

Prakhyat Singh

RE-2022-545341



Universidad del Valle

Document Details

Submission ID

trn:oid:::26066:450801328

Submission Date

Apr 21, 2025, 3:51 AM GMT+5:30

Download Date

Apr 21, 2025, 3:54 AM GMT+5:30

File Name

RE-2022-545341 (1).docx

File Size

623.9 KB

7 Pages**3,961 Words****23,257 Characters**





7% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.




Filtered from the Report

- Bibliography
- Quoted Text

Match Groups

-  **28 Not Cited or Quoted 6%**
Matches with neither in-text citation nor quotation marks
-  **4 Missing Quotations 1%**
Matches that are still very similar to source material
-  **0 Missing Citation 0%**
Matches that have quotation marks, but no in-text citation
-  **0 Cited and Quoted 0%**
Matches with in-text citation present, but no quotation marks

Top Sources

- 3%  Internet sources
- 2%  Publications
- 6%  Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

No suspicious text manipulations found.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

Match Groups

- 28 Not Cited or Quoted 6%**
Matches with neither in-text citation nor quotation marks
- 4 Missing Quotations 1%**
Matches that are still very similar to source material
- 0 Missing Citation 0%**
Matches that have quotation marks, but no in-text citation
- 0 Cited and Quoted 0%**
Matches with in-text citation present, but no quotation marks

Top Sources

- 3% Internet sources
- 2% Publications
- 6% Submitted works (Student Papers)

Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	Internet	www.mdpi.com	<1%
2	Internet	arxiv.org	<1%
3	Submitted works	University of Stellenbosch, South Africa on 2024-08-31	<1%
4	Submitted works	Heriot-Watt University on 2025-03-27	<1%
5	Submitted works	HTM (Haridus- ja Teadusministeerium) on 2023-12-25	<1%
6	Submitted works	Technological University Dublin on 2025-01-19	<1%
7	Publication	P.V. Mohanan. "Artificial Intelligence and Biological Sciences", CRC Press, 2025	<1%
8	Submitted works	University of Adelaide on 2023-09-14	<1%
9	Submitted works	Hong Kong University of Science and Technology on 2024-11-13	<1%
10	Submitted works	Glasgow Caledonian University on 2023-04-21	<1%

11	Submitted works	Livermore High School on 2025-04-16	<1%
12	Submitted works	Brunel University on 2025-04-19	<1%
13	Submitted works	City University on 2023-10-01	<1%
14	Submitted works	Imperial College of Science, Technology and Medicine on 2019-09-11	<1%
15	Submitted works	Instituto de Empress S.L. on 2025-04-10	<1%
16	Internet	aclanthology.org	<1%
17	Publication	"Computer Vision – ECCV 2018 Workshops", Springer Science and Business Media ...	<1%
18	Internet	iis-international.org	<1%
19	Internet	proceedings.mlr.press	<1%
20	Internet	www.naturalspublishing.com	<1%
21	Submitted works	Berlin School of Business and Innovation on 2025-02-08	<1%
22	Submitted works	Liverpool John Moores University on 2024-03-13	<1%

Image-Driven Fashion Recommendations: Clustering and Similarity with CLIP and HDBSCAN

Prakhyat Singh
12218463
prakhyat@lpu.in

Yunesh Kumar
12217704
yunesh@lpu.in

Kirti
29458
kirti.29458@lpu.co.in

Abstract— This project focuses on building a visual-based fashion recommendation system using three input images of men in formal attire. Instead of relying on text or user ratings, we use image-based inputs to capture the user's style preferences. We use CLIP, a deep learning model that converts images into embeddings, to extract visual features from both the input and dataset images. These embeddings are then compared using cosine similarity to find the closest matches in style. To further group similar fashion items, we apply HDBSCAN, an unsupervised clustering algorithm that doesn't require a fixed number of clusters. Our results show that this method accurately recommends fashion items that match the user's formal style, and that HDBSCAN helps organize the data effectively. This paper explains the models, the approach, and the results in a simple and clear way.

I. INTRODUCTION

In today's world, fashion recommendation systems have become an essential part of online retail platforms. With millions of products to choose from, users often find it overwhelming to manually browse for clothing items that match their personal style. To solve this problem, recommendation systems leverage techniques from machine learning and computer vision to analyze user preferences and suggest relevant products automatically.

Traditionally, these systems have relied heavily on structured metadata such as product descriptions, tags, or user ratings. However, such approaches may fall short when product metadata is inconsistent, missing, or overly generic. Moreover, they cannot capture the true "visual style" or aesthetic of clothing items. This is where visual-based recommendation systems come in—offering a more accurate and intuitive way to suggest fashion products by analyzing actual images.

In this project, we developed a visual recommendation engine that uses **CLIP (Contrastive Language–Image Pretraining)** for extracting deep image embeddings from fashion product images, and then uses **HDBSCAN (Hierarchical Density-Based Spatial Clustering of Applications with Noise)** to group similar items based on visual similarity. The goal was to recommend clothing items that matched the user's visual style, as expressed through three reference images—each showing a man dressed formally in a shirt and pants.

The user's outfit preferences were captured by embedding the input images using CLIP and averaging their representations to form a combined "style vector." This vector was then compared against a dataset of catalog embeddings to retrieve visually

similar items. While traditional cosine similarity alone gave us a quick way to find the closest matches, clustering using HDBSCAN offered a more structured approach by grouping similar items into clusters, allowing for more meaningful and diverse recommendations from the same visual style.

One of the key strengths of this project lies in using unsupervised learning. Unlike supervised methods that require labeled training data, our approach requires no labels or manual annotations. This not only reduces the need for curation but also allows the system to adapt to changing fashion trends and styles.

The dataset used was derived from the publicly available Fashion Product Images Dataset on Kaggle, which includes tens of thousands of clothing product images along with metadata. However, this project intentionally focused on a visual-only approach to better explore the effectiveness of image-based techniques.

In summary, this paper explores how combining CLIP-based embeddings with HDBSCAN clustering can help build an efficient, personalized, and visually coherent fashion recommendation engine. We also compare the performance of clustering-based recommendations with those generated using cosine similarity, highlighting the practical trade-offs between the two methods.

Fig 1. Visual Workflow of the Fashion Recommendation System.

II. LITERATURE REVIEW

The field of computer vision has experienced a major leap in recent years due to the availability of large datasets and powerful neural architectures. These advances have made it possible to build systems that not only identify objects in images but also understand high-level features such as aesthetics, texture, and even *style*. In the domain of fashion recommendation systems, this shift has been especially impactful, since style is subjective, subtle, and often cannot be captured by traditional algorithms.

A. Evolution of Fashion Recommendation Systems

Traditional fashion recommendation engines primarily relied on collaborative filtering and content-based filtering techniques [1]. These systems used structured metadata like color, brand, or user ratings to suggest similar products. However, such approaches struggled when dealing with new products (cold start problem) or when visual similarity was more important than user behavior.

Later, methods incorporated low-level visual features using computer vision techniques like SIFT, HOG, or color histograms [2], but they still fell short of capturing the *semantics* of fashion—like whether two outfits "look formal" or share a similar vibe. This led to the adoption of deep learning-based models, which could learn features directly from the raw pixels of an image. Convolutional Neural Networks (CNNs), like ResNet and Inception, brought significant improvements by enabling better generalization of fashion styles, patterns, and silhouettes [3].

B. CLIP and Vision-Language Models

A major milestone in this evolution was the introduction of CLIP (Contrastive Language–Image Pretraining) by OpenAI [4]. CLIP learns a joint representation of images and text using a contrastive loss, aligning semantically similar pairs while pushing apart unrelated ones. It was trained on over 400 million (image, text) pairs from the internet, allowing it to understand broad and nuanced concepts—even ones not explicitly present during training.

In this project, CLIP is used to extract embeddings from fashion product images as well as user-uploaded images. Since CLIP produces a high-dimensional vector that encapsulates the content and style of an image, it serves as the backbone for both similarity ranking and clustering. Unlike CNNs trained for classification, CLIP's embeddings are *general-purpose* and *style-sensitive*, making them ideal for use in fashion recommendation engines.

Why CLIP was used:

- It captures both *semantic* and *visual* features.
- It supports zero-shot learning (no retraining needed on new datasets).
- It performs well on diverse visual categories without task-specific fine-tuning..

C. Dimensionality Reduction using UMAP

Once we have high-dimensional embeddings (typically 512 dimensions in CLIP), it becomes computationally expensive to cluster or visualize them directly. UMAP (Uniform Manifold Approximation and Projection) is a state-of-the-art dimensionality reduction technique that is especially good at preserving the *local and global structure* of data [5].

UMAP works by constructing a graph of nearest neighbors in the high-dimensional space, then optimizes a low-dimensional representation that maintains the topological relationships between points. In this project, UMAP is used to reduce the dimensionality of the catalog embeddings before applying clustering.

Why UMAP was used:

- It preserves both global and local relationships better than t-SNE.
- It is faster and scales better with larger datasets.
- It creates a space where similar items are closer, helping clustering perform better.

D. Clustering with HDBSCAN

After reducing the image embeddings using UMAP, the next step is to group similar items together. While algorithms like KMeans or DBSCAN are commonly used for clustering, they have limitations: KMeans assumes spherical clusters and requires knowing the number of clusters beforehand, and DBSCAN struggles with variable density data.

HDBSCAN (Hierarchical Density-Based Spatial Clustering of Applications with Noise) is a more flexible and robust alternative [6]. It builds a hierarchy of clusters and selects the most stable ones, automatically identifying noise (outliers) and handling clusters of varying density. In the context of fashion data—which is naturally noisy, diverse, and hard to categorize strictly—HDBSCAN is a natural fit.

Why HDBSCAN was used:

- It doesn't require specifying the number of clusters.
- It handles variable-density clusters, which is common in fashion datasets.
- It identifies and filters out noise/outliers effectively.

E. Cosine Similarity for Style Matching

For recommending products similar to the user's style, we compute cosine similarity between the average embedding of the user's images and every item in the catalog. Cosine similarity measures the angle between vectors and is ideal for high-dimensional spaces because it's not affected by the magnitude of the vectors—just their orientation [7].

In practice, this means two outfits that "look similar" will have embeddings pointing in similar directions, even if they come from different distributions.

Why Cosine Similarity was used:

- It's a standard and efficient method for comparing embeddings.
- It is scale-invariant and works well in high-dimensional spaces.
- It aligns with how CLIP embeddings are structured.

III. METHODOLOGY

This section presents a complete breakdown of how the recommendation system was implemented, from understanding the user input to generating contextually rich outfit suggestions using advanced models and clustering techniques. The core idea was to analyze three user-uploaded fashion images and return a coherent set of clothing items (top wear, bottom wear, accessories, and footwear) that align with the user's apparent fashion sense.

A. Input Data and Feature Extraction

The project begins by taking three user-provided images as input. These images depict three different men wearing formal shirts and pants, and the intention is to understand the user's clothing preferences and then recommend similar fashion products. These input images serve as a stylistic reference or a "style fingerprint" that guides the entire recommendation pipeline.

To extract meaningful visual features from these input images, the project uses OpenAI's CLIP model. Specifically, the version employed is CLIP ViT-B/32, which is a Vision Transformer-based architecture pretrained on a massive dataset consisting of image-text pairs. The CLIP model is used here in image encoding mode, which means it only processes the images to generate a fixed-length feature vector (embedding) for each image.

Each input image is first preprocessed (resized, normalized, and converted to a tensor) before being passed to the CLIP model. Once encoded, CLIP outputs a 512-dimensional vector representing the visual content and semantic style of the image. To create a more robust and representative embedding of the user's overall style, the project averages the three individual vectors into a single composite user embedding. This average vector represents the "center" of the user's fashion style in high-dimensional feature space.

This single embedding then becomes the anchor for the recommendation system, which will compare it to a large dataset of catalog embeddings and cluster them for better organization.

B. Catalog Embedding and Comparison

Once the user's style is captured in a single representative vector, the next step is to compare this embedding against a larger fashion product catalog to identify similar items.

i. Dataset Description

The catalog used in this project comes from the Fashion Product Images Dataset sourced from Kaggle. It contains over 44,000 labeled fashion product images, including tops, pants, dresses, shoes, and accessories, each associated with metadata like product ID, gender, and category. However, for this project, the focus is entirely on the visual content, not the metadata.

Only images that were valid (non-corrupted and successfully readable) were used. The product image paths are linked to their corresponding IDs in the dataset's CSV file. These were cleaned and filtered to ensure robustness before processing.

ii. Catalog Feature Extraction with CLIP

Similar to the user images, each catalog image is passed through the same CLIP ViT-B/32 model to obtain its 512-dimensional image embedding. To maintain consistency, preprocessing steps identical to those used for the user inputs (resizing, normalizing, and RGB conversion) are applied.

These embeddings are computed once and saved as .npy files (NumPy arrays) for faster access and reuse. This avoids redundant computation during repeated runs.

Now, every product in the catalog is represented as a point in the same high-dimensional space as the user style vector.

iii. Cosine Similarity: Measuring Visual Closeness

To compare the user's composite style embedding with every product embedding in the catalog, the project uses cosine similarity. Cosine similarity measures the angle between two vectors — a high score (close to 1) indicates that the two vectors point in a similar direction, which implies visual and stylistic similarity in this context.

$$\cos(\theta) = \frac{A \cdot B}{\|A\| \|B\|}$$

The similarity score is calculated between the user's vector and every catalog vector, and the top-N (e.g., 10) most similar products are selected as the first set of recommendations.

C. Dimensionality Reduction and Clustering

While cosine similarity provides a direct way to compare embeddings, clustering helps group visually similar items, which can enhance recommendation diversity and personalization. To achieve this, we use two important techniques: UMAP for dimensionality reduction, and HDBSCAN for clustering.

i. UMAP: Making High-Dimensional Data Usable

UMAP (Uniform Manifold Approximation and Projection) is a powerful tool for reducing high-dimensional data to lower dimensions, typically 2D or 3D for visualization — or more in our case (15D) for better structure retention while still reducing computational cost.

The embeddings from CLIP are 512-dimensional, which is too large for most clustering algorithms to work efficiently or meaningfully. UMAP helps by preserving both the local structure (similar points stay close together) and global structure (clusters and gaps remain visible) of the data.

In this project, we reduce the catalog embeddings from 512D → 15D using UMAP with these key hyperparameters:

- **n_neighbors=15**: balances between local and global structure
- **min_dist=0.1**: controls how tightly UMAP packs points together
- **metric='cosine'**: ensures distances are based on similarity of direction, consistent with previous steps

ii. HDBSCAN: Smart Clustering That Can Ignore Noise

After dimensionality reduction, we apply HDBSCAN (Hierarchical Density-Based Spatial Clustering of Applications with Noise) to group similar catalog items.

Unlike traditional clustering algorithms like k-means, which assume spherical clusters and require specifying the number of clusters (k) ahead of time, HDBSCAN:

- Does not require k
- Can discover clusters of varying shapes and sizes
- Can label outliers as noise (label -1), which is helpful to filter unclusterable or ambiguous items

HDBSCAN builds a hierarchy of clusters and then condenses it based on the stability of those clusters — only keeping the most meaningful ones.

In this project, HDBSCAN is applied to the UMAP-reduced catalog embeddings with:

- **min_cluster_size=30**: minimum number of samples to form a valid cluster
- **metric='euclidean'**: distance metric used for clustering

iii. Visualizing Cluster Distribution

A histogram of the HDBSCAN results shows the distribution of items across different clusters. Each bar represents a cluster and the number of items it contains. This gives insight into:

- Which clusters are dense (popular styles)
- How much of the dataset is treated as noise
- Whether the clustering is balanced or skewed

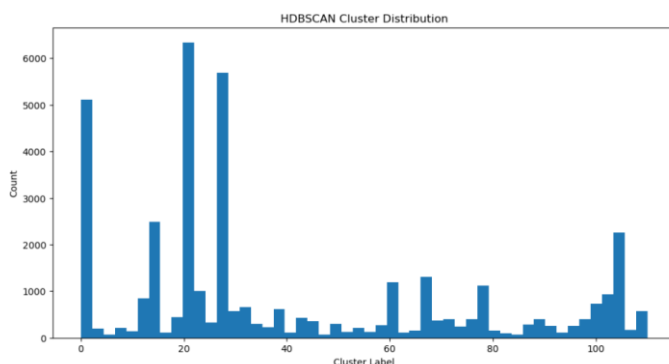


Fig 2. HDBSCAN Cluster Distribution of Fashion Items.

iv. Using Clusters for Recommendation

After clustering:

- We pass the user's averaged embedding through the same UMAP transformation.
- We compare it to each cluster's centroid (mean of all embeddings in that cluster).
- We find the nearest cluster to the user's style.
- We recommend top items from that cluster, using cosine similarity for fine-grained ranking.

This hybrid approach combines global structure (cluster-based filtering) with local similarity (cosine matching), leading to more coherent and style-consistent recommendations.

IV. RESULTS

This section presents and interprets the output generated by the system across two different recommendation strategies: (1) direct cosine similarity, and (2) the HDBSCAN-based clustering method. Both approaches use the same user style input derived from three user-uploaded images of men dressed in formal attire. These images represent the user's fashion preference and guide the recommendation system to generate relevant outputs.

A. Cosine Similarity Recommendations

In the first approach, we calculate cosine similarity between the user embedding (mean of the three input images) and every item in the catalog. Cosine similarity gives us a score that tells us how close in style each catalog item is to the user's style vector. The top 10 items with the highest similarity scores are selected and displayed.

Observations from the output:

- The top results are visibly formal, aligning well with the input.
- Items mostly include shirts and trousers, with subtle stylistic variation.
- Some recommended items are near duplicates or highly similar to each other in colour and cut.

Pros:

- Very direct and accurate matching.
- Fast to compute.
- Excellent when the user's style is very specific.

Cons:

- Lacks diversity – tends to return almost identical items.
- Doesn't understand higher-level grouping or styles beyond embedding closeness.



Fig 3. Cosine Similarity

B. HDBSCAN-based Cluster Recommendations

In the second approach, after clustering the catalog using UMAP + HDBSCAN, we locate the closest cluster to the user's style. We then only consider items within that cluster and re-rank them using cosine similarity to surface the top 10 recommendations.

Observations from the output:

- The recommended items still clearly reflect formal styles but are more varied in design.
- Color range is slightly broader, yet the core formality is retained.
- Items appear more representative of a style group rather than being clones of a single look.

Pros:

- Captures the underlying "style" cluster, leading to more well-rounded recommendations.
- Introduces more diversity while still staying true to the user's taste.
- Helps avoid overfitting to small details in the input images.

Cons:

- Slightly more complex to implement and computationally heavier.
- Some less-relevant items may occasionally appear, especially if clusters are broad.



Fig 3. HDBSCAN Similarity

C. Summary of Output Quality

The final recommendations in both approaches reflect the user's formal fashion preference effectively, but with different strengths:

- **Cosine similarity** excels at **precision**, making it ideal for users who want exact matches or fine-tuned recommendations.
- **HDBSCAN-based clustering** improves **style coherence** and **diversity**, offering a more exploratory experience.

Together, they show how combining visual understanding with structured clustering can create a balanced and intelligent recommendation system.

Feature	Cosine Similarity	HDBSCAN
Precision	High	Moderate-High
Diversity	Low	High
Handles Ambiguity	No(Rigid Matching)	Yes(Cluster context)
Interpretability	Hard to reason	Style based groupings
Noise Filtering	None	Via Outlier removal

Table 1. Comparison Table between Cosine and HDBSCAN

V. DISCUSSION

This project aimed to create a personalized fashion recommendation system that could take a few input images and return clothing items that reflect a user's style. Based on the output and results, it's clear that both cosine similarity and cluster-based recommendation pipelines achieved this goal, each in their own way. But let's dig a little deeper.

A. Understanding the Output in Context

The user's input images—three men dressed in formal shirts and trousers—were not explicitly labeled or described to the model. Still, the system could "understand" their visual style using CLIP embeddings, and this understanding translated well into the results. This highlights the strength of CLIP as a visual encoder trained on broad vision-language data. Even without explicit fashion tags, the model could pick up on what made these images formal—collared shirts, muted colors, structured silhouettes—and find similar aesthetics in the catalog.

B. Cosine Similarity vs. HDBSCAN: A Tradeoff

Both methods have clear use cases:

- If a user has a very specific idea of what they want (say, more white shirts that look like the one in their input), cosine similarity shines. It's fast, simple, and laser-focused on direct similarity.
- If a user is more into exploring a style space, like discovering other types of formal wear, then the HDBSCAN-based method does better. By working within

a style cluster, it offers more variety while still respecting the core look the user is after.

This diversity vs. precision tradeoff is common in recommendation systems. It's also a space where user interfaces can help—perhaps by letting users toggle between “similar” and “diverse” modes based on their mood or needs.

C. Strengths of This System

- **Zero-shot capability:** No need to train the CLIP model from scratch. It already understands image features, thanks to its pretraining.
- **Unsupervised clustering:** HDBSCAN doesn't need labeled data. It discovers style groups on its own.
- **Scalability:** With precomputed embeddings, both cosine and clustering methods are fast and scalable to large catalogs.
- **Interpretability:** Because we can visualize clusters, and because recommendations are image-based, the user can easily understand why certain items were suggested.

D. Limitations and Areas for Improvement

- **Input image quality:** If user-provided images are blurry, low-resolution, or contain distracting backgrounds, embeddings might not be accurate.
- **Catalog bias:** If the catalog is heavily skewed toward one style or color palette (e.g., mostly Western menswear), the recommendations can become narrow.
- **No personalization beyond visuals:** Right now, the system looks only at visual style. It doesn't account for user preferences like price, size, brand loyalty, or cultural factors.

Future work could include:

- Allowing users to provide textual inputs along with images (e.g., “Formal but trendy,” “Avoid black shirts”).
- Combining this system with filtering layers based on metadata like price or brand.
- Training lightweight style classifiers on top of clusters to give names like “business casual,” “smart formal,” etc.

E. Broader Implications

What's exciting is that the system doesn't just “recommend clothes.” It captures personal style in a way that feels intuitive and human. It reduces decision fatigue by narrowing down hundreds of products to a few that truly match a person's taste—without needing that person to manually search, filter, or even know the names of what they want.

For fashion e-commerce platforms, tools like this could massively improve customer experience. For users, it makes discovery fun again.

VI. CONCLUSION

In this project, we set out to build a personalized fashion recommendation system using deep learning and unsupervised clustering. Starting from just three user-provided images of men in formal attire, the system was able to understand the visual essence of the user's style and return clothing items that matched this aesthetic. The process involved extracting embeddings using CLIP, measuring similarity using cosine distance, and exploring broader style spaces through UMAP and HDBSCAN clustering.

One of the standout outcomes was how well the system performed with minimal input and no need for textual metadata. The use of CLIP allowed us to harness a pre-trained visual-language model capable of capturing rich, high-level features. Meanwhile, the combination of UMAP for dimensionality reduction and HDBSCAN for density-based clustering gave structure to an otherwise unlabelled dataset, enabling intelligent grouping of fashion styles.

Comparing the two recommendation approaches—cosine similarity and cluster-based—we found that both have value, depending on the user's needs. Cosine similarity excels in pinpoint accuracy for similar looks, while HDBSCAN-based recommendations offer stylistic diversity by navigating through clusters that share core visual traits.

Overall, this project demonstrates the real-world potential of multimodal AI tools like CLIP in domains such as fashion, where personal style is visual, subjective, and hard to define with labels. It also highlights the power of unsupervised learning for making sense of complex datasets without relying on costly annotation. With some refinement—such as adding support for text prompts or integrating user preferences—this system could easily scale into a robust style discovery tool for e-commerce, fashion enthusiasts, or even stylists.

In the future, blending vision with personalization, metadata filtering, and interactive UX will be key to making these kinds of systems not only intelligent but also delightful to use.

REFERENCES

- [1] Radford, A., Kim, J. W., Hallacy, C., et al. "Learning Transferable Visual Models From Natural Language Supervision." *Proceedings of the 38th International Conference on Machine Learning (ICML)*, 2021.
- [2] McInnes, L., Healy, J., & Astels, S. "hdbscan: Hierarchical density-based clustering." *Journal of Open Source Software*, 2(11), 205, 2017.
- [3] Van der Maaten, L., & Hinton, G. "Visualizing Data using t-SNE." *Journal of Machine Learning Research*, 9(Nov):2579-2605, 2008.
- [4] Wolf, T., Debut, L., Sanh, V., et al. "Transformers: State-of-the-art Natural Language Processing." *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 2020.

- [5] **Pedregosa, F., Varoquaux, G., Gramfort, A., et al.** "Scikit-learn: Machine Learning in Python." *Journal of Machine Learning Research*, 12, 2825–2830, 2011.
- [6] **McInnes, L., Healy, J., Saul, N., & Großberger, L.** "UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction." *arXiv preprint arXiv:1802.03426*, 2018.
- [7] **Agrawal, P.** "Fashion Product Images Dataset." *Kaggle*, 2018. <https://www.kaggle.com/datasets/paramaggarwal/fashion-product-images-dataset>
- [8] **Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J.** "Distributed Representations of Words and Phrases and Their Compositionality." *Advances in Neural Information Processing Systems (NeurIPS)*, 2013.
- [9] **Reif, E., Yuan, A., Tao, M., et al.** "Visualizing and Measuring the Geometry of BERT." *NeurIPS 2019 Workshop on Interpretation*, 2019.
- [10] **Ester, M., Kriegel, H.-P., Sander, J., & Xu, X.** "A density-based algorithm for discovering clusters in large spatial databases with noise." *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining (KDD)*, 1996.