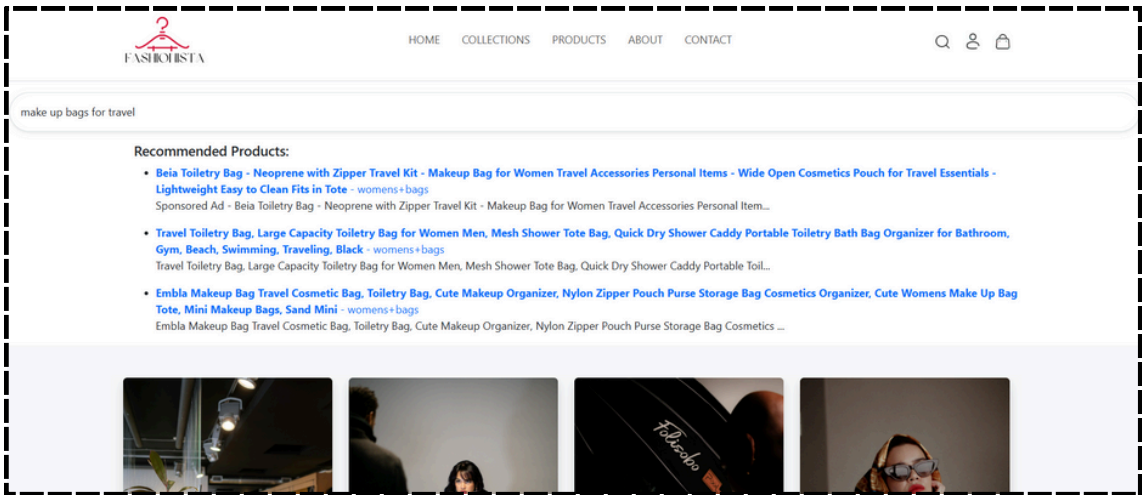
MACHINE LEARNING PROJECT REPORT

DESIGN AND IMPLEMENTATION OF A PERSONALIZED RECOMMENDATION SYSTEM FOR AN E-COMMERCE PLATFORM

FASHIONISTA SMART E-COMMERCE PLATFORM



SECOND YEAR - TDIA

ACADEMIC YEAR: 2024/2025

SUPERVISED BY :M.AZIZ KHAMJANE

PREPARED BY : SIHAM KALACH

Summary

This project focuses on building a full-stack e-commerce website equipped with a recommendation system to enhance user experience and boost sales. The recommendation system comprises two key approaches:

Popularity-Based Recommendation System: Designed for new customers, this system recommends products based on overall popularity, ensuring a straightforward and effective introduction to the platform.
Content-Based Recommendation System: For businesses without existing user-item purchase history, this system utilizes a textual clustering analysis approach. Product descriptions and categories are combined, vectorized using TF-IDF (Term Frequency-Inverse Document Frequency), and grouped into clusters with K-Means clustering. User search queries are matched to clusters using cosine similarity, providing personalized recommendations by fetching the most relevant products from the identified cluster.

Technologies Used:

- Front-end: React
- Back-end: Django Rest Framework
- Database: Postgresql
- Machine Learning: Scikit-learn (TF-IDF, K-Means, Cosine Similarity)
- Data Collection: Web scraping from Amazon using BeautifulSoup

This system ensures a user-friendly interface powered by an intelligent recommendation mechanism, enabling businesses to attract and retain customers effectively.

1. Introduction

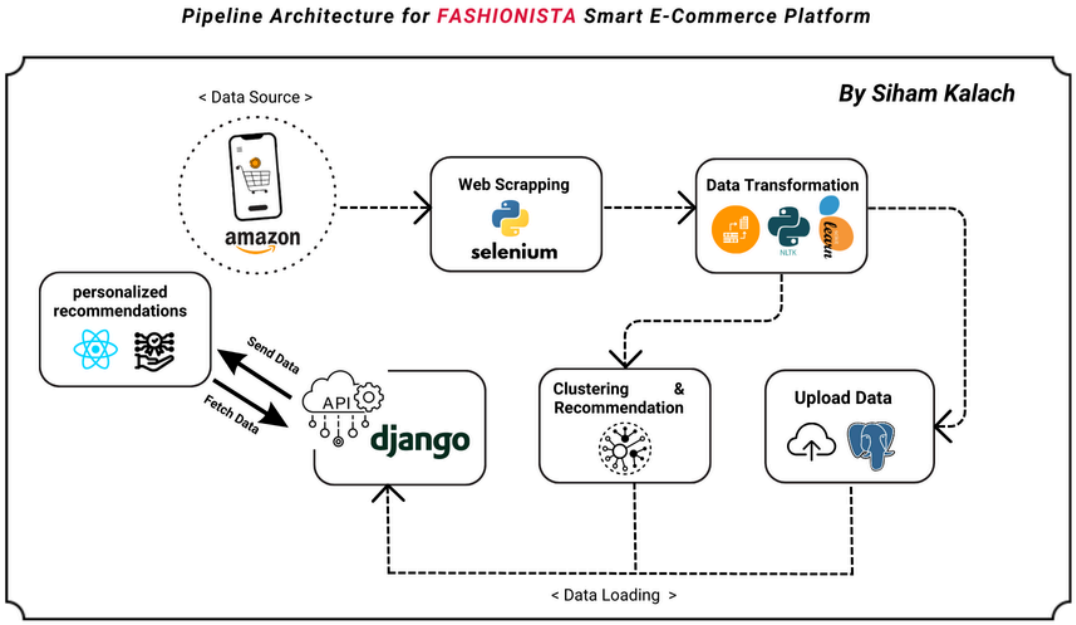
In the rapidly growing world of e-commerce, providing a personalized shopping experience has become essential for attracting and retaining customers. However, many businesses, particularly new ones, face significant challenges in delivering tailored recommendations to users due to the lack of historical user-item interaction data.

This project aims to address this problem by developing a full-stack e-commerce platform equipped with an intelligent recommendation system. The primary goal is to create a system capable of guiding new customers effectively while ensuring personalized product suggestions for returning users, even when user-item purchase history is unavailable.

By implementing a dual-approach recommendation system—one based on product popularity for new customers and another leveraging clustering techniques for personalized suggestions—this project provides a robust solution for enhancing the overall shopping experience. The integration of modern technologies, including React for the front end, Django for the back end, and machine learning techniques such as TF-IDF and K-Means clustering, ensures a scalable and efficient system.

Solving this problem benefits businesses by increasing user engagement, improving product visibility, and ultimately boosting sales. It also enhances customer satisfaction by offering a seamless and relevant shopping journey.

The Architecture :



2. System : Content-Based Recommendation System

2.1 Data Collection:

The Content-Based Recommendation System relies on data collected from Amazon to analyze and categorize products effectively. The data collection process is executed using Selenium, a web scraping library in Python. This step involves extracting product details, including:

- Product Descriptions: Textual content describing the product's features.
- Product Categories: Categorization of products into groups such as electronics, fashion, etc.
- Product Rating
- Images: URLs of product images for visualization.

The collected data serves as the foundation for clustering and recommendation tasks, ensuring that the system provides personalized recommendations .

2.2 Data Transformation and Cleaning:

Following data collection, the next crucial step is data transformation and cleaning to prepare the dataset for machine learning processing. The objective is to standardize the format, eliminate noise, and create a well-structured dataset for model training. The following transformations are applied:

Price Formatting:

- Prices with invalid formatting, such as double dots (..), are corrected by replacing them with a single dot (.) to ensure the correct numerical representation.

Rating Processing:

- Ratings are processed to extract the numerical value. Ratings with the phrase "out of 5 stars" are split to keep only the numerical part (e.g., "4.5" from "4.5 out of 5 stars").
- Products with missing or invalid ratings ("N/A") are excluded from the dataset.

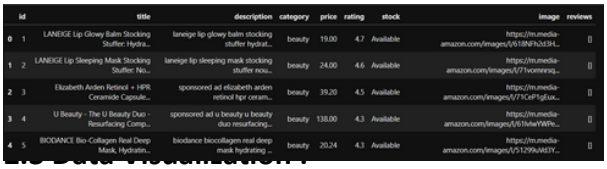
Text Preprocessing:

- Lowercasing: All text is converted to lowercase to ensure consistency in the dataset.
- Removing Punctuation and Numbers: Extraneous characters, including punctuation and numbers, are removed to focus on the textual content.
- Stop Words Removal: Common stop words (e.g., "the", "is", etc.) are removed to reduce noise and improve the model's focus on meaningful words.
- Tokenization: The text is split into individual tokens (words) to facilitate further analysis.

Handling Missing or Invalid Data:

- Products with missing or "N/A" values in key fields such as rating or price are excluded from the dataset to maintain quality and consistency.

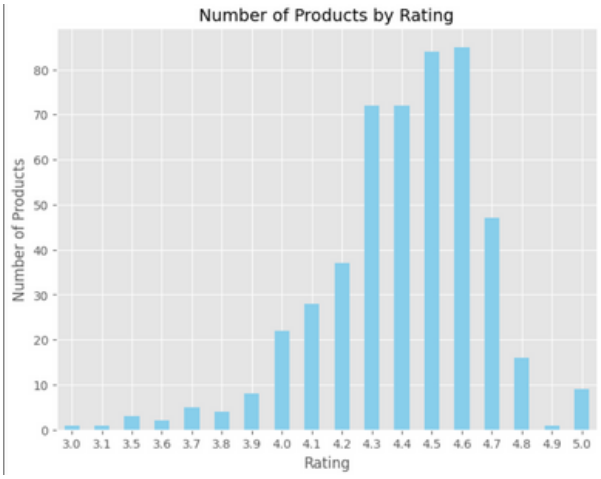
Exemple of dataset :



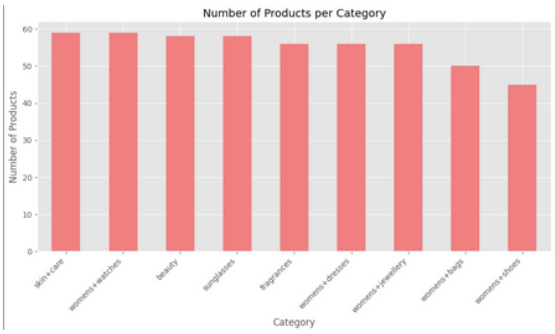
	id	title	description	category	price	rating	stock	image	reviews
0	1	LANEIGE Lip Glowy Balm Stocking Stuffer: Hydra...	laneige lip glowy balm stocking stuffer hydrat...	beauty	19.00	4.7	Available	https://m.media-amazon.com/images/I/618NFh2d3H...	[]
1	2	LANEIGE Lip Sleeping Mask Stocking Stuffer: No...	laneige lip sleeping mask stocking stuffer nou...	beauty	24.00	4.6	Available	https://m.media-amazon.com/images/I/71vomnrsq...	[]
2	3	Elizabeth Arden Retinol + HPR Ceramide Capsule...	sponsored ad elizabeth arden retinol hpr ceram...	beauty	39.20	4.5	Available	https://m.media-amazon.com/images/I/71CoP1gEux...	[]
3	4	U Beauty - The U Beauty Duo - Resurfacing Comp...	sponsored ad u beauty u beauty duo resurfacing...	beauty	138.00	4.3	Available	https://m.media-amazon.com/images/I/61hAeYWPlv...	[]
4	5	BIODANCE Bio-Collagen Real Deep Mask, Hydratin...	biodance biocollagen real deep mask hydrating ...	beauty	20.24	4.3	Available	https://m.media-amazon.com/images/I/51299cMdTY...	[]

Data visualization plays a key role in understanding the distribution and relationships within the dataset. To gain insights into the products, the following visualizations were generated:

1. *Number of Products by Rating :*



2. Number of Products per Category:



2.4 Clustering & Product Categorization :

After data transformation and cleaning, the next step in the product recommendation system involves preparing the data for clustering and similarity calculations. This process ensures that the combination of product descriptions and categories is transformed into numerical representations, which can be effectively used for clustering and querying.Here's the detailed methodology:

TF-IDF Vectorization:

The product data is represented by combining the category and description fields into a single textual feature. This combined feature captures both the contextual meaning of the product descriptions and their categorization, providing a richer representation of each product.
To achieve this, the TF-IDF (Term Frequency-Inverse Document Frequency) vectorizer is employed. This technique assigns importance to terms based on their relevance in describing individual products and their rarity across the entire product catalog. The mathematical formulations for TF, IDF, and TF-IDF weight are outlined below:

1.Term Frequency (TF):

- Measures how often a term ttt appears in the combined "category + description" field of a product.
- Formula :

$$\mathrm{TF}(t,d) = \frac{\text{Frequency of term } t \text{ in document } d}{\text{Total number of terms in document } d}$$

2.Inverse Document Frequency (IDF):

- *Measures how unique a term ttt is across the entire product catalog (set of documents D).*
- *Formula :*

$$\text{IDF}(t) = \log\left(\frac{|D|}{1 + |\{d \in D : t \in d\}|}\right)$$

| D | : Total number of documents (products) in the catalog.

| {d∈D:t∈d} | : Number of documents in which the term t appears.

3.TF-IDF Weight:

- *Assigns higher weights to terms that are frequent in the specific "category + description" combination but rare across the entire product catalog.*
- *Formula :*

$$\text{TF-IDF}(t, d) = \text{TF}(t, d) \cdot \text{IDF}(t)$$

TF-IDF Configuration:

To enhance the representation, the TF-IDF vectorizer is configured with the following parameters:

- **Stop Words:** Common English stop words (e.g., "and," "is," "the") are removed to focus on meaningful terms.
- **N-gram Range:** The parameter ngram_range=(1, 2) is applied to capture both unigrams (single words) and bigrams (pairs of consecutive words). For example:
  - Unigrams: "wireless," "headphones."
  - Bigrams: "wireless headphones."
- **Maximum Document Frequency** (max_df=0.95): Filters out terms appearing in more than 95% of the product catalog, removing overly common terms.
- **Minimum Document Frequency** (min_df=2): Filters out terms appearing in fewer than two product descriptions, removing noise from very rare terms.
- **Maximum Features** (max_features=5000): Limits the feature space to the top 5000 terms for computational efficiency.

2.5 K-Means Clustering Algorithm :

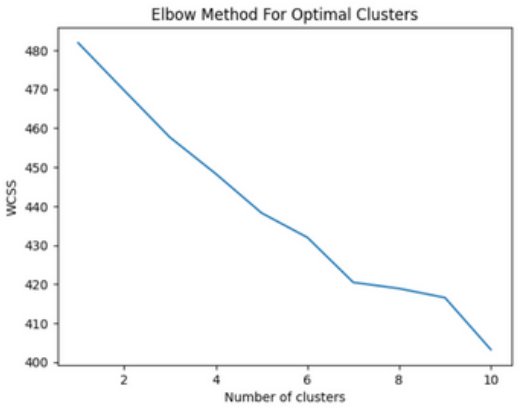
1.Clustering with K-Means:

K-Means is an unsupervised machine learning algorithm that efficiently partitions data into distinct clusters. In this project, it was applied to cluster the products in a way that similar items are grouped together. By leveraging the clusters, we can recommend products that are most likely to match a user's search query.

Using the product embeddings (numerical vectors), K-Means is applied to cluster the products into distinct groups. Each group (or cluster) represents a category of products that share similar characteristics, such as toiletry bags, beauty products, or womens-watches items. The K-Means algorithm assigns each product to one of the k clusters.

2.Choosing the Number of Clusters (k):

The number of clusters, k, is a crucial parameter in the K-Means algorithm.To find the best number of clusters we used the **Elbow Method** . **The Elbow Method** involves plotting the sum of squared distances **(inertia)** for different values of k and selecting the value of k where the rate of inertia reduction slows down (forming an "elbow").



3.Assigning Products to the Nearest Cluster:

Once the K-Means algorithm has clustered the products, we calculate the **cosine similarity** between the search query (e.g., "travel") and the centroids of each cluster. This step involves comparing the search term's vector representation with the cluster centroids (the average representation of products in each cluster).

Cosine Similarity Formula :

$$\text{cosine\_similarity}(A, B) = \frac{A \cdot B}{\|A\|\|B\|}$$

Where:

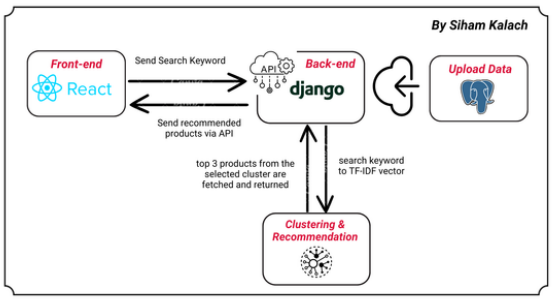
- A is the vector representation of the search query (in this case, the TF-IDF vector of the search term, e.g., "travel").
- B is the vector representation of the cluster centroid (the average vector of the products in that cluster

3.Product Recommendation:

Once the most similar cluster is identified, the products belonging to that cluster are recommended.

2.6 Model Deployment in Web Environments :

To apply the machine learning model we built, we integrated it into a full-stack eCommerce website. The model, which uses K-Means clustering to provide personalized product recommendations, is connected to the website's backend. When a user enters a search query, the system processes the input, calculates cosine similarity with the cluster centroids, and dynamically displays relevant products based on the query. The frontend of the website is responsive and user-friendly, ensuring that the product recommendations are displayed seamlessly to enhance the user experience.here is the architecture how it works:



2.7 Limitations and Future Work:

The current implementation of the recommendation system leverages clustering and keyword similarity to suggest products to users. While this approach provides relevant results based on user searches, it does not account for users' purchasing behavior. This limitation restricts the system's ability to generate recommendations that align with users' actual buying patterns, which are crucial for enhancing personalization and customer satisfaction.
As future work, the recommendation system can be enhanced to include purchase-based recommendations. By analyzing user transaction history and purchase behaviors, the system can provide more accurate and tailored product suggestions. Implementing collaborative filtering based on purchase data, or developing hybrid approaches that combine clustering, keyword similarity, and purchasing behavior, could significantly improve recommendation accuracy. Furthermore, integrating advanced machine learning models capable of capturing user preferences dynamically and in real-time will help ensure the system adapts to evolving user needs. These enhancements will not only improve the overall user experience but also increase conversion rates and customer loyalty.

2.8 Conclusion:

This project successfully demonstrated the application of K-Means clustering to create a machine learning-driven recommendation system for an eCommerce platform. The integration of the model with a full-stack web application provided a personalized and user-friendly shopping experience. The system effectively mapped user queries to product clusters, delivering relevant recommendations based on the semantic similarity of search terms and cluster centroids.

Although the project has limitations, it highlights the potential of combining machine learning algorithms with modern web development to address real-world challenges.

2.9 References :

- **https://www.kaggle.com/code/shawamar/product-recommendation-system-for-e-commerce**
- **https://www.youtube.com/watch?v=e6blyg4GFto&list=PL2aJidc6QnyPntmbmtd4QNSoPS-AwfhiA**
- **https://github.com/kritebh/ecommerce-django-react**

DEMO :