Make Up Products Analysis with Prediction

- Nice One Website -

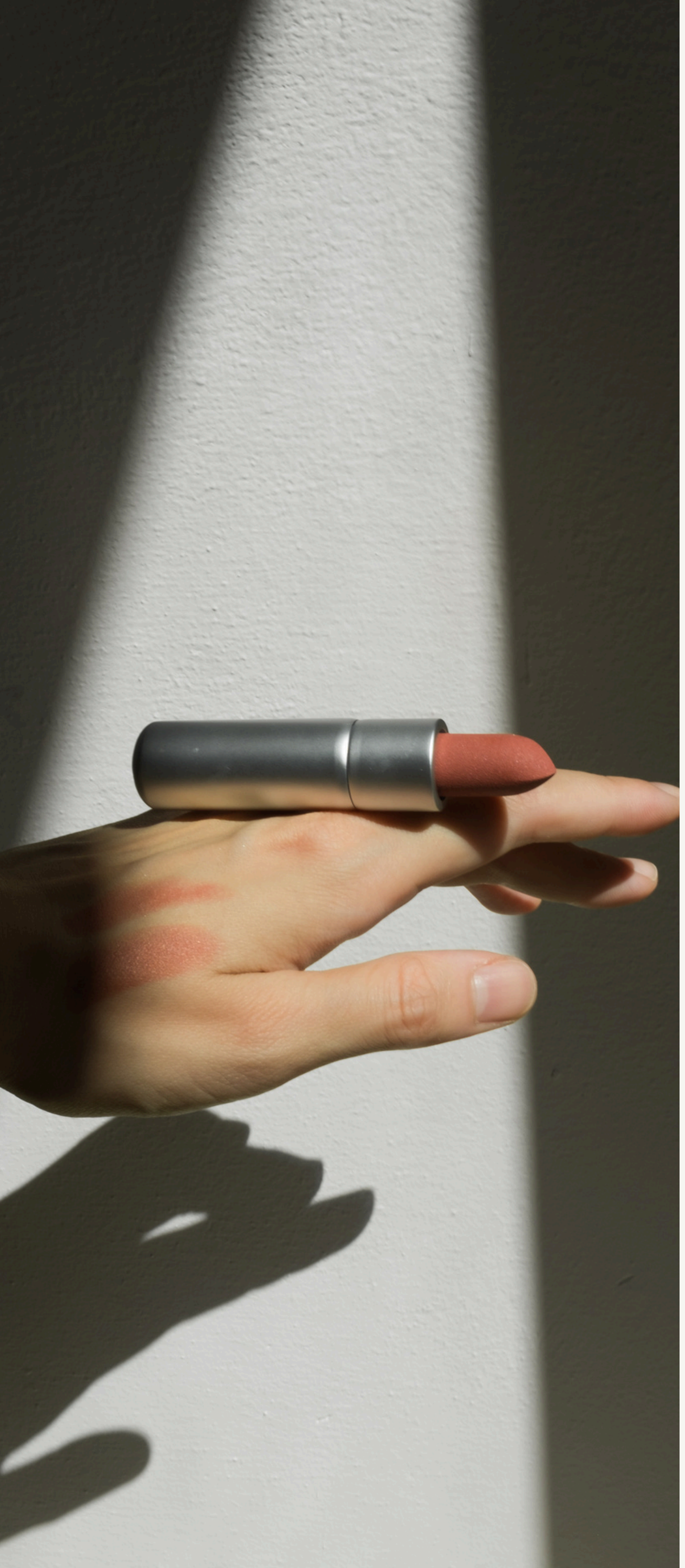
Presented By:
Munirah Alzuman
Raghad Alharbi
Yaqeen Alhalal
Tahani Alotaibi

Introduction

RAMADAN MUBARAK! 🌙✨

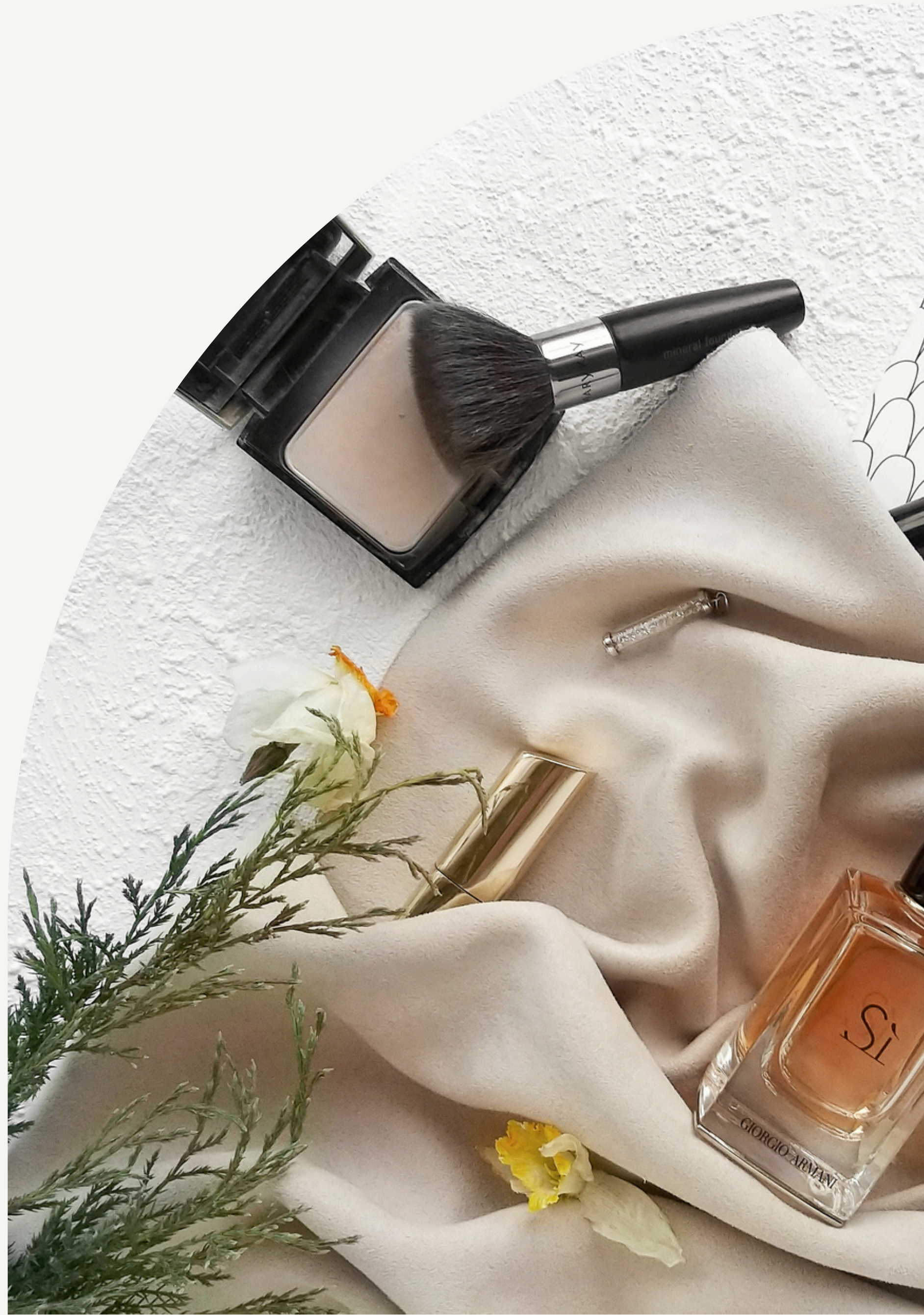
Are you ready to glow this Eid with stunning new makeup? Or perhaps you're a beginner eager to dive into the beauty world without spending a fortune? No matter what brings you here, you’ve come to the right place!

From budget-friendly must-haves to luxurious beauty essentials, we’ve got everything you need to create your perfect look.



Problem Statement

Customers often struggle to find makeup products that fit their budget and preferences. This project aims to use K-means clustering to group makeup products based on price, discount price, makeup type, brand name, and review count, helping users discover low-cost, affordable, and luxury options more easily.



Objectives

1

Explore the dataset to uncover trends in pricing, ratings, and customer preferences.

2

Perform feature engineering to enhance the dataset for modeling.

3

Build predictive models to analyze patterns.

4

Provide actionable insights that can help optimize business strategy

Data Collection

In the Data Collection phase, we gathered data from the makeup section of the Nice One website on March 7, 2025. A script was developed to extract detailed information for each product, including:

- Product name
- Brand name
- Original price
- Discounted price
- Rating number
- Reviews number
- Skin type
- Makeup type
- Product texture

To accomplish this, we utilized Web Scraping techniques using Python libraries (Selenium and BeautifulSoup) to extract the required data accurately and efficiently.

Data Collection EDA

The Main Data Quality Issues:

1- ***Uniqueness***: Some products appear more than once due to their range of shade options. We chose to keep these duplicates as they are accurate.

2- ***Completeness***: Some rows were filled with 'Not Available' for all features, indicating that they weren't scraped. Therefore, they were dropped.

3- ***Accuracy***:

- The original price, rating number, and reviews number were scraped and saved as objects (strings). Their data types were corrected to float for the original price and rating, and int for the number of reviews.
- All the outliers in the dataset are natural outliers.

Data Collection EDA

Feature Engineering:

- **Retain only the rows where categorical features appear more than three times in the dataset**. This ensures that infrequent categories are excluded to improve model robustness.
- **Encoding Categorical Features** : Apply Frequency Encoding to transform categorical features into numerical representations based on their occurrence frequency in the dataset.
- **Feature Scaling**: Standardize or normalize the features to ensure they are on the same scale.
- **Feature Selection**: Focus on key features for the analysis, such as:
    - Original Price
    - Price After Discount
    - Brand Name
    - Makeup Type
    - Review Count

Data Collection EDA Modellling

Model: K-means clustering

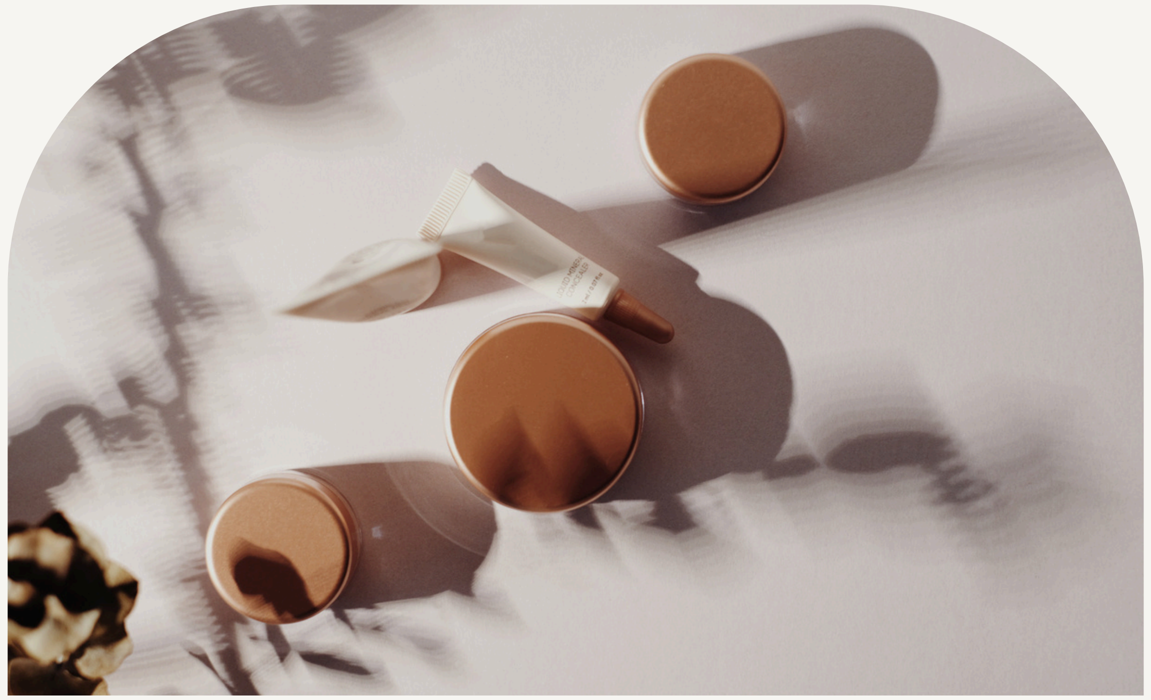
Purpose: To categorize makeup products into distinct groups based on key features such as price, discounted price, makeup type, brand name, and review count. This approach helps users effortlessly discover products that match their preferences and budget, including:

- **Affordable Essentials**: Cost-effective options for everyday use.
- **High-Quality Tools:** Premium products known for their performance and durability.
- **Luxury Essentials**: High-end, exclusive makeup items for a luxurious experience.
- **Budget-Friendly Options**: Products that combine quality with affordability.

Data Collection EDA Modellling Model Deployment

We deployed the model using FastAPI, starting with installing the necessary libraries, then creating a file containing the FastAPI code to run the API. Next, we ran the application locally and tested it via Swagger UI. Later, we uploaded the project to GitHub and deployed it on Render by setting up a Web Service and linking it to the repository, specifying the installation and startup commands. Additionally, we integrated the API with Streamlit to create an interactive user interface for sending data to the model and displaying predictions. After deployment, we could access the API through the final Render URL for use in various applications.

https://app-app-idrjhys4fh8ndhuqznrprq.streamlit.app



Conclusion

Our analysis provides valuable insights into makeup product trends on Nice One, helping customers discover products that match their budget and preferences. By leveraging K-Means clustering, we categorized products we offering a more personalized shopping experience. The integration of FastAPI and Streamlit allows for real-time analysis, making it easier to explore and compare products. This project can support Nice One in optimizing pricing strategies, enhancing product recommendations, and improving customer satisfaction. Future improvements could involve incorporating more user behavior data to refine predictions and further personalize recommendations.