

H&M Personalized Fashion Recommendations

1. Investigación

1.1. Contexto del negocio

Actualmente, los consumidores están aumentando la demanda de sistemas personalizados de recomendación de moda, especialmente en el comercio electrónico. La tienda en línea de H&M es una de las familias de marcas y negocios más importantes a nivel mundial, cuando se trata de la industria de la moda, esta cuenta con 53 mercados en línea y aproximadamente 4850 tiendas, las cuales ofrecen a los compradores una amplia selección de productos de ropa para explorar. Sin embargo, con demasiadas opciones, es posible que los clientes no encuentren rápidamente lo que les interesa o lo que buscan y, en última instancia, es posible que no realicen una compra. Por lo tanto, para mejorar la experiencia de compra, las recomendaciones de productos juegan un papel esencial.

1.2. Objetivo del negocio

Se busca realizar recomendaciones de artículos de ropa disponibles, de acuerdo con el historial de compras de cada cliente. Esto con el fin de mejorar la experiencia de compra de las personas y así, aumentar las ventas.

1.3. Indicadores de éxito

Desde el punto de vista del negocio se establecen como indicadores de éxito: la reducción del porcentaje de las personas que ingresan al sitio web/tiendas y no realizan ninguna compra; el aumento en el porcentaje de satisfacción de los clientes con el sistema de recomendación planteado; y el aumento del porcentaje de aumento de ventas físicas/línea a partir de la fecha de implementación del sistema de recomendación.

1.4. Beneficios

Los beneficios que traería el sistema de recomendación son: el aumento en los ingresos de la empresa H&M, la mejora de la experiencia de compra para los clientes, y un mejor control de la producción de productos con baja demanda.

1.5. SMART Question

¿Cómo podemos recomendar artículos de ropa a los clientes de H&M con la ayuda de un modelo de clasificación no supervisado, teniendo en cuenta sus historiales de compra?

1.6. Entendimiento de los datos

Los datos fueron recolectados de la competencia publicada en la plataforma Kaggle, [H&M Personalized Fashion Recommendations](#). Este contiene 3 archivos .csv de conjuntos de datos y una carpeta de imágenes, los cuales son recursos para la realización de este proyecto:

- customer.csv: conjunto de datos correspondientes a clientes de H&M.
- articles.csv: conjunto de datos de los artículos de H&M disponibles para la compra.
- transactions_train.csv: conjunto de datos que contiene información de las compras realizadas por cada cliente.
- La carpeta de imágenes contiene imágenes de los artículos y son nombradas con cada article_id; las imágenes se colocan en subcarpetas que comienzan con los tres primeros dígitos del article_id; no todos los valores de article_id tienen una imagen correspondiente.

2. Desarrollo

Las herramientas que se utilizan para realizar la exploración y análisis de datos corresponden a Google Colab junto con el empleo de librerías de Python tales como pandas, matplotlib, seaborn, numPy, sklearn, yellowbrick y KModes.

También, se realiza la limpieza de los 3 conjuntos de datos, realizando operaciones para la eliminación de columnas, datos nulos, datos atípicos y filas duplicadas. Así como el reemplazo de valores categóricos y el cambio a los tipos de datos de ciertas columnas.

Para la eliminación de columnas, se tiene en cuenta aquellas que proporcionan información redundante o no relevante para el problema. En cuanto al reemplazo de valores categóricos, se busca agrupar valores categóricos, con el fin de reducir la cantidad de valores únicos para una sola columna, por ejemplo, para la columna perceived_colour_master_name del dataset de artículos, se renombraron los valores de tal manera que se formaran grupos de colores neutrales, cálidos, fríos y un grupo de aquellos que no están definidos.

Posteriormente, se realiza la implementación de dos algoritmos de clustering, K-Means y K-Modes, sobre el conjunto de datos de los artículos. K-Means es un algoritmo de clasificación no supervisada que agrupa objetos en k grupos basándose en sus características. El agrupamiento se realiza minimizando la suma de distancias entre cada objeto y el centroide de su grupo o cluster. Debido a que para este algoritmo se necesitan datos numéricos, antes de aplicarlo, se implementa TFIDF sobre la columna detail_desc, que corresponde a una descripción corta del producto, con el fin de obtener más información del artículo. Asimismo,

sobre las demás columnas categóricas, se realiza One Hot Encoding, lo cual aumenta la dimensionalidad del conjunto de datos, por lo que se emplea PCA para reducirla.

Por otro lado, K-Modes es un algoritmo de clustering para agrupar datos similares en grupos basados en sus atributos categóricos. Identifica las modas o los valores más frecuentes dentro de cada clúster para determinar su centroide. Antes de aplicarlo, se realiza Label Encoding sobre las columnas categóricas y se omite la columna de descripción.

Finalmente, para recomendar artículos a un cliente, se crea el método recommendation, el cual retorna 6 artículos de cada cluster diferente a los que pertenezcan los artículos que haya comprado anteriormente. Para comparar aquello que se le ha recomendado con lo que está en su historial de compras, se crea el método purchasedArticles, el cual permite obtener aquellos artículos que ha comprado el cliente.

3. Despliegue

Con el framework Flask se realizó una página web que tiene las secciones de Login y Home, esto con el fin de simular el escenario en el que un cliente después de iniciar sesión en el sitio web de H&M, pudiera encontrarse con un apartado de recomendaciones de productos se le mostraran los resultados obtenidos a partir de su historial de compras y la aplicación del algoritmo K-Modes.

Algunos de los archivos más relevantes para el despliegue del presente proyecto fueron:

- model.py se importan 3 datasets: uno con todos los artículos, otro que relaciona cada cliente con sus compras y otro que incluye los clusters en donde quedaron clasificados los artículos. En la clase doRecommendations los métodos recommendation y purchasedArticles mencionados anteriormente, implementan dichos datos.
- web_.py en el que se incluye todo el código relacionado con las rutas de login y home. Para la primera de ellas se realizó un simple formulario de inicio de sesión con usuario y contraseña, en este caso, el usuario será el id del cliente y no importa lo que esté en el otro campo. Además, en este archivo se plantean las funciones getImages y getDescriptions que extraen las imágenes y descripciones respectivas de los artículos recomendados y comprados anteriormente por el cliente para ser mostrados.

Cabe resaltar que con el fin de que nuestro modelo pueda mejorar, es necesaria la retroalimentación constante por parte de los clientes, por esto, en la página web para cada artículo se planteó una sección de me gusta y no me gusta, cuya implementación en un futuro ayudará a obtener predicciones más acertadas.