**Dynamic Multi-Modal Product Recommendation System**

# 1. Introduction

This project is a Dynamic Multi-Modal Product Recommendation System that combines both Natural Language Processing (NLP) and Computer Vision (CNN-based image analysis) to recommend products based on their textual description and associated images. It is built using Python, pandas, scikit-learn, TensorFlow/Keras, and Streamlit for web deployment, utilizing cosine similarity to identify and suggest the most relevant items to users.

# 2. Objectives

- Recommend similar products based on a given product.

- Fuse textual and image features to enhance recommendation accuracy.

- Visualize results interactively through a web app using Streamlit.

- Deploy the app for external use via a secure tunnel.

**3. Dataset Information for Product Recommendation Engine**

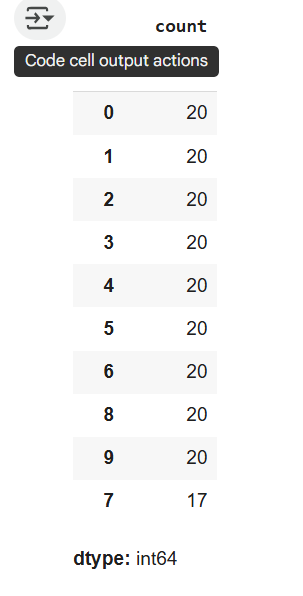
For this project, a subset of the **Kaggle Fashion Product Recommendation Dataset** (originally containing over **44,000+ fashion products**) was utilized. This dataset includes a variety of fashion items with structured metadata and product images.

**Dataset Selection & Preparation:**

* From the original dataset, a focused subset was created by selecting:
  + **10 different fashion product types** (e.g., **Tshirts, Kurtas, Jeans, Dresses, etc.**).
  + For each product type, **20 brand variations** were selected for diversity.
* This resulted in a filtered dataset of **200 product entries** to be used in this project.

**Files Used in the Project:**

**1. final\_products.csv**

* This file contains the **filtered subset** of the original dataset.
* It includes:
  + productDisplayName – A clear product title (e.g., "Vishudh Women Green Kurta").
  + articleType – The product category (e.g., "Kurtas").
  + Other metadata such as gender, baseColour, season, etc.
  + Each product is also manually assigned a **unique label** (used for recommendation reference).

****

**2. product\_data.csv**

* This file is generated by linking product metadata to corresponding **image files**.
* Fields included:
  + product\_name
  + description
  + filename (e.g., 18.jpg)
  + filepath (complete path to the image)
  + label (unique index for product)
* Used during **recommendation generation and visualization**.

**Image Handling:**

* Product images are **downloaded and stored locally** in a directory named images/.
* The image filenames (e.g., 18.jpg) are **mapped** to the corresponding products using the label field.
* These images are later used to **visually display recommendations** inside the **Streamlit web app**.

**Why This Setup Was Used:**

* The subset enabled efficient processing during model development.
* A diverse but controlled dataset size improved recommendation testing and visualization.
* Separating metadata (final\_products.csv) and processed visual data (product\_data.csv) allows for **clear modularity** in the code.
* Storing images locally made it **easy to load and visualize recommendations in real-time** without additional downloads.

# 4. Libraries Used

• pandas – For data handling and CSV operations.

• numpy – For numerical computations.

• sklearn – Used for TF-IDF vectorization and cosine similarity computation.

• tensorflow/keras – Used for feature extraction from images using CNN (VGG16).

• PIL – To handle image loading and resizing.

• matplotlib – For image display.

• Streamlit – To build and deploy the web-based interface.

# 5. Code Explanation

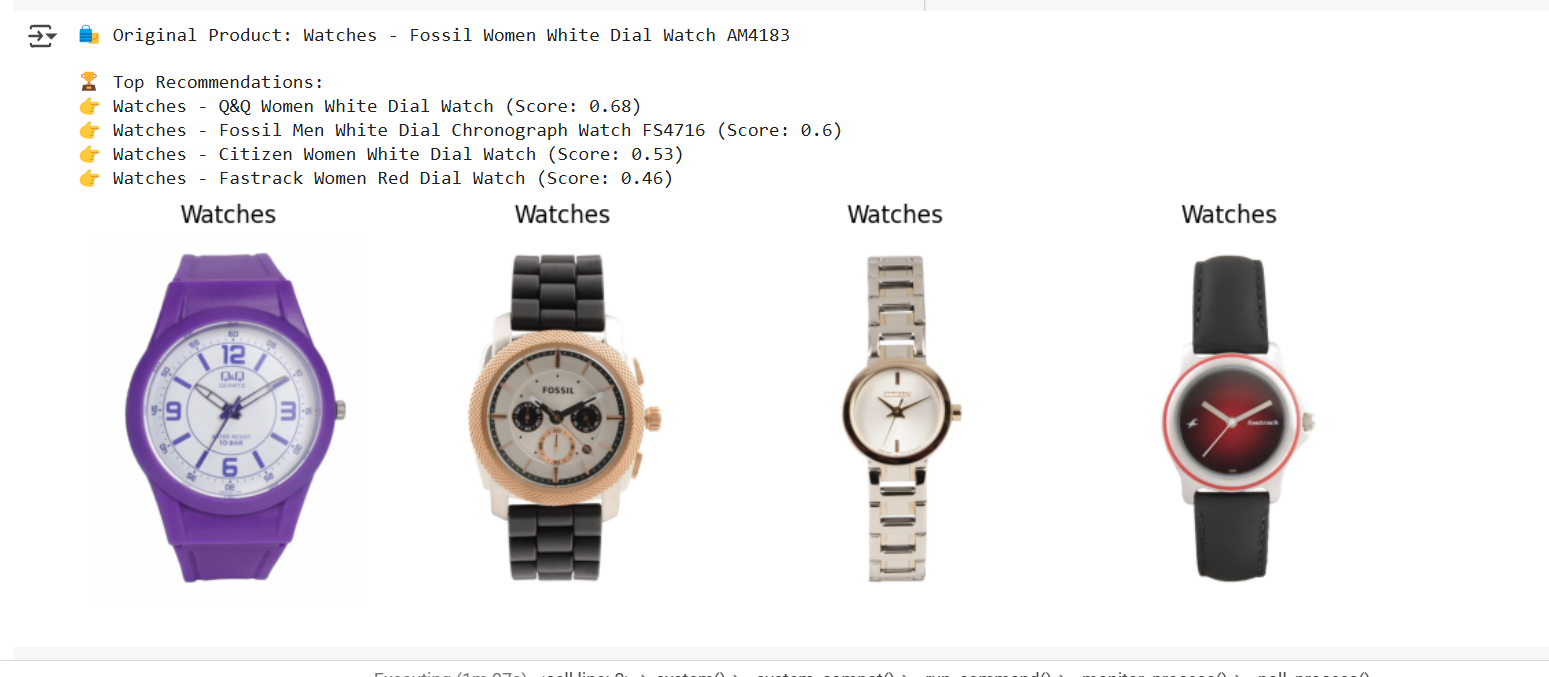
• Product Metadata Preparation – Product details like name, description, and labels were taken from a CSV file, and a mapping was created to match image file names with metadata.

• Feature Extraction – TF-IDF is applied on product descriptions to convert text into numerical vectors. Image features are extracted using pre-trained CNN (VGG16 model without the top layers).

• Cosine Similarity – Text and image vectors are merged, and cosine similarity is computed between products to find top recommendations.

• Recommendation Function – The `recommend\_by\_id` function finds the most similar products based on a given product label and displays the best matches.

• Image Display – A visual interface displays the recommended products and their images using Streamlit and Matplotlib.



# 6. Role of Data Science

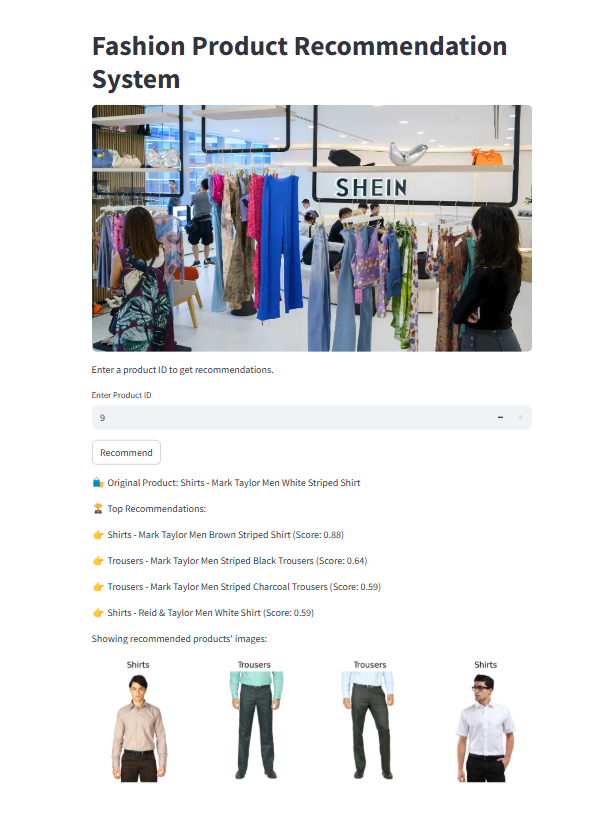
Data science plays a pivotal role in this project by enabling feature extraction, pattern recognition, and similarity computation from complex data (text and images). NLP helps in understanding user intent and product semantics, while image analysis through deep learning enhances visual similarity. Combining these allows us to build a highly intelligent recommender that mimics real-world e-commerce intelligence systems.

# 7. Applications

• E-Commerce Platforms – Recommend related products to boost cross-selling.

• Fashion Retailers – Suggest similar clothing items based on text and image.

• Personalized Shopping – Help users find relevant products visually and textually.

• Smart Inventory Search – Enable intelligent filtering and recommendations for retail managers.

# 8. Outputs and Deployment

• The Streamlit application accepts user input (e.g., product ID or image), computes recommendations, and displays top 3 similar products.

• The app is deployed using a tunnel-based approach (e.g., localtunnel or ngrok), making it accessible via a public URL for live demonstration.

# 9. Conclusion

This project demonstrates the effective use of data science techniques in building a real-world, AI-powered product recommendation system. By fusing NLP and CNN models, and delivering it through a web interface, the solution is both practical and scalable.

