una puntuación, y en función de esta se cogen aquellos *outfits* con una puntuación más alta. En la selección, se eliminan duplicados para que una prenda no pueda aparecer varias veces en distintos atuendos. Esto se realiza para la explicación y validación del modelo, aunque en el mundo real se tendrían en cuenta otras cosas para tomar esa decisión.

El modelo final resulta ser relativamente sencillo ya que solo tienen un millón de parámetros para los 4 millones de combinaciones con los que se ha entrenado. Esto resulta ser algo

positivo ya que se crea un modelo menos pesado que da un mejor rendimiento con menores tiempos de espera. Esta parte es fundamental en un sistema de recomendación online de generación constante de contenidos, donde una latencia baja es deseable.

Las recomendaciones del modelo aparecerían de una en una, teniendo el formato de la figura. Como se ve, se verían las dos prendas junto con las funcionalidades básicas de cualquier aplicación de compra online, mejorando la calidad de las recomendaciones y la funcionalidad de estas.



Ilustración 15. Ejemplo de la aplicación

3.4.3 ENTRENAMIENTO DEL MODELO

El modelo es entrenado por tandas, ya que, al ser 4 millones de combinaciones, la memoria RAM no puede soportar tanto peso. Los datos se separan en 10 grupos de unas 400.000 muestras y se entrena. La función de entrenamiento es Adam, con el valor habitual de 0.001

en el *learning rate* y los demás valores típicos. Para cada conjunto de datos se utiliza un batch pequeño de 256, igual que en el caso del *autoencoder*. Para cada conjunto de dato se entrenan 10 épocas, con el mismo *early stopping*,

Igual que en el caso del *autoencoder*, el modelo se entrena en *Google Collaboratory* con la versión extendida de RAM de 84GB y la mejor tarjeta gráfica disponible la A100 de Nvidia. En este caso al haber capas densas el modelo es más complejo en referencia al número de parámetros y está más limitado en relación con la memoria. La RAM limitaba el entrenamiento del modelo, por lo que se separaban las ejecuciones en dos. La primera ejecución tenía los primero 5 sets de combinaciones y la segunda los restantes.

Una de las partes importantes de este modelo es que no sea demasiado complejo para que la latencia de la generación de recomendaciones no sea muy alta. En este sentido el modelo ya no necesita ni GPU ni una alta capacidad de la RAM para utilizarse. El tiempo que el modelo necesita para hacer la recomendación de las 625 combinaciones es de 0.37 segundos, un tiempo aceptable ya que esto genera la primera tanda de recomendaciones que el usuario podrá ver mientras se van generando las demás ejecuciones.

Capítulo 4. ANÁLISIS DE RESULTADOS

4.1 WEB SCRAPING

Los resultados de esta sección son sencillos, ya que simplemente se va a comentar el número de prendas que se pudieron sacar de Zalando. Se aprecia que el número de prendas ha sido suficiente, obteniendo 4153 partes de arriba y 3737 partes de abajo. La variedad de prendas es grande y están bien distribuidas. Únicamente en la chaquetas y blazers, que, en el caso de las partes de arriba, sí que representan un porcentaje excesivo.

Distribución de las prendas por categoría

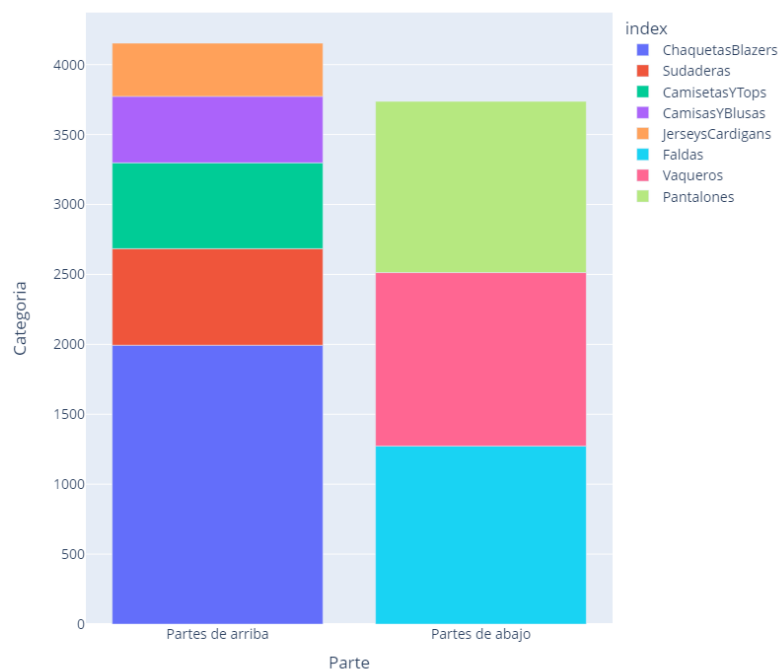


Ilustración 16. Distribución de las prendas por categoría

En total se han utilizado para todo el proyecto 7890 prendas. Otra cuestión a tener en cuenta este módulo es el tiempo utilizado para la extracción de toda la información. Sin contar errores y pruebas que han alargado este proceso, en la ejecución limpia cada prenda tarda unos 30 segundos en extraer la información. Además de estas prendas, también hay otras 4763 prendas que o han dado error o no tenían o pertenecían a una categoría que no se terminó utilizando, como los abrigos. Esto supone un total de unas 12653 prendas, que teniendo en cuenta la paralelización de 6 hilos, darían unas 17.5 horas.

4.2 AUTOENCODER

Los resultados del *autoencoder* se miran desde dos puntos de vista. El primero es desde el punto del error de reconstrucción.

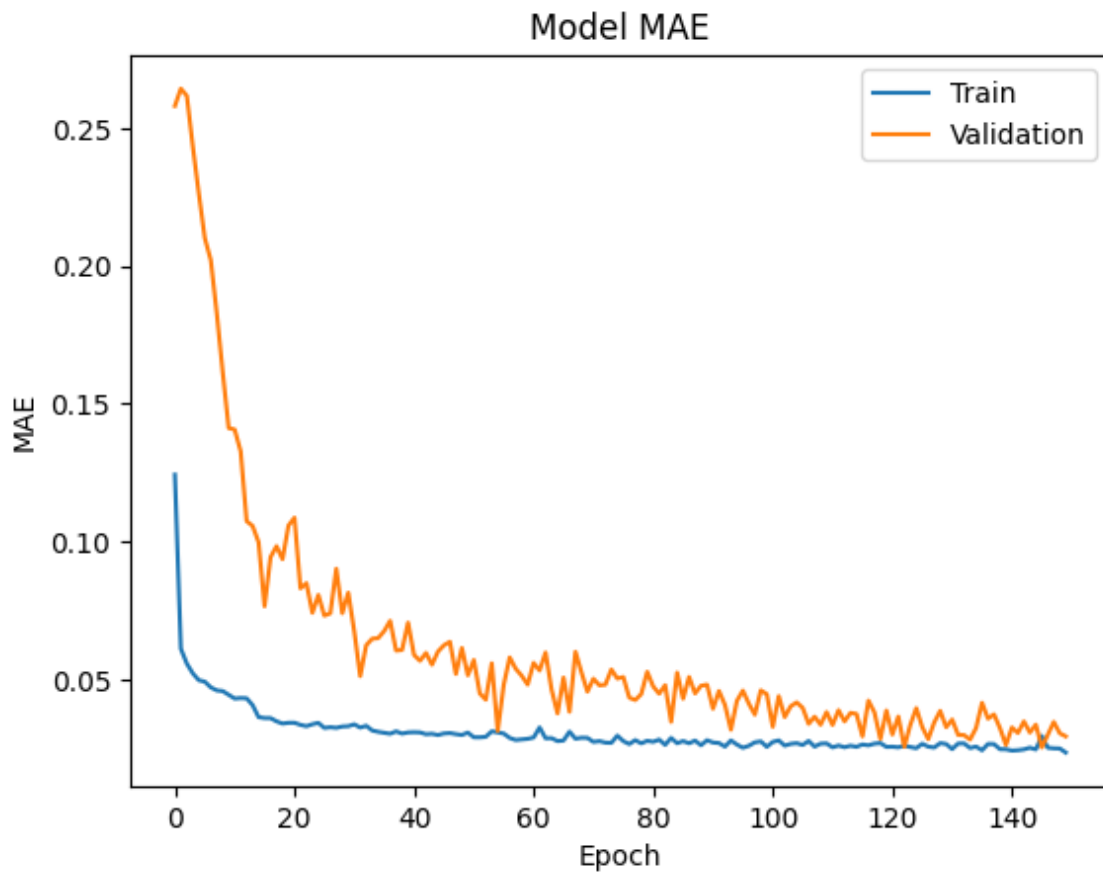


Ilustración 17. Error del Autoencoder durante el entrenamiento

La gráfica corresponde al modelo de las partes de arriba, el cual es el que mayor error ha tenido. Como se aprecia los resultados son interesantes ya que vemos que el modelo mantiene un error de entrenamiento que baja muy poco desde la época 40, al contrario del error de validación. Este no obtiene buenos resultados y no baja definitivamente hasta el final del entrenamiento que salta el *early stopping*. El modelo y entrenamiento de las partes de abajo es mejor, tiene menos error ya que la diversidad de prendas es menor.

La otra parte de la validación es ver como acaban las reconstrucciones, para comprobar que los vectores comprimidos acumulan la gran mayoría de la información.



Ilustración 18. Reconstrucción prendas inferiores

La reconstrucción de la parte de abajo es buena y se identifican bien las prendas, incluso aquellas que son blancas. La reconstrucción de la parte de arriba también es buena, aunque algo más borrosa. Ambas reconstrucciones son suficientes y los vectores capturan la mayoría de la información de la prenda, incluyendo su forma, color y del estilo.

4.3 FACTORIZACIÓN DE MATRICES

Los resultados de esta etapa se valoran observando la varianza que explican las predicciones del modelo. Este dato es dependiente del número de componentes en los que se factorizan las matrices.

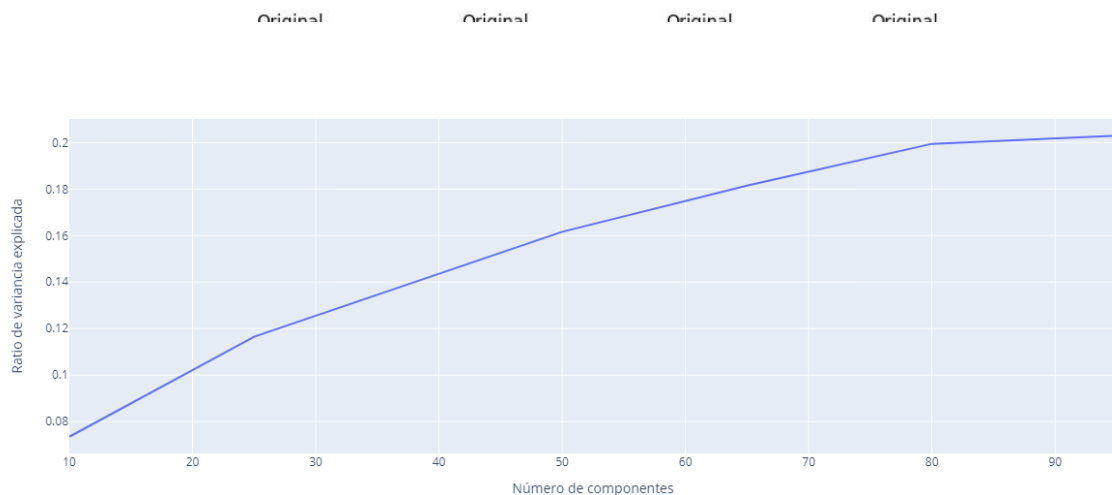


Ilustración 20. Explicación de la varianza en función del número de componentes



Ilustración 19. Reconstrucción prendas superiores

En este caso, vemos que la factorización no se ajusta demasiado a los datos iniciales y no consigue explicar la varianza de los datos originales. El error es bajo, alrededor de un 0.02,

teniendo en cuenta la distribución de los datos. Se aprecia en la gráfica que la estabilidad se empieza a estancar, por lo que no se prueban un mayor número de componentes.

Además, tras observar más detenidamente las recomendaciones llama la atención que no haya apenas blazers y chaquetas. Observando la base de datos de HyM, se puede observar que apenas hay compras de este tipo de prendas, representando un porcentaje muy pequeño. Esto afecta negativamente al sistema ya que, entre todas las prendas, destaca ese grupo como el más numeroso.

4.4 MODELO FINAL

El modelo final, como se adelantó en su desarrollo, no es cercano a 0. El modelo tiene un error de un 0.096, que, considerando la distribución de los datos, es un buen resultado. La mayoría de los datos se encuentran en el rango de 0.1-0.5, y unos pocos se encuentran entre el 0.7 y 1. Además, los ratings dentro de estos rangos son aleatorios por lo que el modelo simplemente será capaz de identificar los rangos a los que pertenece esa combinación, y no el valor exacto, ya que este es aleatorio.

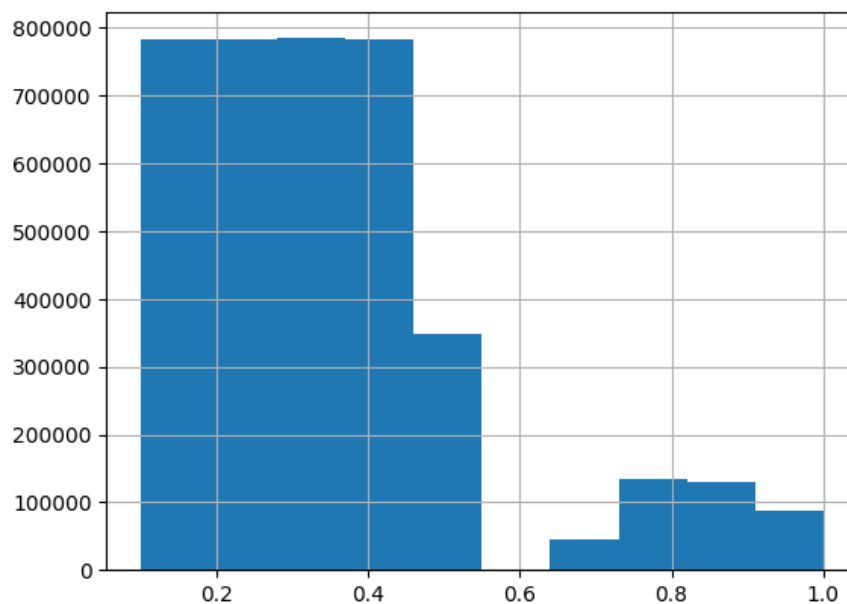


Ilustración 21. Distribución de las puntuaciones generadas

La distribución de la puntuación pone en valor la funcionalidad del modelo, ya que se ha introducido mucho ruido dificultando la identificación de patrones.

Lo más importante del modelo es si ha sabido identificar los patrones que se han impuesto a la hora de la generación aleatoria de la puntuación. Se hicieron 3 grupos de combinaciones. El primero el de la mejor combinación, que juntaba el grupo de prendas coloridas, rojo rosa y naranja, con faldas. Esta combinación tenía una puntuación entre 0.9 y 1. La segunda combinación se trata de las prendas oscuras con vaqueros, y tienen una puntuación entre 0.7 y 0.9. El tercer grupo está compuesto por el resto de las prendas, y tiene la peor puntuación entre 0.1 y 0.5.

Los outfits generados por el modelo se ajustan al ajuste de las puntuaciones de los atuendos que se habían hecho previamente.



Ilustración 22. Atuendos recomendados para un cliente

Los dos primeros outfits pertenecen claramente a la primera combinación, ya que a pesar de no ser las chaquetas que se veían en las imágenes del *clustering*, sí que son de colores vivos y están mezclados con faldas. Los demás outfits también son positivos, ya que representan la segunda combinación, partes de arriba oscuras con pantalones vaqueros.

Aunque para este usuario los resultados hayan salido a la perfección, observando el resto de los clientes, no en todas las recomendaciones se ven los conjuntos preestablecidos representados en los datos. En algunos sí que se encuentran las dos combinaciones presentes, pero en otras muchas la primera no está presente. Esto se debe a la escasez de chaquetas, que tras el filtro de la factorización de matrices no pasan el primer filtro.



Ilustración 23. Atuendos generados para un cliente

En esta recomendación se puede ver que el patrón de oscuro con vaquero se cumple, aunque estos sean de otro color. En este caso la segunda opción preestablecida pasa a ser la primera. En los atuendos 3 y 4 se han recomendado prendas del *cluster* 0, pero en este caso no se ha

podido recomendar con una falda. En el caso de la 4 recomendación, destacar que el modelo confunde una falda con un pantalón corto, debido a su similitud, prendas más pequeñas en las que casi no hay hueco entre las piernas.

Capítulo 5. CONCLUSIONES

El estudio ha presentado un sistema recomendador híbrido de varias fases adaptado a las necesidades actuales. El planteamiento inicial deja claro que se busca que la recomendación y el sistema en sí esté preparado para adaptarse a una generación constante de contenido en un formato móvil. La recomendación de outfits vuelve más dinámico el sistema de compra, fomentando la interactividad entre la aplicación y el usuario. Estas interacciones alimentan otro de los puntos clave, la generación de una retroalimentación implícita que ayude a valorar de una manera más precisa el rendimiento de la aplicación, poniendo en valor también el tiempo de uso frente a las ventas.

El modelo ha demostrado aprovechar al máximo las distintas tecnologías utilizadas en el proceso como son el *autoencoder*, las recomendaciones de la factorización de matrices no negativas y de las redes neuronales como modelo de recomendación. El *autoencoder* ha permitido comprimir la información de tal manera que su uso sea escalable y sencillo en los posteriores modelos. Tras observar las reconstrucciones, apenas se pierde información, habiendo reducido considerablemente los puntos de datos. Esto ha sido posible para todos los tipos de prendas, las prendas de arriba y las prendas de abajo.

La factorización de matrices ha demostrado ser una pieza fundamental para realizar un primer filtrado de cara a la parte final. A pesar de no ser el mejor modelo que se pueda conseguir, ha permitido filtrar aquello que la gente no compraba chaquetas y blazers. Esto ha sido especialmente relevante, ya que en los datos había una sobrecarga de este tipo de prendas en la base de datos inicial.

El modelo último con la ayuda del filtrado inicial y con la información de las imágenes comprimida, ha sido capaz de crear atuendos que se ajustaban a unas reglas preestablecidas. Favoreciendo un patrón de colores, los estampados claros y los oscuros, y un tipo de prenda, los vaqueros y las faldas, el modelo ha sabido identificarlos y favorecerlos en la recomendación. Esto significa que el modelo para estos datos es capaz de identificar estos

patrones de forma y colores, en una etapa sencilla al menos, sin entrar en demasiada complejidad.

Este estudio confirma también la tendencia a la especialización por parte de las herramientas. Para comprimir las imágenes, para hacer un primer filtrado o para agrupar las prendas por sus *embeddings*, da mejores resultados segmentar lo máximo posible. El modelo ha sido diseñado para identificar esos patrones primitivos, pero para un uso real con clientes que compran habitualmente, esta tendencia tiene que verse también reflejada.

Los gustos de las personas son diferentes y lo que a una persona le puede gustar a otra no. Este lógico pensamiento, junto con el desarrollo de este proyecto, arroja que se tendrían que crear varios modelos para distintos modelos de clientes. Partiendo de un modelo base pre-entrenado, habría cientos de modelo o tantos como permita el sistema aguantar, que recomienden ropa de la manera más personalizada posible.

En definitiva, el sistema recomendador completo es capaz de ofrecer una recomendación adaptada a las tecnologías y necesidad de hoy en día. Fomenta el uso de una posible aplicación dinámica, similar a las que hoy en día se utiliza habitualmente. Y, por último, es capaz de identificar patrones, formas y prendas que se están vendiendo, para ofrecer un atuendo que le guste al cliente y quiera seguir viendo más.

5.1 FUTUROS POSIBLES DESARROLLOS DEL PROYECTO

Tras el desarrollo de este proyecto se abren un número de proyectos que se podrían desarrollar en un futuro, haciendo el proyecto viable.

- Recoger datos empíricos para entrenar y validar el modelo propuesto. Este sería el siguiente paso natural a la continuación de este proyecto. El trabajar con datos sintéticos o arbitrariamente aleatorios no asegura que el sistema funcione con datos reales. Este proyecto introduce lo que se conoce como MVP o Producto Mínimo Viable, que es una propuesta ante una futura implementación comercial, para la que se necesitan datos empíricos.

- Desarrollar nuevas propuestas para la validación del modelo propuesto. Encontrar nuevas formas de validar la funcionalidad del modelo sería básico. Establecer grupos de control, monitorizar las puntuaciones y medir la satisfacción general del usuario son algunas de los posibles desarrollos del proyecto
- Desarrollar una interfaz gráfica para el sistema recomendador. Esta parte sería el culmen final y la consecución de un producto completo. El sistema actual presenta dificultades a la hora de enseñar los atuendos, solo puede cargar dos fotos sin ninguna interacción entre ellas. El establecer un modelo virtual, incluso en representación de uno mismo puede ser otro atractivo para utilizar más la aplicación y recoger así más datos.
- Extender las funcionalidades del modelo, eliminando las limitaciones que tienen el diseño actual. Al final, se trata de una primera propuesta que combina únicamente dos prendas específicas, y que puede ser complementada para crear una mejor aplicación. Se puede incluir joyería, accesorios o calzado enriqueciendo la variedad y la calidad del atuendo recomendado.

Capítulo 6. BIBLIOGRAFÍA

- Beck, K., Beedle, M., Van Bennekkum, A., Cockburns, A., Cunningham, W., Fowler, M., . . . others. (2001). *The Agile Alliance*. Obtenido de [www. agilemanifesto. org](http://www.agilemanifesto.org)
- Burke, R. (2007). Hybrid Web Recommender Systems. En R. Burke, *The Adaptive Web: Methods and Strategies of Web Personalization* (págs. 377-408). Berlin.
- Çano, E., & Morisio, M. (2017). *Hybrid recommender systems: A systematic literature review*.
- Cervi, L. (2021). Tik Tok and generation Z. 198-204.
- Goldberg, D., Nichols, D., Oki, B. M., & Terry, D. (1992). *Using Collaborative Filtering to Weave an Information Tapestry*.
- Gunawardana, A., & Meek, C. (2009). *A Unified Approach to Building Hybrid Recommender Systems*.
- Hu, Y., Koren, Y., & Volinsky, C. (2008). *Collaborative Filtering for Implicit Feedback Datasets*.
- Jannach, D., Lerche, L., & Zanker, M. (2018). Recommending Based on Implicit Feedback. En *Social Information Access: Systems and Technologies* (págs. 510-569).
- Jannach, D., Pu, P., Ricci, F., & Zanker, M. (2021). Recommender Systems: Past, Present, Future.
- Kiela, D., & Bottou, L. (2014). Learning image embeddings using convolutional neural networks for improved multi-modal semantics. *Proceedings of the 2014 Conference on empirical methods in natural language processing (EMNLP)*, 36-45.

- Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix Factorization Techniques for Recommender Systems. *Computer*, 30-37.
- Lawson, R. (2015). *Web scraping with Python*.
- Lee, D., & Seung, H. (2000). Algorithms for Non-negative Matrix Factorization. *Advances in Neural Information Processing Systems*.
- Ma, L., Feng, J., Feng, Z., & Wang, L. (2019). Research on User Loyalty of Short Video App Based on Perceived Value—Take Tik Tok as an Example. *2019 16th International Conference on Service Systems and Service Management (ICSSSM)*, 1-6.
- Menon, D. (2022). Factors influencing Instagram Reels usage behaviours: An examination of motives, contextual age and narcissism. *Telematics and Informatics Reports*.
- Peña-Fernández, S., Larrondo-Ureta, A., & Morales-i-Gras, J. (2022). *Current affairs on TikTok. Virality and entertainment for digital natives*.
- Pfalzgraf, B. (Abril de 2020). *How to Use Selenium to Web-Scrape with Example*. Obtenido de Selenium : <https://towardsdatascience.com/how-to-use-selenium-to-web-scrape-with-example-80f9b23a843a>
- Rocca, B. (June de 2019). *Introduction to recommender systems*. Obtenido de Towardsdatascience: <https://towardsdatascience.com/introduction-to-recommender-systems-6c66cf15ada>

ANEXO I: ALINEACIÓN DEL PROYECTO CON LOS ODS

Los Objetivos de Desarrollo Sostenible (ODS) son una agenda global adoptada por la comunidad internacional en 2015, que busca abordar los desafíos más importantes a los que se enfrenta el mundo como la pobreza, la desigualdad, el cambio climático y la degradación ambiental. Estos 17 objetivos interconectados y sus 169 metas representan un llamado a la acción colectiva para lograr un desarrollo sostenible, equitativo y respetuoso con el medio ambiente, con el objetivo de garantizar un futuro mejor para todos.



Producción en colaboración con TROLLBÄCK + COMPANY | TheGlobalGoals@trollback.com | +1 212 529 1010
Para cualquier duda sobre la utilización, por favor comuníquese con: dpicampaign@un.org

Ilustración 24. Objetivos de Desarrollo Sostenible

Las personas deberíamos trabajar por intentar solventar estas problemáticas y este proyecto se alinea con estos objetivos de la siguiente manera:

1. **Producción y consumo responsable.** Es el objetivo principal de este proyecto. Con esta nueva aplicación, las empresas serán capaces de registrar, almacenar y procesar el uso que cada cliente tiene de la aplicación. Se podrá registrar tiempo de uso, hábitos de los clientes, número de prendas que guardan, número de prendas que compran, perfiles de comportamiento ... De esta manera, aparte de obtener información que luego puede ser utilizada con fines comerciales, se puede utilizar para concienciar a los compradores. Con un cuadro de mano se podrán mostrar las estadísticas más significativas a los usuarios para que puedan identificar usos abusivos de la aplicación. Desde las empresas, se pueden recoger datos y realizar modelos para identificar a gente obsesionada con la búsqueda y compra de ropa, para poder notificarles de la situación y así promover un consumo responsable. Este mismo algoritmo se puede utilizar para ventas de segunda mano, cayendo así en una economía circular más sostenible.
2. **Industria, innovación y e infraestructura.** Este sería el primer objetivo secundario. Con un mayor desarrollo de los algoritmos de recomendación, acompañado con inteligencias artificiales que generen ropa, se puede recomendar ropa a la vez que se produce. Si los algoritmos son los suficientemente buenos para conseguir crear ropa que a la gente le gusta, con técnicas de impresión 3D se podría fabricar esta ropa según se va comprando. Esto reduciría inventarios, ropa que se desperdicia por falta de compradores y aumentaría la felicidad general de los clientes. Para llegar a este punto, los algoritmos tendrían que extraer muy bien las características de la ropa para luego poder replicarlas para generar nuevas prendas. Esta nueva forma de producir y vender ropa cambiaría por completo la industria, haciéndola más eficiente e innovadora.

3. **Alianza para lograr objetivos.** Este sería el segundo objetivo secundario. Con este proyecto se desea compartir el conocimiento con todo el mundo, incluido a los países en desarrollo. La falta de infraestructura puede hacer difícil el acceso a esta información, pero una vez conseguido un ordenador e internet, toda esta información estaría disponible. Esto ayuda a distribuir el conocimiento a aquellas personas que quieren aprender para implementar este tipo de técnicas para otros casos de uso en sus países de origen.

ANEXO II

Tabla con las librerías utilizadas y sus versiones.

Librería	Última versión	Descripción
Selenium	4.10.0	Librería de automatización de navegadores web que permite controlar un navegador de manera programática. Es utilizada para pruebas automatizadas y web scraping.
Beautiful Soup	4.10.0	Librería de Python utilizada para extraer datos de archivos HTML y XML. Permite analizar y manipular el contenido de una página web de manera sencilla.
pyautogui	0.9.53	Librería de Python para controlar el mouse y el teclado de una computadora de forma programática. Se utiliza para la automatización de acciones en el escritorio.
sklearn	1.2.2	Librería de aprendizaje automático (machine learning) que proporciona algoritmos y herramientas para tareas como clasificación, regresión y agrupamiento. Se ha utilizado en el desarrollo de la factorización de matrices
numpy	1.22.4	Librería fundamental para la computación científica en Python. Proporciona estructuras de datos eficientes para la manipulación y cálculo de matrices.

tensorflow	2.12.0	Librería de aprendizaje automático desarrollada por Google, utilizada para construir y entrenar modelos de aprendizaje profundo, como redes neuronales.
pandas	1.5.3	Librería para el análisis y manipulación de datos. Ofrece estructuras de datos y herramientas para el procesamiento de datos tabulares.
pymongo	4.4.0	Librería de Python para interactuar con bases de datos MongoDB. Permite realizar operaciones de inserción, consulta y actualización en MongoDB.
PIL	8.4.2	Librería de procesamiento de imágenes de Python que proporciona herramientas para abrir, manipular y guardar imágenes en varios formatos.
plotly	5.13.1	Librería de visualización interactiva en Python utilizada para crear gráficos y visualizaciones de datos dinámicos y atractivos.
