**📑 CultureCircle Scraper & Product Similarity Project**

**🌐 Project Overview**

**Objective:**  
The CultureCircle project focuses on scraping fashion product data from [Culture Circle](https://culture-circle.com) and creating a pipeline to find visually similar products. This project lays the foundation for AI-driven applications, such as fashion recommendation systems or “lookalike” product matching.

**Scope:**

* Scrape products across **4 categories**: Shoes, Bags, Accessories, Clothing.
* Scrape for **3 genders**: Men, Women, Unisex.
* Extract **metadata, prices, and images**.
* Store **local copies of images**.
* Build **embeddings-based similarity search** using AI models (CLIP).

**📂 Tech Stack**

| **Component** | **Tool/Library** |
| --- | --- |
| Web Scraping | Selenium (Edge/Chrome) |
| HTTP Requests | Requests library |
| Data Handling | Pandas, JSON |
| Image Processing | PIL, OpenCV |
| Embeddings & Similarity | OpenAI CLIP, NumPy, scikit-learn (cosine similarity) |
| Logging & Progress | logging, rich |
| Visualization | Matplotlib |
| Version Control | Git |

**⚙️ Scraper Overview**

**Driver Setup**

* Supports **Edge** (preferred) or **Chrome**.
* Uses **headless mode** for faster scraping.
* Randomizes **User-Agent** to avoid detection.

**Utilities**

* Text cleaning and normalization for product names.
* Price extraction and numeric conversion.
* Generation of safe **local image paths**.
* Image download with retry mechanism.

**Infinite Scroll**

* Scrolls pages until all products are loaded.
* Handles **stable cycles** to detect when the page has no new items.

**Product Parsing**

* Extracts:
  + Product name, brand, category, gender.
  + Price and discounted price.
  + Price tier (affordable, mid, expensive).
  + Image URL and source platform metadata.

**Category & Keywords**

* 4 main categories: Shoes, Bags, Accessories, Clothing.
* Each category contains **keywords for Men, Women, and Unisex**.
* Keywords are used to query search results on the website.

**Workflow Summary**

* Iterates through categories, genders, and keywords.
* Loads search results page and scrolls to load all items.
* Parses product data and downloads images locally.
* Outputs are saved to **CSV** and **JSON** files for further analysis.

**💾 Output Fields**

* **product\_name**: Name of the product.
* **brand**: Brand of the product.
* **category**: Product category.
* **gender**: Gender classification (Men/Women/Unisex).
* **price**: Original price.
* **discounted\_price**: Discounted price (if any).
* **price\_tier**: Affordable, mid, or expensive.
* **combined\_tier**: Consolidated price tier.
* **image\_url**: Online image link.
* **local\_image\_path**: Path to downloaded image.
* **source\_platform**: Platform scraped from (Culture Circle).
* **product\_url**: Direct URL to the product page.

**🧬 Product Similarity Analysis**

**Objective:**  
To find visually similar products using image embeddings generated from the CLIP model. This allows for:

* Identifying “lookalike” products.
* Pairing expensive vs affordable items.
* Building a recommendation engine.

**Workflow**

1. **Load Scraper Output:**
   * CSV with product metadata and precomputed embeddings.
2. **Embedding Extraction:**
   * Use OpenAI **CLIP** to compute embeddings for query images.
   * Normalize embeddings to ensure consistent similarity computation.
3. **Similarity Computation:**
   * Compute **cosine similarity** between the query image and all product embeddings.
   * Retrieve **top N most similar products**.
4. **Visualization & Review:**
   * Display the query image alongside the top similar products.
   * Include product details like brand, price, category, gender, and URLs.
   * Enables qualitative assessment of similarity results.

**Benefits of Similarity Notebook**

* Fully **interactive** using Jupyter Notebook.
* Supports **both local CSV embeddings** and **new query images**.
* Can scale to large datasets with **efficient similarity search**.
* Lays the foundation for **AI-driven recommendation systems** or gamified applications like “LIT Game”.
* Enables **visual validation** of embedding quality.

**📊 Project Results**

* Scraped hundreds of products across all categories and genders.
* Local images stored in structured folders: /images/<category>/<gender>/.
* Embeddings stored in CSV allow for **quick and scalable similarity search**.
* Similarity notebook successfully identifies top visually similar products for any query image.

**🚀 Future Improvements**

* Integrate **affiliate links** and price-tracking for product recommendations.
* Use **Faiss** or other vector databases for faster similarity queries.
* Implement **material/color features** in similarity matching.
* Automate expensive vs affordable product pairing for applications like “LIT Game”.
* Expand categories and keywords for broader coverage.

**✅ Deliverables**

1. **Scraper Outputs**
   * CSV and JSON of all scraped products.
   * Downloaded product images in structured folders.
2. **Similarity Search Notebook**
   * End-to-end documentation of embedding generation.
   * Top visually similar product searches with images and metadata.
3. **Logs**
   * Detailed logging of scraping sessions for monitoring and debugging.
4. **Documentation**
   * Project summary and workflow explanation (this document).