**Project Documentation: Fashion Product Search System**

**Introduction**

This project aims to create a system that enables users to search for fashion products based on either textual queries or image uploads. The system utilizes machine learning techniques, including deep learning for image embeddings and natural language processing (NLP) for text embeddings, to match user inputs with fashion items from a dataset. The project was implemented in Python 3.11, and TensorFlow served as the backbone for feature extraction.

**Approach**

**1. Data Preprocessing**

**Steps:**

* Loaded the dataset (styles.csv) containing metadata about fashion products.
* Sorted the dataset by the id column and extracted the first 2000 rows for processing.
* Created a new folder for the first 2000 images, copying the corresponding images from the original dataset folder.
* Saved a new CSV file (new\_style.csv) containing metadata for the extracted images.

**Code Reference:** 1st step.py

**2. Feature Extraction**

**Image Embeddings:**

* Loaded images and resized them to 224x224 pixels to align with ResNet50's requirements.
* Used the pre-trained ResNet50 model to extract image embeddings, focusing on the average-pooled feature representations.

**Text Embeddings:**

* Combined relevant textual metadata such as gender, masterCategory, subCategory, etc., into a single textual representation for each product.
* Used the TF-IDF vectorizer to convert the textual data into numerical embeddings.

**Normalization:**

* Normalized both image and text embeddings to improve similarity computations.

**Code Reference:** 2nd step.py

**3. Similarity Computation**

**Dimensionality Reduction:**

* Used PCA to reduce the dimensionality of image embeddings to match the size of text embeddings.

**Similarity Functions:**

* Used cosine similarity to compute the similarity between a query (image or text) and all stored embeddings.
* Implemented functions for:
  + Finding similar images based on a textual query.
  + Finding similar images based on an image query.

**Code Reference:** 3rd step.py

**4. Deployment**

**User Interface:**

* Built an interactive web application using Streamlit.
* Provided options for users to search by:
  + Text query: Users enter a descriptive text (e.g., "blue shirt").
  + Image upload: Users upload an image to find similar items.

**Workflow:**

* For text queries:
  + Extracted embeddings using the pre-trained TF-IDF vectorizer.
  + Computed similarities and displayed the top-matching images.
* For image uploads:
  + Processed the uploaded image using ResNet50.
  + Applied PCA and normalized the embedding.
  + Computed similarities and displayed the top-matching images.

**Code Reference:** app.py

**Challenges and Limitations**

1. **Deployment Issues:**
   * Faced version conflicts between Python (3.11) and TensorFlow during deployment on platforms like Render and Streamlit.
   * These conflicts prevented successful deployment despite extensive efforts.
2. **Knowledge Limitations:**
   * As this project was highly advanced, it required learning new topics such as deep learning, image processing, and NLP.
   * Relied heavily on online resources, including Google, YouTube, and documentation, to understand and implement various components.
3. **Project Delays:**
   * Delays were primarily due to the steep learning curve and troubleshooting deployment-related issues.

**Conclusion**

This project was a high-end endeavor, and I gave my best effort to bring it as close to completion as possible. Despite deployment challenges, I was able to develop a robust system for fashion product search, showcasing the integration of machine learning techniques in real-world applications.

The project reflects my dedication to learning and implementing advanced concepts, and I am proud of the progress made. Future iterations could focus on resolving deployment issues and refining the system further.