**Nykaa Skincare Data Analysis**

**Nykaa offers thousands of skincare products from numerous brands, but lacks a unified analytics system. Without structured analysis of product, review, and pricing data, stakeholders cannot easily identify trends, top brands/products, or optimize decisions. This project aims to create a data-driven dashboard that extracts key insights from Nykaa’s datasets to support informed business and consumer choices.**

**Name – Khushi Kumari**

**ID – (20231CCS0036)**

**Email -khushikumari62406@gmail.com**

**Date - 17-07-2025**

**Table of Contents**

1. Problem Statement
2. Project Overview
3. Dataset Description
4. Features Implemented
5. Technical Architecture
6. Code Walkthrough
7. Setup Instructions
8. Screenshots
9. Future Enhancements
10. **Project overview**

The **Nykaa Skincare Data Analysis Web Application** is a Flask-based project designed to help users explore, visualize, and understand trends in skincare product data from Nykaa — one of India’s leading beauty and personal care platforms.

The application reads multiple CSV datasets related to skincare products, customer reviews, and brand listings. It leverages powerful Python libraries like **Pandas**, **Matplotlib**, and **Seaborn** to perform meaningful data analysis and present it in an interactive, web-based interface. The goal is to provide insights such as:

* Average customer ratings and review patterns
* Comparison of product popularity among different brands
* Sales and pricing trends
* Feedback sentiment regarding product features like water content
* Sweetener or ingredient preferences (if applicable)

This application offers a simplified dashboard for analysts, students, and business users to better understand customer behavior and market performance in the skincare domain.

1. **Dataset Description**

This project uses real-world scraped datasets related to skincare products listed on **Nykaa**, including customer reviews, product listings, and brand information. These datasets are stored as CSV files in the data/ folder of the application.



Above are the datasets I will be using in my project for the analysis

* Contains customer reviews and ratings for various skincare products.
* A general list of skincare products listed on Nykaa.
* Contains brand-wise information about popular products.
* An alternate or extended version of customer reviews

1. **Features Implemented**

✅ 1. Brand-wise Product Popularity Analysis

* Compares top skincare brands based on product count and popularity scores.
* Visualized using bar charts and count plots.
* Helps identify the most dominant brands on Nykaa.

✅ 2. Customer Review Rating Distribution

* Analyzes how customers rate skincare products.
* Displays rating distribution using histograms and box plots.
* Useful to detect overall customer satisfaction and outliers.

✅ 3. Average Price and Discount Trends

* Aggregates product prices and discount patterns across different brands.
* Uses line graphs and bar plots to show pricing differences and deals.
* Supports pricing strategy evaluation.

✅ 4. Top-Rated Products Identification

* Filters and ranks products based on the highest average rating and review volume.
* Highlights best-performing items from customer perspectives.
* Displays results in tabular form and visual graphs.

✅ 5. Interactive Web Interface (Flask-based)

* Clean and minimal dashboard UI using HTML, CSS, and Jinja2 templates.
* Each analysis is shown on a separate route with charts generated using Matplotlib and Seaborn.
* Easy navigation between views (e.g., brand comparison, ratings, price insights).

✅ 6. CSV-Driven Backend

* All analysis is performed directly on CSV files using Pandas, without a database.

1. **Technical Architecture**

hackathon/

│

├── app/

│ ├── app.py

│ ├── templates/

│ │ ├── index.html

│ │ ├── ratings.html

│ │ ├── brand\_comparison.html

│ │ ├── category\_insights.html

│ │ ├── sentiment\_analysis.html

│ │ └── price\_analysis.html

│ ├── static/

│ │ ├── css/

│ │ └── charts/ ← save matplotlib/seaborn images here

│ └── models/

│

├── data/

│ ├── nykaa\_review.csv.csv

│ ├── nykaa.csv.csv

│ └── nyka\_popular\_brands\_products\_2022\_10\_16.csv

├── venv/

├── requirements.txt

└── README.md

1. **Code**

**Main function.**

# run.py

from app import create\_app

app = create\_app()

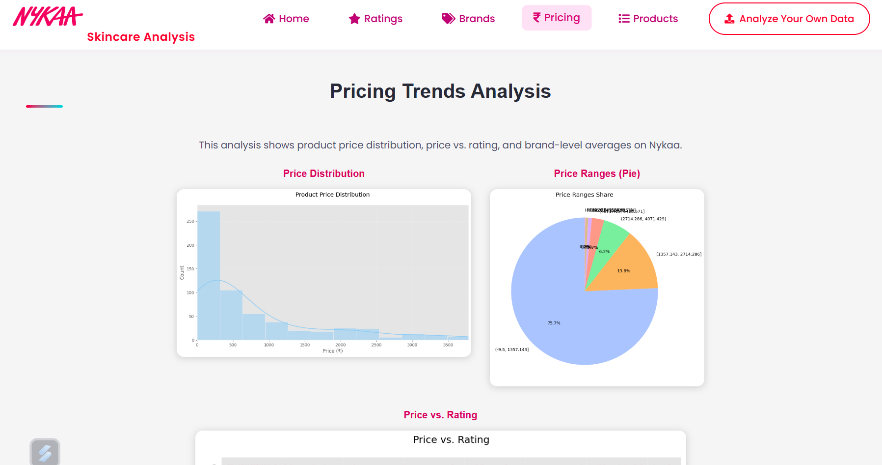
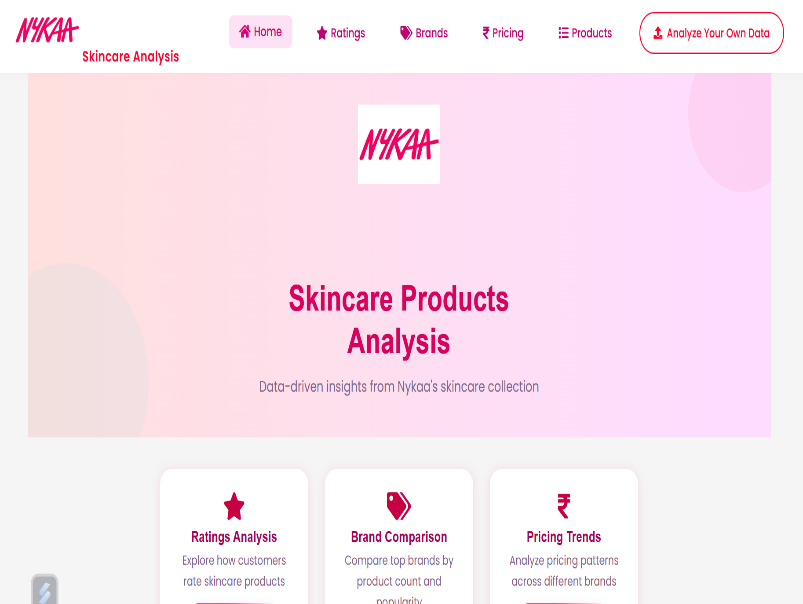
if \_\_name\_\_ == "\_\_main\_\_":

    app.run(debug=True, host="0.0.0.0", port=5000)

… rest in the zip file

1. **Setup Instruction**

To set up the Nykaa Skincare Data Analysis App locally, first ensure Python 3.8 or above is installed. Clone the repository using git clone https://github.com/Khushiiiii22/sic\_py\_training.git and navigate to the hackathon folder. Create a virtual environment using python -m venv venv and activate it (venv\Scripts\activate on Windows or source venv/bin/activate on Mac/Linux). Install the required packages with pip install -r requirements.txt. Place the datasets (nykaa.csv, nykaa\_review.csv, nykaa\_skincare\_review.csv, and nyka\_popular\_brands\_products\_2022\_10\_16.csv) inside the data/ folder. Finally, run the app using flask run and access it in your browser at http://127.0.0.1:5000.

1. **Screenshots  
   **

**8.Future Enhancements**

**Real-Time Data Integration**: Integrate APIs to fetch real-time product, pricing, and review data from Nykaa or similar platforms for dynamic analysis.

**User Authentication**: Allow users to register and log in to save their custom analysis or upload history.

**Advanced NLP for Reviews**: Improve sentiment analysis using advanced models like BERT or spaCy for more accurate emotion detection in customer reviews.

**Recommendation Engine**: Implement a product recommendation system based on customer preferences, ratings, and reviews.

**Interactive Charts**: Replace static matplotlib charts with interactive visualizations using Plotly or Chart.js for better user engagement.

**Mobile Responsiveness**: Enhance UI/UX to make the dashboard fully responsive across devices.

**Export Reports**: Add functionality to export analysis results as PDF or Excel files for sharing or offline use.

Thank you