Informe de Proyecto: Sistema de Recomendación de Moda Personalizado

Cristina Hernández Fornaris

Julio de 2025

Índice

1. Resumen

Este informe detalla el proceso de desarrollo de un sistema de recomendación de moda personalizado, abordado como un problema de aprendizaje automático supervisado. El objetivo fue predecir la probabilidad de que un cliente compre un artículo específico, basándose en su perfil y las características del producto. Se utilizó un dataset de transacciones de H&M, enfocado en un segmento de mujeres jóvenes. El proceso incluyó una exhaustiva limpieza de datos, una robusta ingeniería de características, y la evaluación comparativa de dos modelos: un Árbol de Decisión como baseline y un LGBMClassifier como modelo principal. Tras un ajuste de hiperparámetros, el modelo LightGBM final alcanzó un rendimiento excelente, con un ROC AUC de 0.9087, demostrando una alta capacidad para generar recomendaciones relevantes y personalizadas. El proyecto culmina con la implementación de un motor de recomendación funcional que utiliza el modelo optimizado para sugerir productos a los clientes.

2. Introducción

2.1. Contexto y Motivación

En la industria del comercio electrónico de moda, la competencia es feroz y la capacidad de ofrecer una experiencia de compra personalizada es un diferenciador clave. Los sistemas de recomendación ayudan a los clientes a navegar por catálogos extensos, descubrir nuevos productos de su interés y, en última instancia, aumentan la satisfacción del cliente y las ventas. Este proyecto se enfoca en crear un sistema de este tipo, utilizando técnicas de Machine Learning para ir más allá de las recomendaciones genéricas de "lo más vendido".

2.2. Definición del Problema

El desafío se formuló como un problema de **clasificación binaria supervisada**. En lugar de usar métodos no supervisados como el clustering o el filtrado colaborativo tradicional, el objetivo es predecir una variable binaria:

- **Input (X):** Un vector de características que combina el perfil de un cliente y los atributos de un producto candidato.
- **Output (y):** Una etiqueta de 1 (si el cliente comprará el producto) o 0 (si no lo comprará).

Este enfoque nos permite utilizar la potencia de los modelos de clasificación para aprender las complejas interacciones entre los gustos del usuario y las características de los artículos.

3. Preparación y Procesamiento de Datos

3.1. Descripción del Dataset

El proyecto se basó en el dataset *H&M RecSys Young Female Data*, un subconjunto del dataset de la competencia de Kaggle "H&M Personalized Fashion Recommendations". Este dataset es ideal por su tamaño manejable y su enfoque en un segmento demográfico específico. Contiene tres archivos principales:

- `articles.csv`: Metadatos detallados de los productos (tipo, color, departamento, etc.).
- `customers.csv`: Características de los clientes (ID, edad, etc.).
- `young_female_trans.csv`: Un historial de transacciones filtrado para el segmento de interés.

3.2. Limpieza y Unión de Datos

El primer paso consistió en cargar y unir estos tres archivos para crear un DataFrame consolidado. Durante este proceso, se encontró el primer obstáculo técnico: un `ValueError` al intentar unir ('merge') los DataFrames de transacciones y artículos.

Problema Encontrado: La columna clave `article_id` tenía tipos de datos inconsistentes (e.g., `int64` en un DataFrame y `object` en otro).

Solución Adoptada: Se estandarizó el tipo de dato de todas las columnas de ID a `string` (`object`) en todos los DataFrames **antes** de realizar cualquier operación de unión.

3.3. Ingeniería de Características (Feature Engineering)

Esta fue la fase más crítica para la personalización. El objetivo fue crear perfiles numéricos que representaran tanto a los usuarios como a los productos.

3.3.1. Perfiles de Usuario

Para cada cliente, se creó un perfil basado en su historial de compras. Las características clave se infirieron utilizando funciones de agregación ('groupby().agg()'):

- **Características de Comportamiento:** Se calcularon métricas como `avg_price_paid` (gasto promedio) y `total_articles_bought`.
- **Características de "Gusto":** Para inferir las preferencias, se utilizó la **moda** (el valor más frecuente) de las características de los productos que el cliente había comprado. Así se crearon columnas como `fav_product_type`, `fav_color`, y `fav_department`.

3.3.2. Perfiles de Producto

Se utilizaron las ricas características categóricas del archivo `articles.csv` y se enriquecieron con una métrica de popularidad, `times_purchased`, calculada como el número de veces que cada artículo fue comprado.

3.4. Creación del Dataset de Entrenamiento

Para entrenar un modelo de clasificación, se necesitaron tanto ejemplos positivos como negativos.

- **Ejemplos Positivos (target=1):** Se definieron como cada par único '(customer$_id$, $article_id$)'$queaparecee$

- **Ejemplos Negativos (target=0):** Dado que no se dispone de datos explícitos de "no me gusta", se utilizó la técnica de **Negative Sampling**. Para cada ejemplo positivo de un usuario, se generaron dos ejemplos negativos, seleccionando aleatoriamente artículos que dicho usuario nunca había comprado. Esto resultó en un dataset final con una proporción 2:1 de negativos a positivos, un desbalance manejable que permite al modelo aprender eficazmente.

El DataFrame final, con más de 3.3 millones de filas, se guardó en formato `Parquet` para un acceso eficiente en las fases posteriores.

4. Análisis Exploratorio de Datos (EDA)

El análisis del dataset de entrenamiento final reveló insights importantes:

- **Distribución de Edad:** Se confirmó que la población se concentra en el segmento de 21 a 26 años, validando el enfoque del dataset.

- **Preferencias de Producto:** Las categorías "Trousersz "Dress"dominan como las favoritas, lo que sugiere que son productos clave para este segmento.
- **Interacción Edad-Gusto:** Se observó una correlación clara entre la edad y las preferencias. Por ejemplo, la preferencia por "Dress.ªumenta con la edad (dentro del rango 16-29), mientras que la preferencia por "T-shirt"disminuye. Este hallazgo es crucial, ya que valida la hipótesis de que se pueden aprender patrones de recomendación complejos.

- **Correlaciones Numéricas:** Un heatmap de correlación mostró que las características numéricas de ingeniería (‘age‘, ‘avg$_p rice_p aid$‘, $etc.$)$tienenunacorrelacinlinealmuybajaentres, loqueindicaquecadaunaapor$

5. Metodología de Modelado

5.1. Selección de Modelos

Se adoptó una estrategia de comparación para asegurar una elección de modelo robusta.

- **Baseline - Árbol de Decisión (`DecisionTreeClassifier`):** Se eligió un árbol de decisión único con una profundidad limitada (`max_depth=8`) como modelo de referencia. Es interpretable, rápido y representa el bloque de construcción fundamental de modelos más complejos.
- **Principal - LightGBM (`LGBMClassifier`):** Se seleccionó como modelo principal por ser el estándar de la industria para datos tabulares. Su capacidad para manejar interacciones no lineales y su eficiencia computacional lo hacen ideal para este problema.

Se descartaron otros modelos como KNN (ineficiente en alta dimensionalidad) y SVM (computacionalmente inviable para este tamaño de dataset). La Regresión Logística se descartó inicialmente debido a un `MemoryError` causado por la explosión de dimensionalidad del One-Hot Encoding, lo que guió la estrategia hacia modelos basados en árboles que manejan mejor las características categóricas.

5.2. Métricas de Evaluación

Para una evaluación completa, se utilizaron las siguientes métricas:

- **ROC AUC Score:** La métrica principal para comparar el rendimiento general de los modelos, ya que es insensible al desbalance de clases.
- **Informe de Clasificación:** Para analizar la `Precision`, `Recall` y `F1-Score`, especialmente para la clase positiva (target=1).
- **Matriz de Confusión:** Para visualizar directamente los tipos de errores del modelo (Falsos Positivos vs. Falsos Negativos).

6. Resultados y Discusión

6.1. Rendimiento de los Modelos

Tras el entrenamiento y la evaluación en el conjunto de prueba, se obtuvieron los siguientes resultados:

Cuadro 1: Comparación de Rendimiento de Modelos

Modelo	ROC AUC	F1-Score (Clase 1)	Precision (Clase 1)	Recall (Clase 1)
Árbol de Decisión	0.8989	0.73	0.72	0.74
LightGBM (Defecto)	**0.9064**	**0.74**	**0.73**	**0.75**

6.2. Análisis Comparativo

Ambos modelos mostraron un rendimiento muy alto, validando la calidad del proceso. Sin embargo, **LightGBM demostró ser consistentemente superior**. La ventaja más significativa se observó en el **Recall**, indicando que LightGBM es mejor para identificar el conjunto completo de productos que le interesarían a un cliente, reduciendo las oportunidades de venta perdidas (Falsos Negativos).

6.3. Optimización con Ajuste de Hiperparámetros

Para maximizar el rendimiento de LightGBM, se realizó un `GridSearchCV` sobre una submuestra del dataset.

- **Mejores Parámetros:** `{'learning_rate': 0.1, 'max_depth': 7, 'n_estimators': 200, 'num_leaves': 31}`
- **Resultado Final:** El modelo ajustado alcanzó un **ROC AUC de 0.9087**, una mejora medible sobre el modelo con parámetros por defecto.

7. Sistema de Recomendación y Conclusiones

7.1. Implementación del Motor de Recomendación

El modelo LightGBM optimizado y guardado se utilizó para construir una función final. Esta función toma un `customer_id`, evalúa un conjunto de artículos candidatos que el cliente no ha comprado, y devuelve una lista ordenada de los 10 productos con la mayor probabilidad de compra predicha. La salida de la función es una lista de recomendaciones accionables y personalizadas.

7.2. Conclusiones Generales

Este proyecto ha demostrado exitosamente la viabilidad y efectividad de abordar un problema de recomendación como una tarea de clasificación supervisada. A través de una cuidadosa ingeniería de características, fue posible construir un modelo con un alto poder predictivo (AUC $>$0.90) capaz de generar recomendaciones diversas y personalizadas.