Lipstick Recommendation System

A Data-Driven Approach to Personalized Beauty

Team #23 - Shubin Luan

Abstract

The cosmetics industry is undergoing a significant shift toward personalization, with consumers increasingly seeking products tailored to their unique preferences and characteristics. This report presents the development of a **Lipstick Recommendation System** that leverages data science methodologies to provide personalized product suggestions based on individual user profiles. The system integrates data processing, machine learning, and software engineering to enhance the consumer shopping experience. By analyzing extensive product data and customer feedback, the system aims to simplify the decision-making process for consumers. This report details the problem addressed, the methodologies employed, the evaluation strategies, and the broader impacts of the system, including ethical considerations.

Project Statement

The core problem addressed in this project is the development of a personalized **Lipstick Recommendation System** that efficiently matches consumers with lipstick products best suited to their individual characteristics and preferences. The system analyzes user-provided data, such as uploaded images reflecting their skin tones, and recommends products that not only match in color but also meet quality standards indicated by customer reviews and ratings.

By tackling this problem, the system seeks to enhance the consumer shopping experience, reduce decision fatigue, and provide a more engaging and satisfying interaction with cosmetic products.

Methodology

Data Collection Architecture

The foundation of the Lipstick Recommendation System lies in data collection. Utilizing Python and its extensive library ecosystem, a web scraping system was developed to extract comprehensive product information from Sephora's e-commerce platform. The requests library facilitated HTTP communication, while BeautifulSoup parsed HTML content to navigate complex page structures and extract relevant data efficiently.

Over 1,100 lipstick products were collected, capturing attributes such as brand names, product descriptions, color options, pricing, ratings, and URLs to product images. The data was stored in a structured format using pandas DataFrames, enabling efficient manipulation and analysis.

Initial Data Processing

To enrich the dataset with consumer insights, over 147,000 customer reviews were collected. This extensive corpus provided valuable data on consumer sentiments, preferences, and perceptions of product quality. Data cleaning involved handling missing values and normalizing data to ensure consistency. Integrating reviews with product data allowed for a comprehensive analysis of products from both technical specifications and consumer satisfaction perspectives.

Image Processing and Color Extraction

Image Processing

To efficiently store and handle image data within the dataset, images were encoded into Base64 strings. This encoding transformed binary image data into ASCII strings, allowing for easy inclusion in pandas DataFrames and facilitating data transfer between the backend and frontend systems. Base64 encoding also enabled inline image rendering in web applications without the need for separate image files.

The encoded images were stored in two new columns within the DataFrame: Cover Image and Lipstick Image. The Cover Image column holds the representative color of the lipstick, while Lipstick Image contains the image of the lipstick.

Analysis of RGB Color Distributions

The RGB values were normalized to ensure compatibility with machine learning algorithms. This normalization preserved the relative differences between colors while standardizing the data range.

To understand the overall color distribution of the lipstick products, an analysis of the RGB color values was conducted. This generated visualizations of chromatic distributions, including density plots for each color channel and a 3D scatter plot of the RGB color space.

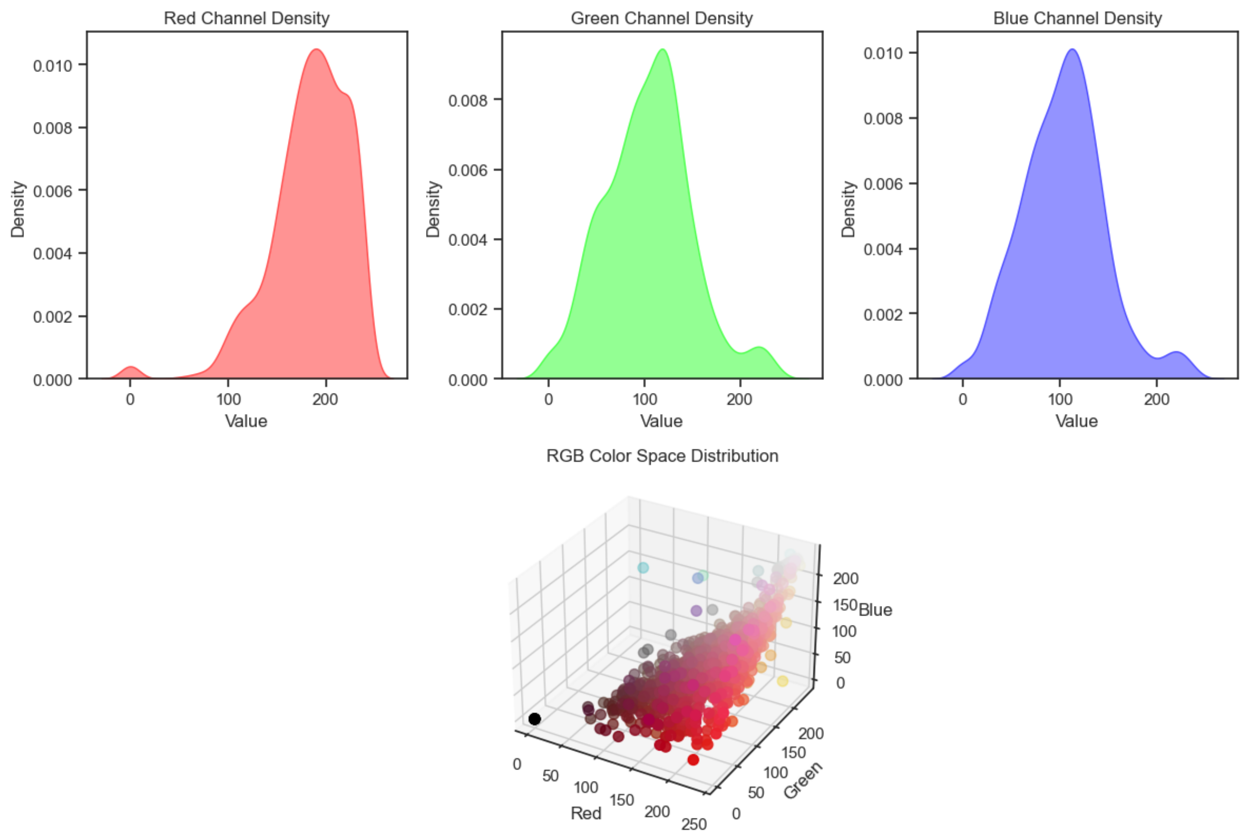

Figure 1: Visualization of RGB Color Distributions. The top row shows KDE plots for the Red, Green, and Blue channels, respectively. The bottom panel displays a 3D scatter plot of the RGB color space distribution.

Clustering Analysis

Determining Optimal Clusters

To categorize lipsticks into meaningful color groups, the K-Means clustering algorithm was employed. Selecting the optimal number of clusters (*k*) was a critical step. A cross-validation framework was implemented, evaluating clustering performance across different *k* values using metrics such as the Silhouette Score, Calinski-Harabasz Index, and Davies-Bouldin Index. The Silhouette Score measures how similar an object is to its own cluster compared to other clusters, with higher values indicating better clustering. The Calinski-Harabasz Index assesses cluster separation and compactness, while the Davies-Bouldin Index evaluates cluster overlap. The analysis indicated that **k = 6** provided the best balance between cluster cohesion and separation.

- **Silhouette Score**: 0.389 ± 0.001
- **Calinski-Harabasz Score**: 13,643.531 ± 262.525
- **Davies-Bouldin Score**: 0.818 ± 0.002

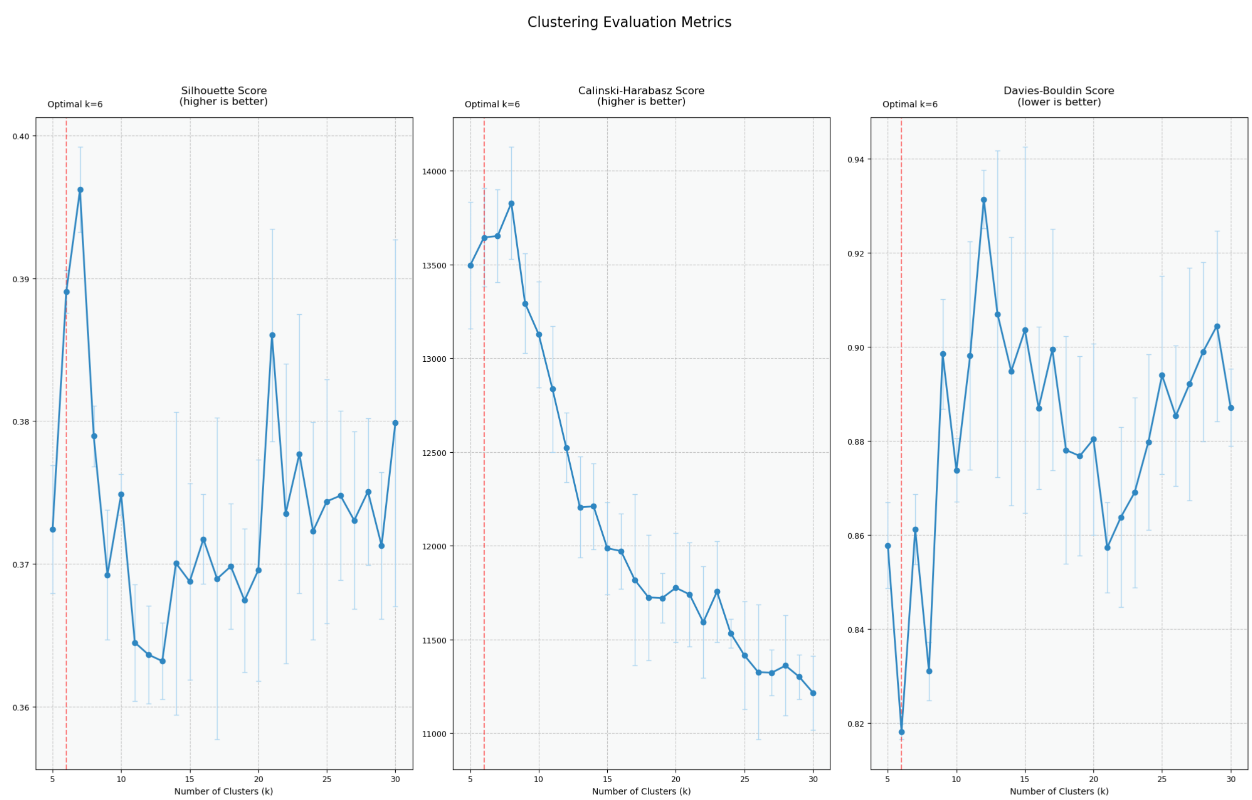

Figure 2: Evaluation of Clustering Performance Metrics Across Different Values of k.

Cluster Interpretation

Each cluster was analyzed to understand its predominant color characteristics. Clusters were assigned descriptive names such as “Classic Red,” “Soft Pink,” and “Warm Brown.” This categorization facilitated personalized recommendations by aligning products with user-preferred color groups.

Visualization of the clusters in RGB space provided insights into the distribution of lipstick shades and the distinctiveness of each cluster.

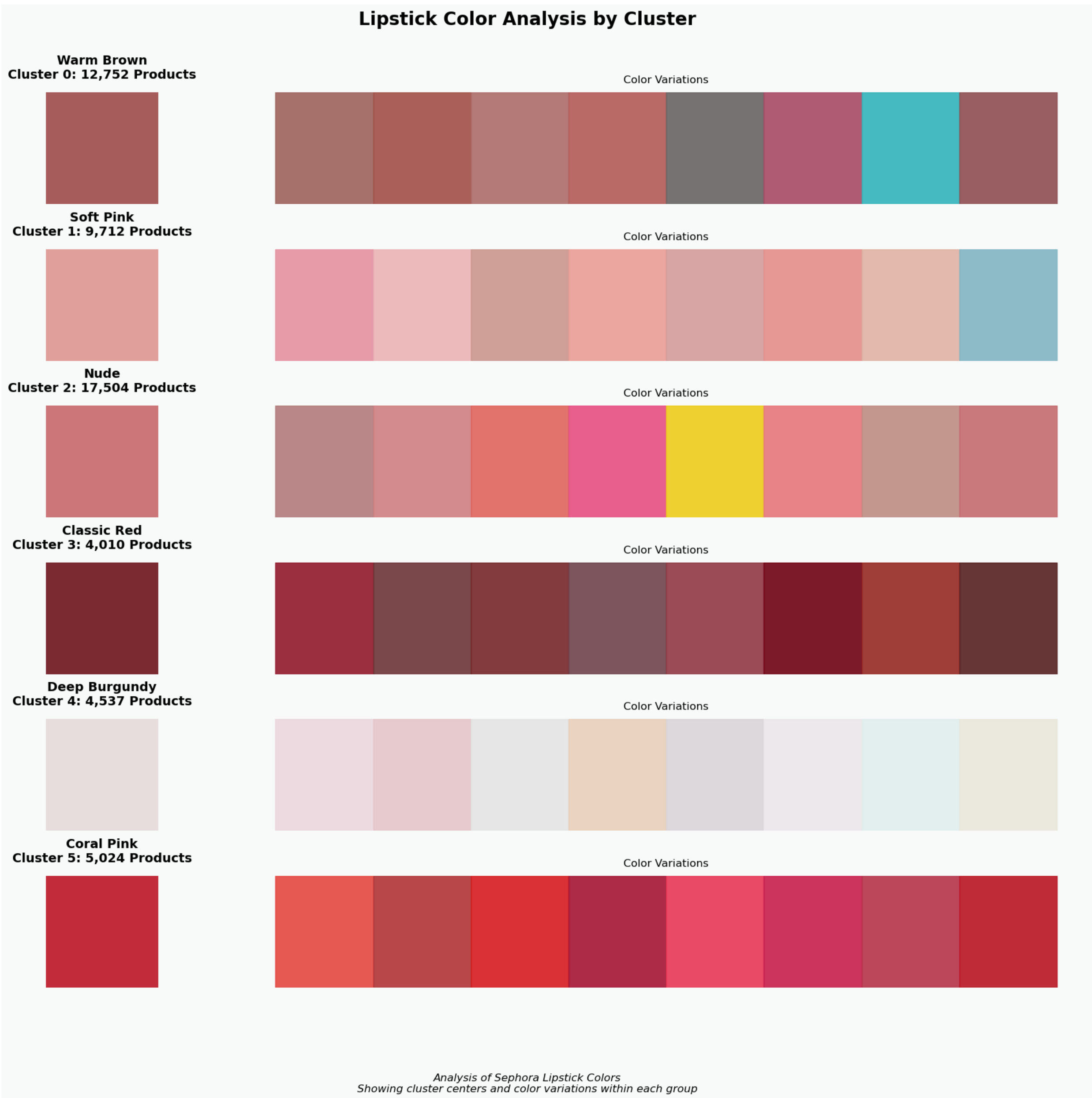

Figure 3: Visualization of Lipstick Color Clusters in RGB Space. Each cluster is represented by a different color, illustrating the distinct groupings of lipstick shades.

Visualizing Color Composition

To visualize the distribution of lipstick colors across the clusters, a circular composition (pie chart) was created, representing each cluster's proportion in the dataset.

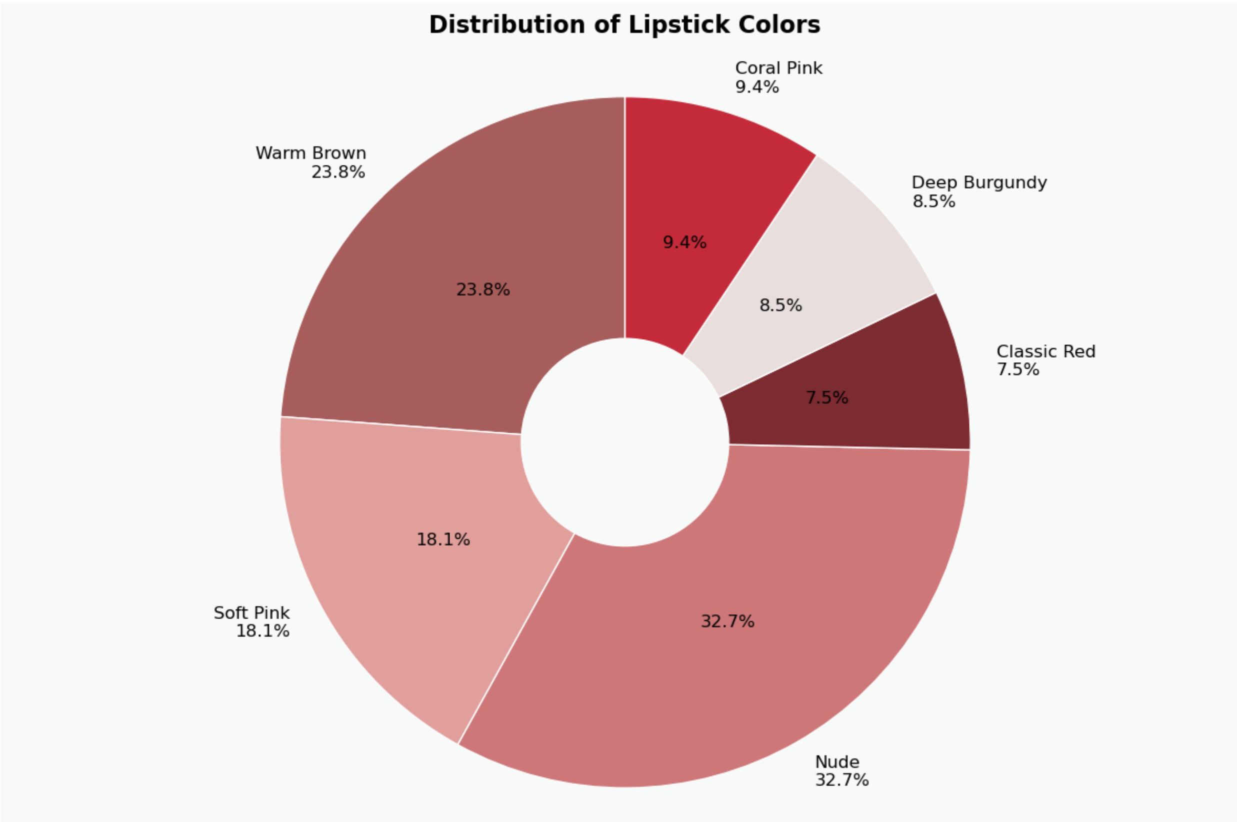

Figure 4: Distribution of Lipstick Colors Across Clusters. Each segment represents a color cluster, labeled with its name and percentage share. The colors correspond to the cluster centroids in RGB space.

Recommendation Engine Development

Weighted Scoring Algorithm

The recommendation engine was designed to integrate multiple factors influencing consumer choices, specifically focusing on product ratings and the number of reviews. The calculation of the recommendation score employed a weighted approach that balanced these factors to reflect both product quality and popularity.

The scoring methodology assigned a weight of 70% to the product's average rating and 30% to the number of reviews it received. This weighting emphasized the importance of product quality as perceived by consumers while still considering the influence of product popularity.

To account for the wide range of review counts and mitigate the impact of products with exceptionally high numbers of reviews overshadowing others, logarithmic scaling was applied to the review counts. This approach acknowledged the diminishing returns of additional reviews in influencing consumer perception.

The recommendation score for each product was calculated by normalizing the average rating to a scale between 0 and 1, dividing by the maximum possible rating. The scaled review count was similarly normalized by dividing the logarithmically scaled review count by the logarithm of the maximum review count across all products. The weighted sum of these two normalized values was then scaled to a percentage for easier interpretation.

This scoring method ensured that products with high ratings and a substantial number of reviews were prioritized in the recommendations, providing users with high-quality and popular options that align with their preferences.

Personalization and Filtering

After a customer uploads their photo, the system uses Google MediaPipe to analyze the image and determine the customer's skin tone. The extracted skin tone data is input into the pre-trained K-Means clustering model, which classifies the customer into one of six predefined skin tone clusters.

The recommendation engine then selects lipstick products from the color cluster that complements the customer's skin tone cluster. This process enhances the personalization aspect of the recommendations by ensuring that the suggested products are well-suited to the individual's complexion.

Customers have the flexibility to sort the displayed products by "Recommendation Score", "Number of Reviews", "Rating" or "Price". They can also apply filters to refine the product list according to their preferences, such as setting a desired price range or selecting the number of products to display per page.

Additionally, each lipstick image in the recommendation list is clickable. When a user clicks on a lipstick image, they are redirected to the product page of the lipstick on the Sephora website. This feature provides a seamless transition from recommendation to purchase, allowing users to access detailed product information and complete transactions efficiently.

Frontend and Backend Implementation

Frontend Development

The frontend was developed using React.js, leveraging features such as hooks and the Context API for efficient state management. Material-UI provided a consistent and responsive design framework, with custom themes aligning with the aesthetic of a cosmetics application. Key components included:

- **ImageUpload Component:** Allowed users to upload images, which were processed client-side for immediate feedback. Client-side processing ensured faster response times and reduced server load.
- **RecommendationView Component:** Displayed recommended products with dynamic animations using the Framer Motion library. Products were presented in an interactive grid, with sorting and filtering options readily accessible.

Backend Development

The backend was implemented using Flask, providing a lightweight yet powerful framework for API development. RESTful API endpoints were established for:

- **Image Processing:** Received images from the frontend, and extracted color features.
- **Recommendation Generation:** Computed recommendation scores based on user inputs and returned a list of recommended products.
- **Feedback Collection:** Stored user feedback for continuous improvement of the system.

Cross-Origin Resource Sharing (CORS) policies were configured to ensure seamless communication between the frontend and backend.

Integration of MySQL Database

To manage and store user feedback effectively, a MySQL database was integrated into the backend infrastructure. Utilize MySQL containers through Docker isolated database environments. The database captured user feedback on product recommendations, enabling the collection of qualitative and quantitative data. It leveraged stored data to refine algorithms, adjust recommendation weights, and enhance personalization features based on real user input.

Containerization with Docker

To ensure scalability and ease of deployment, the application was containerized using Docker. Separate containers were created for the frontend, backend, and database services, facilitating independent development and deployment cycles. Docker Compose was used to manage multi-container applications, simplifying the orchestration of services.

Evaluation Strategy

Lipstick Recommender System Evaluated through user feedback on the suitability of recommended products. Users provide numerical ratings (1-5 stars) and written feedback on their recommended products. By analyzing users' ratings and feedback sentiments, it is possible to assess users' satisfaction with the product, which features they favor, and which features they suggest for improvement.

Technical Depth

Data Mining and Web Scraping

Techniques were employed to collect data from complex web sources. The system handled dynamic content, pagination, and nested data structures, utilizing asynchronous requests mechanisms to ensure data integrity.

Machine Learning and Clustering

Unsupervised learning algorithms were applied for clustering lipstick shades based on color features and for skin tone analysis from user-uploaded images. The project involved selecting appropriate clustering techniques, determining optimal parameters, and interpreting the results. Cross-validation and multiple evaluation metrics were used to validate the clustering outcomes.

Data Visualization

Data visualization was integral to interpreting and presenting results. Libraries such as matplotlib and seaborn were used to create visualizations that elucidated clustering performance, color distributions. Visualizations were designed to be informative for audiences.

Database Management

Integration of a MySQL database demonstrated proficiency in database management and data storage solutions. The use of MySQL containers allowed for scalable and efficient handling of user feedback data, critical for the system's continuous improvement.

Translation of Results

The Lipstick Recommendation System personalized the shopping experience by aligning product suggestions with individual user preferences and skin tones. Numerical results from clustering analysis and recommendation scoring translated into tangible benefits for users:

1. Users were presented with a curated list of products that matched their skin tone, reducing the time and effort required to find suitable options.
2. High relevance of recommendations, coupled with the ability to sort and filter products, led to increased user satisfaction and a positive perception of the platform.
3. Users discovered new products within their skin tone clusters, potentially leading to increased sales and brand exposure.

Test Results

To evaluate the system's performance across diverse skin tones, testing was conducted using facial images representing different racial and ethnic backgrounds. All test images were sourced from freepik, and are copyrighted by freepik.

Distances to cluster centers ranging from 21.12 to 36.85 in RGB space. This range indicates reasonable clustering performance while acknowledging natural variations in skin tone.

Image	Extracted RGB	ID	Cluster Name	Cluster Center RGB	Distance to Center
Asian Face	[168 124 98]	0	Warm Brown	[165 92 90]	33.12
African Face	[139 100 72]	0	Warm Brown	[165 92 90]	32.62
European Face	[191 144 117]	5	Coral Pink	[205 118 120]	29.68
Latin American Face	[181 136 113]	5	Coral Pink	[205 118 120]	30.81
Middle Eastern Face	[214 172 142]	1	Soft Pink	[223 158 155]	21.12
Native American Face	[213 191 168]	1	Soft Pink	[223 158 155]	36.85

Broader Impacts

The implementation raised several important considerations for e-commerce personalization. Privacy concerns were addressed by processing user images in real-time without storing them. The system's color matching algorithm was tested across different skin tones to ensure consistent performance. Test results showed consistent performance across Asian, African, European, Latin American, Middle Eastern, and Native American facial features, with similar clustering accuracy and recommendation quality.

The project demonstrated how data science techniques can improve online shopping efficiency while highlighting areas requiring careful attention, such as data privacy and algorithmic fairness. These findings may inform future developments in e-commerce personalization.

Critical Evaluation

The integration of Google MediaPipe enabled precise skin tone detection. And user feedback was instrumental in refining the recommendation algorithm. Nevertheless, variations in image quality affected skin tone detection accuracy. Furthermore, the dataset's focus on products from a single retailer may limit diversity. Regular reviews are necessary to ensure fairness in recommendations.

Statement of Work

I completed all aspects of this project independently, including system design, implementation, testing, and documentation. The work involved developing the recommendation algorithm, implementing the database structure, and building the user interface. The project integrated concepts from data science, software engineering, and user experience design.

References

Kalinova, M. (2024). Transforming beauty with AI: The future of personalization. LinkedIn Pulse. Retrieved from https://www.linkedin.com/pulse/transforming-beauty-ai-future-personalization-milena-kalinova-k81ee/

Pandey, N., & Krishna, C. (2021). Virtual makeup try-on using deep learning. GitHub Repository. Retrieved from https://github.com/Charan0/Virtual-Makeup

Zhang, L., Chen, Y., & Wang, H. (2021). Color-based recommendation system for cosmetic products using machine learning approaches. arXiv:2109.08794 [cs.CV]. Retrieved from https://arxiv.org/pdf/2109.08794

Appendix

Table 1: Lipstick Product Data Schema

Field Name	Type	Description	Example
Brand	string	Cosmetics brand name	"Anastasia Beverly Hills"
Name	string	Product display name	"Liquid Lipstick"
Color Description	string	Color description	"American Doll classic"

Color Cluster	integer	Color group (0-5)	3
RGB	string	RGB color value	"rgb(149,19,44)"
Rating	float	User rating (1-5)	4.5
Reviews	integer	Number of user reviews	1250
Price	string	Product price	"$20.00"
URL	string	Product URL	"https://www.sephora..."
Cover Image	string	Base64 encoded cover image	(base64 string)
Lipstick Image	string	Base64 encoded lipstick image	(base64 string)
Recommendation Score	float	Overall score (0-100)	85.67

Color Cluster Classifications

1. Warm Brown
2. Soft Pink
3. Nude
4. Classic Red
5. Deep Burgundy
6. Coral Pink

Recommendation Score Calculation

Score = $(0.7 \times R/5 + 0.3 \times \ln(N+1)/\ln(N_{max}+1)) \times 100$

- R = Product rating (1-5)
- N = Number of reviews for the product
- Nmax = Maximum number of reviews in the dataset

The final score is rounded to two decimal places and ranges from 0 to 100.