Find me a better product!

Jenson Chang, Chukwunonso Ebele-Muolokwu, Catherine Meng, Jingyuan Wang

Table of contents

Executive Summary

FinlyWealth is an affiliate marketing platform that rewards customers for applying for financial products. It is now looking to expand its business by offering e-commerce products through its platform. To support this transition, a team of Master of Data Science students from the University of British Columbia is developing a fast and scalable multimodal search engine that allows users to search using text, images, or both, to find the most relevant products. The final product delivers advanced product discovery by leveraging semantic understanding, enabling more accurate and relevant search results beyond simple keyword matching.

Our hybrid retrieval strategy combines CLIP-based (Radford et al. 2021) and MiniLM (Face 2024) embeddings with FAISS indexing (Johnson, Douze, and Jégou 2017) for efficient large-scale similarity search. After retrieval, we leverage an LLM-based reranking module to reorder the candidates, ensuring that the most relevant products are ranked at the top. The system consists of a Streamlit frontend (Streamlit Inc. 2019), a Flask API backend (Ronacher 2010), and a vector store for embedding-based retrieval. We evaluate the system's performance using Recall@20, Precision@20 based on human relevance judgments, and query latency. Our results show a Recall@20 of 0.56, Precision@20 of 0.64, and an average search time of 4.24 seconds over a dataset of one million products.

Introduction

As FinlyWealth expands its offerings from personal finance into the e-commerce sector, it faces the challenge of delivering a scalable and effective product search experience across a rapidly growing and diverse catalog. To address this, a team of Master of Data Science students at the University of British Columbia is developing a machine learning-powered multimodal search engine that understands the semantic meaning of user queries, handling both text and image inputs to help users find relevant products more intuitively and efficiently.

Search in the e-commerce domain presents unique challenges due to the wide variety of ways users express their search intent. Traditional approaches, such as TF-IDF-based text search, work well for simple queries like "iPhone" or "laptop." However, most user queries are free-form, complex, and infrequent. The existing system relies on basic keyword matching, lacking semantic understanding, support for multimodal inputs, and large-scale performance evaluation.

Objective

To address these gaps, this project designed and implemented a fast, scalable multimodal search system that captures semantic meaning of user queries and returns the most relevant products to the users. Architecture components include:

- **Preprocess Script**: Python scripts runnable via make commands to generate text and image embeddings from raw CSV and image data, and to build FAISS indexes
- **Frontend**: Streamlit for handling interactive text and image queries, and displaying search results along with summary statistics and response time (Streamlit Inc. 2019)
- **Backend API**: Flask for query handling and results retrieving (Ronacher 2010)

  ```
  - **Similarity Engine**: FAISS for approximate nearest neighbor search [@faiss]

  - **Post Retrieval Reranking**: GPT-3.5-turbo LLM for reranking the top 30 retrieved car
  ```

- **Vector Store**: Google Cloud PostgreSQL for affordable and scalable storage of embeddings and metadata

The final data product is evaluated using the following success metrics:

- Recall@20 for retrieval accuracy
- Latency for query responsiveness (target: under 5 seconds)
- Precision@20 based on manual relevance assessments for qualitative validation

Data Science Techniques

Data Source, Description and Cleaning

The dataset consists of multimodal product data, including images (14,684,588 JPEG files, approximately 67 GB), textual information (product names and descriptions), and structured metadata (e.g., `Category`, `Brand`, `Color`). The metadata is stored in a 12 GB CSV file containing 15,384,100 rows and 30 columns.

After conducting exploratory data analysis and consulting with our partner, we selected the 16 most relevant columns that capture the key information users care about. We excluded non-English market entries—retaining approximately 70% of the dataset—in line with our partner's business focus. Additionally, we merged the `Brand` and `Manufacturer` columns into a single `MergedBrand` field to reduce duplication while preserving distinct brand information. We chose to ignore missing values in the metadata columns, as these fields are likely to provide supplementary information, while the product name already contains the primary details.

Table 1: Table: Summary of Retained Columns and Their Characteristics

Group	Attribute	Description / Examples
Identifiers	`Pid`	Unique product ID; links to image filenames
Text Fields	`Name`	Product title (0.2% missing)
	`Description`	Product description (0.03% missing)
	`Category`	Product category (28% missing; ~15 K unique values)
Pricing & Availability	`Price`	Listed price
	`"PriceCurrency"`	Currency of the price
	`FinalPrice`	Final price after discounts
	`Discount`	Discount percentage or value
	`isOnSale`	Boolean flag
	`IsInStock`	Boolean flag
Branding	`Brand`	Brand name (53% missing; ~21 K unique values)
	`Manufacturer`	Manufacturer name (34% missing; ~26 K unique values)
Product Features	`Color`	Product color (49% missing; ~170 K unique values)
	`Gender`	Target gender (54% missing; 3 values: e.g., male/female)
	`Size`	Product size (46% missing; ~55 K unique values)
	`Condition`	Product condition (e.g., new, used; 5 values)

Data Science Techniques

Our goal was to build a multimodal search engine that returns relevant product results in response to diverse customer queries. To achieve this, we focused on combining text and image understanding with scalable retrieval techniques. We designed a hybrid retrieval system that combines full text search, multimodal embeddings and LLM reranking to improve the relevance of product search results. This pipeline integrates TF-IDF (Term Frequency-Inverse Document Frequency) for full text search, CLIP for image-text understanding, MiniLM for lightweight semantic understanding, FAISS (Facebook AI Similarity Search) for fast retrieval, and OpenAI GPT3.5 LLM (Large Language Model) for reranking.

TF-IDF

TF-IDF is a keyword-based search method that ranks products based on how uniquely their descriptions match the search query.

CLIP and MiniLM Embedding

We were inspired by (Liu and Lopez Ramos 2025), who combined CLIP and a (Devlin et al. 2019) model fine-tuned on e-commerce data to improve product search. Since we didn't have access to labeled domain-specific data for fine-tuning, we chose a smaller, faster transformer model that performs well out-of-the-box that supports semantic understanding.

(Radford et al. 2021) (Face 2024)

FAISS

FAISS is a library for efficient similarity search and clustering of dense vectors. It enables fast retrieval from large-scale embedding databases using indexing techniques like IVF (Inverted File Indexing). A preprocessing step is required to build this index so it can be searched through when a user submits a query.

Large Language Model

A LLM is used to help improve search quality by re-ranking initial product results based on deeper semantic understanding. It interprets the user's intent and assesses the relevance of each product based on the following:

1. Semantic similarity to the query intent
2. Direct keyword matches
3. Brand Name mentions
4. Price comparison

Preprocessing Pipeline

Search Pipeline

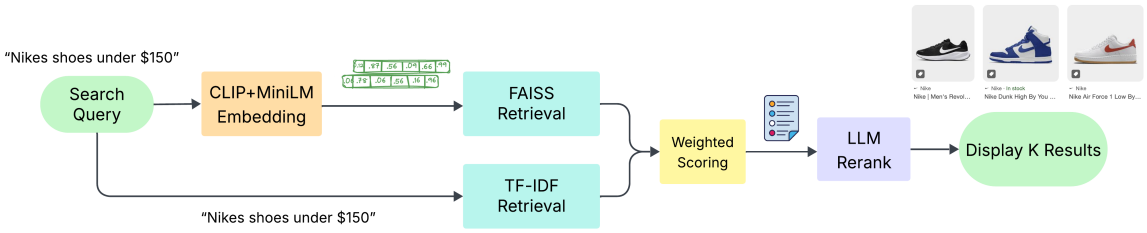

Figure 1: search-pipeline

Evaluation

Data Product Results

Appendix

Tools and Libraries

Library	Purpose in Project
NumPy	Efficient numerical operations, especially for vector manipulation and math ops.
Flask	Lightweight web framework used for rapid prototyping of API endpoints.
FAISS	Approximate nearest neighbor search for CLIP embeddings; enables fast vector search.
Hugging Face	Access to pretrained models like CLIP; used for text and image embedding.
Pillow	Image processing library used for resizing, normalization, and format conversion.
spaCy	Natural language processing toolkit for tokenization, NER, and text normalization.
Pinecone	Scalable, cloud-based vector database for fast and persistent similarity search.
PostgreSQL	Relational database to store Embeddings. Allows for multiple columns to have ebeddings

Definitions

CLIP: Generates embeddings for both text and images, mapping them into a shared embedding space. We are not training any embedding model, instead we use off-the-shelf CLIP models to generate embeddings.

Embedding Generation: The preprocessed query is then transformed into a numerical representation (an embedding) that captures its semantic meaning.

FAISS (Facebook AI Similarity Search) is a library that allows developers to quickly search for embeddings of multimedia documents.Enables efficient approximate nearest neighbor search over embeddings.

TF-IDF: A numerical statistic used to evaluate the importance of a word in a document within a collection of documents

References

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. "BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding." In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, 4171–86. Minneapolis, MN, USA: Association for Computational Linguistics. https://doi.org/10.18653/v1/N19-1423.

Face, Hugging. 2024. "MiniLM on Hugging Face." https://huggingface.co/models.

Johnson, Jeff, Matthijs Douze, and Hervé Jégou. 2017. "FAISS: A Library for Efficient Similarity Search and Clustering of Dense Vectors." https://github.com/facebookresearch/faiss.

Liu, Dong, and Esther Lopez Ramos. 2025. "Multimodal Semantic Retrieval for Product Search." *arXiv Preprint arXiv:2501.07365*, January. https://doi.org/10.48550/arXiv.2501.07365.

Radford, Alec, Jong Wook Kim, Luke Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, et al. 2021. "Learning Transferable Visual Models from Natural Language Supervision." *Proceedings of the International Conference on Machine Learning (ICML)*. https://github.com/openai/CLIP.

Ronacher, Armin. 2010. "Flask: Web Development, One Drop at a Time." https://flask.palletsprojects.com/.

Streamlit Inc. 2019. "Streamlit." https://streamlit.io.