DEPARTMENT OF INFORMATION TECHNOLOGY
VIDYALANKAR SCHOOL OF INFORMATION TECHNOLOGY
(Affiliated to University of Mumbai)
MUMBAI, 400 037
MAHARASHTRA
2020 – 2021

ANALYSIS OF SKIN CARE PRODUCTS

A Project Report
Submitted in partial fulfillment of the
requirements for the award of the Degree of
MASTER OF SCIENCE (INFORMATION TECHNOLOGY)
By

Maheshwar Singh — Seat Number: 19306A1004
Bhushan Salunke — 19306A1046

Under the esteemed guidance of
Shajil Kumar
Assistant Professor, Department of Information Technology

VIDYALANKAR SCHOOL OF INFORMATION TECHNOLOGY
(Affiliated to University of Mumbai)
MUMBAI-MAHARASHTRA-400037
DEPARTMENT OF INFORMATION TECHNOLOGY

VSIT | Vidyalankar School of Information Technology

CERTIFICATE

This is to certify that the project entitled, **"Analysis of Skin Care Products "**, is bona fide work of **Maheshwar Singh** bearing Seat No: 19306A1004 and **Bhushan Salunke** bearing Seat No: 19306A1046 submitted in partial fulfilment of the requirements for the award of degree of MASTER OF SCIENCE in INFORMATION TECHNOLOGY from University of Mumbai.

Internal Guide | **Coordinator**

Internal Examiner | **External Examiner**

Date: | **College Seal** | **Principal**

ABSTRACT

Proper skin care is important because our skin is the largest barrier against infection that we have. Keeping our skin healthy and moist helps keep this barrier strong. These personal hygiene and skincare activities are integral parts of the nursing practice. This study examines the many different aspects related to which age group is more concerned with their skin and how better they know it. There is a huge need for finding out the common ingredient between the products and what the people’s sentiments are towards.

Among the challenges of the pharma companies is to fulfil the needs of the people by providing them the essential and suitable products according to their skin needs. Even it's very important to predict whether there will be more skin problems in the future if the same sample of products continues for a longer time.

There is also a lot of questions that arise when someone is about to start a new business as in this paper its about skincare products. So, knowing about which gender and age group is keener on which product and how much do they spend annually on it is important. What is preferable for them which will, in turn, affect their sales and marketing and will get to know to the current trend in the market. It will only target the customers which will be good for their business aspects.

There is a lot of need in the accuracy of the choice whether which product to use for their suitable skin to avoid any future skin problem. According to the survey, most people do not use samples before using any new product and due to which they face many problems and, in the end, the only solution remains is proper medication for that cause. For that, they visit dermatologists and knowledge comes to the second barrier though the doctors are well known about the problems most of the time the problems are not appropriate to the use of skin products. Through this paper one can find out want is similarities between the two products which can, in turn, give the broad idea of whether what caused this skin problem, which was that ingredient. This was related to avoiding future skin problems.

Nowadays many of the new start-up folks are not having the idea of whether to keep which products for how long and which is the current trend in the market. So, this project would help them to get the prescription of that. Also, whom to approach more to scale up their

business. It includes whether which age and gender are keener on their products which in turn helps them to do their sales and marketing in an efficient manner.

According to the survey it is found that people choose their products on the bases of product reviews more. So, for that, this paper includes the sentimental analysis for the products that customers want about in more graphical representation. As the data mining techniques are better to give the results in the graphical format as it is more understandable.

Also, most of the time prediction becomes an important concept to know what is going to happen in the future according to previous data to avoid further problems and raise efficiency.

Keywords: Products Management, Trend Analysis, Recommendation System, Skin type Management; Product Comparison; Sentimental Analysis, Classification, Time series analysis.

Keyword: *Products Management, Trend Analysis, Recommendation System, Skin type Management; Product Comparison; Sentimental Analysis, Classification, Time series analysis.*

ACKNOWLEDGEMENT

It gives me immense pleasure in expressing my heartfelt thanks to the people who were part of this project in numerous ways. I owe my thanks to all those who gave endless support right from the conception of the project idea to its implementation, it would not have materialized without the help of many.

The dedication, hard work, patience and correct guidance makes any task proficient & a successful achievement. Intellectual and timely guidance not only helps in trying productive but also transforms the whole process of learning and implementing into an enjoyable experience.

I would like to thank our Principal “**Dr. Rohini Kelkar**” and vice principal **“Mr. Asif Rampurawala”** for providing this opportunity, a special thanks to our MSc IT coordinator “**Ms. Beena Kapadia**” for their support, blessings and for being a constant source of inspiration to us. With immense gratitude, I would like to convey my special honour and respect to “M**r. Shajil Kumar**” **(Project Guide)** who took keen interest in checking the minute details of the project work and guided us throughout the same.

A sincere thanks to the non-teaching staff for providing us with the long lab timings that we could receive along with the books and with all the information we needed for this project, without which the successful completion of this project would not have been possible.

Finally, I wish to avail this opportunity & express a sense of gratitude and love to my friends and my beloved parents for their support, strength and help for everything.

Mr. Maheshwar Singh & Mr. Bhushan Salunke

DECLARATION

I hereby declare that the project entitled, **“Analysis of Skin Care Products”** done at Vidyalankar School of Information Technology, has not been in any case duplicated to submit to any other universities for the award of any degree. To the best of my knowledge other than me, no one has submitted to any other university.

The project is done in partial fulfillment of the requirements for the award of degree of **MASTER OF SCIENCE (INFORMATION TECHNOLOGY)** to be submitted as final semester project as part of our curriculum.

Mr. Maheshwar Singh

Name and Signature of the Student

Mr. Bhushan Salunke

Name and Signature of the Student

IMPLEMENTATION DETAILS

Nowadays many of the new start-up folks do not have an idea of whether to keep which products for how long and which is the current trend in the market. So, this project would help them to get the detailed analysis of that. Also, whom to approach more to scale up their business. It includes whether which age and gender are more likely to buy their products that will help them to do their sales and marketing in an efficient manner.

So, I conducted a survey and collected data by means of Google form and stored in the form of .csv file.

It contains about 500 rows:

Columns – **Skin type, age, gender, Importance of skin, mode of buying, purchase recommendation, try samples before use, choosing products (what makes them buy the particular product), face wash, face cream, body lotion, average spending, skin problem, skin problem type, ayurvedic products**.

Using this dataset, I implemented initial analysis. And made some decisions whether what is the possibility of product getting sold using Decision Tree Algorithm.

Also, applied Logistic Regression to find the accuracy of model and getting the false positive and true negative frequency of dataset.

The dataset is segregated values as per the age group and importance of skin and also, age group and annual spending on skin care products by using K-Means clustering.

According to the survey it is found that people choose their products on the bases of product reviews more. There is a lot of need in the accuracy of the choice whether which product to use for their suitable skin in order to avoid any skin problem. Whenever we want to try a new cosmetic item, it's so difficult to choose. It's sometimes scary because new items that I've never tried end up giving me skin trouble. We know the information we need is on the back of each product, but it's really hard to interpret those ingredient lists unless you're a chemist. So Instead of just being worried about our new choice, I decided to build a simple cosmetic recommendation.

Also created a modal where it returns the common ingredients between two products I applied jaccard similarity algorithm for this. This would be helpful for pharmacists to get the common ingredient names.

The implemented Sentimental Analysis for getting the positive and negative reviews out of the whole reviews for each product. It categorises the negative and positive reviews of product with the count of them.

For its implementation the data is scrapped from Sephora website. To focus on all skin care products. The dataset has 1472 items in total.

I took six different categories - moisturizing cream, facial treatments, cleanser, facial mask, eye treatment, and sun protection. This all includes in one field named **Label**.

It also has the information about the **brand, the price, the rank, skin types, reviews and the ingredients of each item.**

- **Technology Used:**

  For all this analysis I used **Python** which is an interpreted, high-level and general-purpose programming language**.** Highly used in **Data science**. For its overall implementation, I used Django framework for implementing UI and backend part and maintain MVC format.

- **Libraries Used:**

  To make some graphs visually more interactive, I used **Bokeh library** from python.

Pandas - It is a fast, powerful, flexible and easy to use open source data analysis and manipulation tool, built on top of the Python programming language.

Matplotlib and Seaborn - It provides a high-level interface for drawing attractive and informative statistical graphics.

Scikit-learn(Sklearn)- It is a machine learning library for Python. I have used **KMeans** for clustering , for classification used **Decision tree**, **Logistic regression**.

For recommendation system, I used **t-SNE algorithm** to check similarity between ingredients of products and recommending products for specific skin type and product type.

Nltk- Used for sentimental analysis to remove white spaces and create corpus and then apply **Naïve bayes**

Classifier to make out positive and negative reviews.

EXPERIMENTAL SET UP AND RESULTS

Python language with Django framework and VS code as an IDE for project working

There are two datasets in all used for data analysis and visualizations.

1. One is created by conducting survey and the other is scraped from the website.
2. Analysis is done on dataset using visualization tools and algorithms implemented to get best results.

Created the recommendation system using python algorithm and giving the recommendation of which product to use for their type of skin as per product type.

Done sentimental analysis to categorize positive and negative reviews for each product and get the graphical representation of same.

Applied some classification algorithms to get the accuracy of data set and getting the predictions from them and taking the decision from that predictions.

Some initial analysis from the dataset using python visualization libraries. Trend analysis of product according to gender, from the survey dataset.

METHODOLOGY

1. Initial graphical representation: *Used countplot*

   a. Which kind of skin type is maximum?

   b. Which gender uses more skincare products?

   c. Mode of buying of products

   d. Whom they get more purchase recommendation

   e. What do people look at while purchasing any product?

   f. Which face wash, face cream, body lotion is most used by people?

2. Clusters according to the importance of skin and annual spending on products as per the age

   a. A cluster refers to a collection of data points aggregated together because of certain similarities.

   b. You'll define a target number *k*, which refers to the number of centroids you need in the dataset. A centroid is the imaginary or real location representing the center of the cluster.

   c. Every data point is allocated to each of the clusters through reducing the in-cluster sum of squares.

d. In other words, the K-means algorithm identifies *k* number of centroids, and then allocates every data point to the nearest cluster, while keeping the centroids as small as possible.

e. The *'means'* in the K-means refers to averaging of the data; that is, finding the centroid.

3. Product recommendation system according to product and skin type:

a. Some pre-processing step before diving into.

i. The next step is cleaning the text data in the Ingredients column. The data is scraped from the Sephora page. As you can see below, there are two parts-descriptions for some particular compositions, the list of all ingredients, and additional info.

j. The result will be the lists in a list. In other words, the separated parts are put as a list, and the lists for each item are gathered inside a list. So how are we going to extract only the part for ingredients? My strategy here is using several patterns for detecting the unwanted part.

b. Applying NLP concept to chemicals.

i. To get to our end goal of comparing ingredients in each product, we first need to do some pre-processing tasks and bookkeeping of the actual words in each product's

ingredients list. The first step will be tokenizing the list of ingredients

in Ingredients column. After splitting them into tokens, we'll make a binary bag of words. Then we will create a dictionary with the tokens ingredient_idx as follows.

j. Passing through one by one, we first lower all the letters and split the text. All the tokenized words are put into the corpus.

k. The next step is making a document-term matrix (DTM). Here each cosmetic product will correspond to a document, and each chemical composition will correspond to a term. This means we can think of the matrix as a *"cosmetic-ingredient"* matrix.

c. **Dimension reduction with t-SNE.**

i. Algorithm used for nonlinear dimensionality reduction technique that is well-suited for embedding high-dimensional data for visualization in a low-dimensional space of two or three dimensions.

j. All of these cosmetic items in our data will be vectorized into two-dimensional coordinates, and the distances between the points will indicate the similarities between the items.

d. **Mapping the cosmetic items with Bokeh**

The data df_all here is the data with all possible combination of choices. The same process we've made so far is done. I just combined for all the possible options into one data frame. Therefore when we make a callback function and give it an optional choice, it will take the data for the given condition and update the plot.

4. What are the common ingredients between two products?

a. Used Jaccard similarity concept to get the common ingredients from two products.

b. You need to provide names of two products and you will get the outcome.

c. It gives you products as per the product type e.g. Moisturizer, Cleanser and select the names it will give the similarity.

5. Sentimental analysis:

With social media channels such as Facebook, LinkedIn, and Twitter, it is becoming feasible to automate and gauge what public opinion is on a given topic, news story, product, or brand. Opinions that are mined from such services can be valuable. Datasets that are gathered can be analyzed and presented in such a way that it becomes easy to identify if the online mood is positive, negative or even indifferent.

a. Removing stop words and creating the corpus.

b. Training the some predefined reviews that we get from nltk library by extracting features from the dictionary of this predefined reviews.

c. Now testing the actual reviews and getting the sentiments out of it i.e. positive or negative.

d. Visually representing the output by showing- review, predicted sentiment and probability value of the prediction because reviews does include false positive and true negative values in it. Also a count plot for count of positive and negative reviews of product.

ANALYSIS OF THE RESULTS

- **Technologies Used:**

For all this analysis I used **Python** which is an interpreted, high-level and general-purpose programming language**.** Highly used in **Data science**. For its overall implementation, I used Django framework for implementing UI and backend part and maintain MVC format.

- **Libraries Used:**

To make some graphs visually more interactive, I used **Bokeh library** from python.

Pandas - It is a fast, powerful, flexible and easy to use open source data analysis and manipulation tool, built on top of the Python programming language.

Matplotlib and Seaborn - Seaborne aims to make visualization a central part of exploring and understanding data. Its dataset-oriented plotting functions operate on data frames and arrays containing whole datasets and internally perform the necessary semantic mapping and statistical aggregation to produce informative plots.

Scikit-learn(Sklearn)- It is a machine learning library for Python. I have used **KMeans** for clustering, for classification used **Decision tree**, **Logistic regression**.

For recommendation system, I used **t-SNE algorithm** to check similarity between ingredients of products and recommending products for specific skin type and product type.

It is a library in Python that provides many unsupervised and supervised learning algorithms. It’s built upon some of the technology you might already be familiar with, like NumPy, pandas, and Matplotlib!

The functionality that scikit-learn provides include:

Regression, including Linear and Logistic Regression **Classification**, including K-Nearest Neighbors

Clustering, including K-Means and K-Means++
Model selection

Preprocessing, including Min-Max Normalization

Nltk- NLTK is one of the leading platforms for working with human language data and Python, the module NLTK is used for natural language processing. NLTK is literally an acronym for Natural Language Toolkit.

Used for sentimental analysis to remove white spaces and create corpus and then apply **Naïve bayes Classifier** to make out positive and negative reviews.

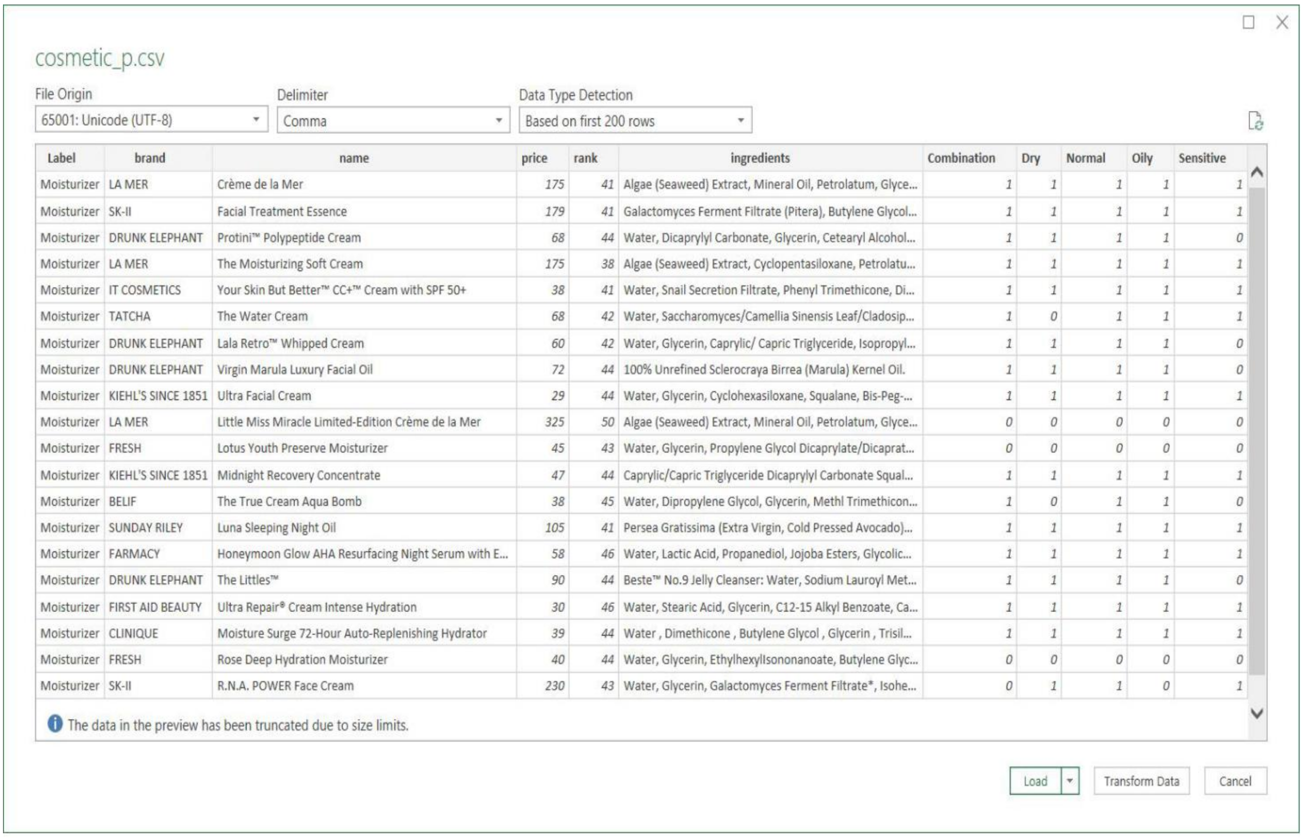
Used for sentimental analysis and many more.

DATASET

1. Survey dataset

	Skin_type	age	Gender	Imp_	Mode_of_buying	Purchase_recommendation	use_san	choosing_product	Face_wash	Face_cream	Body_lotion	average_spend	skin_problems	Ayurvedic_Pro
5	Oily Skin	23	Male	8	Pharmacy	Google search	no	Product reviews	Garnier Men acne fight face wash	No	Nivea Nourishing Lotion Body Milk	less than 200	1	
6	Dry Skin	21	Female	5	Pharmacy	Friends or Family	yes	Product reviews	Himalaya Herbals Purifying Neem Face Wash	Fair and Lovely	Vaseline Intensive Care Deep Restore Body Lotion	500-1000	1	
7	Normal Skin	49	Female	8	Pharmacy	Youtube gurus	no	Lower price rate	Medimix	Vicco Turmeric cream	Vaseline Intensive Care Deep Restore Body Lotion	less than 200	0	
8	Normal Skin	21	Male	9	Pharmacy	Friends or Family	yes	Brand recognition	Garnier Men acne fight face wash	Patanjali face cream	Vaseline Intensive Care Deep Restore Body Lotion	less than 200	0	
9	Normal Skin	21	Female	6	from store	Renowned makeup artists	no	Product reviews	Himalaya Herbals Purifying Neem Face Wash	Ponds white	Nivea Nourishing Lotion Body Milk	500-1000	0	
10	Normal Skin	23	Male	1	Online	Friends or Family	no	Product reviews	Garnier Men power white double action Face was	Garnier Men power white	Lotus Herbals Safe Sun UV-Protect Body Lotion	200 -500	1	
11	Oily Skin	22	Female	8	from store	Youtube gurus	no	Product reviews	Glutafine facewash	Himalaya fairness cream	Nivea Nourishing Lotion Body Milk	less than 200	1	
12	Oily Skin	22	Female	6	from store	Friends or Family	no	Product reviews	Clean & Clear Foaming Face Wash	Lotus Herbals	Nivea Nourishing Lotion Body Milk	Doesn't use any	0	
13	Dry Skin	21	Female	7	Online	Renowned makeup artists	yes	Product reviews	Pond's white beauty spotless face wash	Vicco Turmeric cream	Vaseline Intensive Care Deep Restore Body Lotion	200 -500	0	
14	Dry Skin	22	Female	8	from store	Friends or Family	no	Brand recognition	Clean & Clear Foaming Face Wash	Ponds	Vaseline Intensive Care Deep Restore Body Lotion	200 -500	1	
15	Oily Skin	21	Female	8	Online	Television adds	no	Brand recognition	Ponds activated carbon	Vicco Turmeric cream	Johnson and Johnson	less than 200	0	
16	Combination Skir	23	Female	10	Online	Friends or Family	no	Product reviews	Body shop vitamin E	Jhonson and Johnson moisturiser	Nivea Nourishing Lotion Body Milk	200 -500	1	
17	Combination Skir	22	Female	10	from store	Youtube gurus	yes	Product reviews	Organic Harvest	Swiss Tempelle	Nivea Nourishing Lotion Body Milk	1000-2000	1	
18	Oily Skin	21	Female	9	from store	Friends or Family	no	Lower price rate	Himalaya Herbals Purifying Neem Face Wash	Vicco Turmeric cream	Parachute Advansed Coconut Milk Deep Nourish Body Lotio	less than 200	0	
19	Dry Skin	22	Female	8	from store	Friends or Family	no	Brand recognition	Himalaya Herbals Purifying Neem Face Wash	Lotus Herbals	Nivea Nourishing Lotion Body Milk	200 -500	1	
20	Combination Skir	21	Female	10	from store	Friends or Family	yes	Product reviews	Himalaya Herbals Purifying Neem Face Wash	Himalaya fairness cream	Nivea Nourishing Lotion Body Milk	500-1000	0	
21	Dry Skin	23	Female	5	from store	Friends or Family	no	Product reviews	Himalaya Herbals Purifying Neem Face Wash	No	Parachute Advansed Coconut Milk Deep Nourish Body Lotio	200 -500	0	
22	Combination Skir	21	Female	6	Pharmacy	Friends or Family	no	Product reviews	Himalaya Herbals Purifying Neem Face Wash	Vicco Turmeric cream	Nivea Nourishing Lotion Body Milk	less than 200	0	
23	Oily Skin	26	Female	10	Pharmacy	Friends or Family	yes	Brand recognition	Himalaya Herbals Purifying Neem Face Wash	Fair and Lovely	Vaseline Intensive Care Deep Restore Body Lotion	less than 200	1	
24	Not tested yet	17	Male	5	Online	Nothing	yes	Brand recognition	Medimix	Fair and Lovely	Nivea Nourishing Lotion Body Milk	Doesn't use any	1	
25	Oily Skin	22	Female	6	from store	Friends or Family	no	Product reviews	Nioglow	no	I use coconut milk and aloevera	less than 200	0	
26	Oily Skin	15	Female	10	from store	Friends or Family	yes	Product reviews	Himalaya Herbals Purifying Neem Face Wash	Himalaya fairness cream	Nivea Nourishing Lotion Body Milk	less than 200	0	
27	Oily Skin	19	Male	10	Pharmacy	Myself	no	Product reviews	Garnier Men acne fight face wash	Garnier Men power white	Vaseline Intensive Care Cocoa Glow Body Lotion	less than 200	0	
28	Dry Skin	19	Male	10	Pharmacy	Youtube gurus	no	Product reviews	Himalaya Herbals Purifying Neem Face Wash	Fair and Lovely	Vaseline Intensive Care Deep Restore Body Lotion	less than 200	0	
29	Dry Skin	21	Female	9	Pharmacy	Friends or Family	yes	Product reviews	Everyuth Naturals Moisturizing Fruit Face Wash	Sunbun lotion (As I am not suitable to any cream	Vaseline Intensive Care Deep Restore Body Lotion	200 -500	1	
30	Oily Skin	30	Male	6	from store	Friends or Family	no	Packaging	Medimix	Patanjali face cream	No	less than 200	0	
31	Normal Skin	55	Male	3	Pharmacy	Friends or Family	no	Lower price rate	no	Vicco Turmeric cream	Vaseline Intensive Care Deep Restore Body Lotion	Doesn't use any	0	
32	Combination Skir	40	Male	5	from store	Television adds	yes	Brand recognition	Garnier Men acne fight face wash	Fair and Lovely	Parachute Advansed Coconut Milk Deep Nourish Body Lotio	200 -500	1	
33	Oily Skin	23	Male	10	Pharmacy	Friends or Family	yes	Brand recognition	Pears	Vicco Turmeric cream	Dove Purely Pampering Nourishing Body Lotion	200 -500	0	
34	Oily Skin	21	Female	10	Pharmacy	Friends or Family	yes	Brand recognition	Himalaya Herbals Purifying Neem Face Wash	Lotus Herbals	Not using	less than 200	0	
35	Normal Skin	25	Male	10	Pharmacy	Friends or Family	no	Product reviews	Nivea Total Face Clean Up	Garnier Men power white	Nivea Nourishing Lotion Body Milk	200 -500	0	
36	Combination Skir	23	Female	10	Online	Youtube gurus	yes	Product reviews	Himalaya Herbals Purifying Neem Face Wash	Biotique morning nectar	Vaseline Intensive Care Deep Restore Body Lotion	200 -500	1	
37	Dry Skin	23	Female	9	Online	Renowned makeup artists	yes	Brand recognition	Nivea Total Face Clean Up	Patanjali face cream	Vaseline Intensive Care Deep Restore Body Lotion	200 -500	1	
38	Dry Skin	29	Male	5	Pharmacy	Youtube gurus	no	Product reviews	Garnier Men acne fight face wash	Patanjali face cream	Dove Purely Pampering Nourishing Body Lotion	less than 200	0	
39	Oily Skin	22	Male	10	from store	Friends or Family	no	Brand recognition	Medimix	Vicco Turmeric cream	Parachute Advansed Coconut Milk Deep Nourish Body Lotio	less than 200	0	
40	Oily Skin	31	Female	10	Online	Youtube gurus	no	Product reviews	Clean & Clear Foaming Face Wash	Himalaya fairness cream	Nivea Nourishing Lotion Body Milk	200 -500	0	
41	Combination Skir	22	Female	10	from store	Friends or Family	no	Product reviews	Nioglow	Pears	Nivea Nourishing Lotion Body Milk	200 -500	1	
42	Combination Skir	22	Female	10	from store	Friends or Family	no	Product reviews	Nioglow	Sunbun lotion	Nivea Nourishing Lotion Body Milk	200 -500	1	
43	Oily Skin	26	Female	9	Online	Friends or Family	yes	Product reviews	Clean & Clear Foaming Face Wash	Fair and Lovely	Nivea Nourishing Lotion Body Milk	500-1000	1	
44	Normal Skin	30	Male	7	Pharmacy	Television adds	yes	Product reviews	Don't use	Patanjali face cream	Don't use	less than 200	0	
45	Dry Skin	18	Male	5	from store	Friends or Family	no	Don't use	Don't use	Don't use	no	Doesn't use any	0	
46	Oily Skin	35	Female	9	Online	Youtube gurus	no	Brand recognition	Nivea Total Face Clean Up	Vicco Turmeric cream	Vaseline Intensive Care Cocoa Glow Body Lotion	1000-2000	1	
47	Normal Skin	29	Male	4	Pharmacy	Television adds	no	Lower price rate	Nivea mens Face wash	Don't use	no	200 -500	0	
48	Normal Skin	29	Male	4	Pharmacy	Television adds	no	Lower price rate	Nivea mens Face wash	Don't use	no	200 -500	0	
49	Not tested yet	25	Male	4	from store	Television adds	no	Lower price rate	Garnier Men acne fight face wash	Garnier Men power white	no	200 -500	0	
50	Oily Skin	40	Female	10	Online	Renowned makeup artists	yes	Brand recognition	Nivea Total Face Clean Up	Fair and Lovely	Nivea Nourishing Lotion Body Milk	1000-2000	0	
51	Dry Skin	39	Female	8	from store	Television adds	yes	Product reviews	Ayurvedic Multanimitti facewash	Vicco Turmeric cream	Nivea Nourishing Lotion Body Milk	200 -500	1	
52	Oily Skin	40	Male	5	Pharmacy	Friends or Family	no	Brand recognition	no	Home remidy	Vaseline Intensive Care Cocoa Glow Body Lotion	less than 200	0	

2. Product details dataset



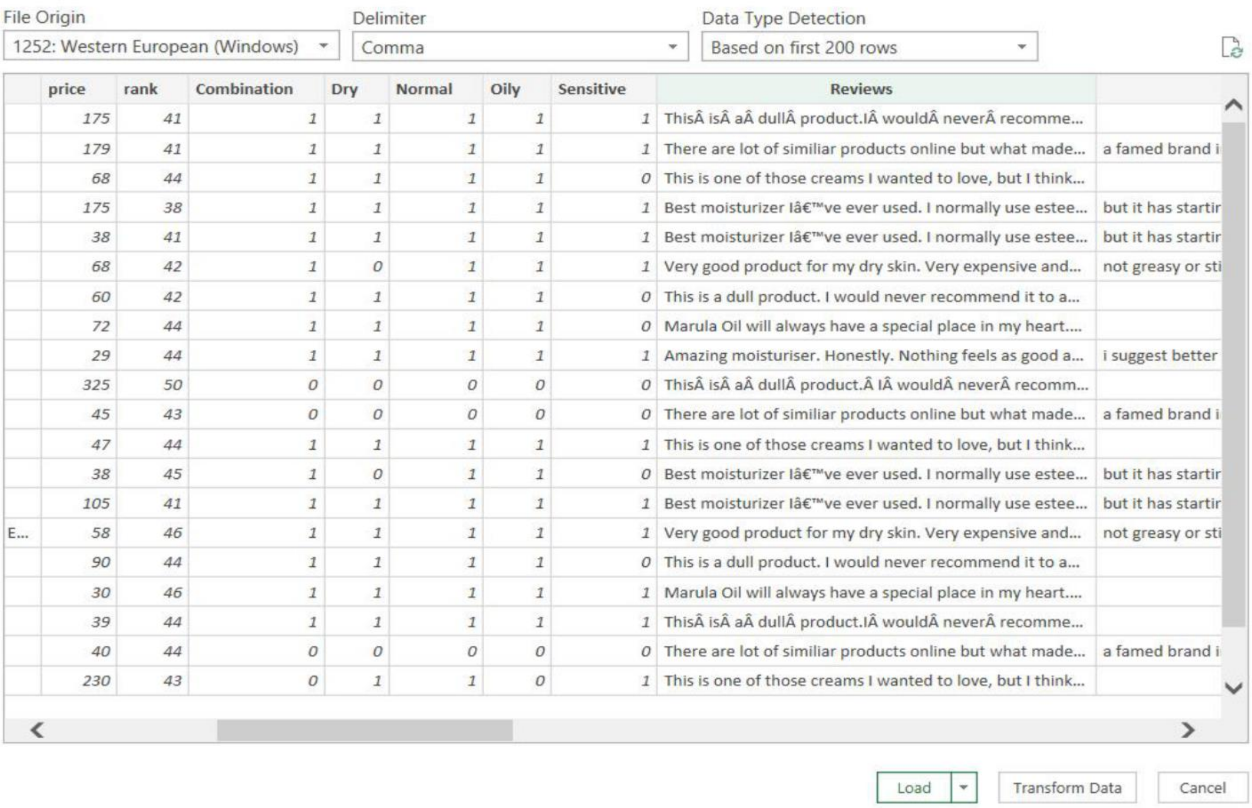
cosmetic_p.csv

File Origin: 65001: Unicode (UTF-8) | Delimiter: Comma | Data Type Detection: Based on first 200 rows

Label	brand	name	price	rank	ingredients	Combination	Dry	Normal	Oily	Sensitive
Moisturizer	LA MER	Crème de la Mer	175	41	Algae (Seaweed) Extract, Mineral Oil, Petrolatum, Glyce...	1	1	1	1	1
Moisturizer	SK-II	Facial Treatment Essence	179	41	Galactomyces Ferment Filtrate (Pitera), Butylene Glycol...	1	1	1	1	1
Moisturizer	DRUNK ELEPHANT	Protini™ Polypeptide Cream	68	44	Water, Dicaprylyl Carbonate, Glycerin, Cetearyl Alcohol...	1	1	1	1	0
Moisturizer	LA MER	The Moisturizing Soft Cream	175	38	Algae (Seaweed) Extract, Cyclopentasiloxane, Petrolatu...	1	1	1	1	1
Moisturizer	IT COSMETICS	Your Skin But Better™ CC+™ Cream with SPF 50+	38	41	Water, Snail Secretion Filtrate, Phenyl Trimethicone, Di...	1	1	1	1	1
Moisturizer	TATCHA	The Water Cream	68	42	Water, Saccharomyces/Camellia Sinensis Leaf/Cladosip...	1	0	1	1	1
Moisturizer	DRUNK ELEPHANT	Lala Retro™ Whipped Cream	60	42	Water, Glycerin, Caprylic/ Capric Triglyceride, Isopropyl...	1	1	1	1	0
Moisturizer	DRUNK ELEPHANT	Virgin Marula Luxury Facial Oil	72	44	100% Unrefined Sclerocraya Birrea (Marula) Kernel Oil.	1	1	1	1	0
Moisturizer	KIEHL'S SINCE 1851	Ultra Facial Cream	29	44	Water, Glycerin, Cyclohexasiloxane, Squalane, Bis-Peg-...	1	1	1	1	1
Moisturizer	LA MER	Little Miss Miracle Limited-Edition Crème de la Mer	325	50	Algae (Seaweed) Extract, Mineral Oil, Petrolatum, Glyce...	0	0	0	0	0
Moisturizer	FRESH	Lotus Youth Preserve Moisturizer	45	43	Water, Glycerin, Propylene Glycol Dicaprylate/Dicaprat...	0	0	0	0	0
Moisturizer	KIEHL'S SINCE 1851	Midnight Recovery Concentrate	47	44	Caprylic/Capric Triglyceride Dicaprylyl Carbonate Squal...	1	1	1	1	1
Moisturizer	BELIF	The True Cream Aqua Bomb	38	45	Water, Dipropylene Glycol, Glycerin, Methl Trimethicon...	1	0	1	1	0
Moisturizer	SUNDAY RILEY	Luna Sleeping Night Oil	105	41	Persea Gratissima (Extra Virgin, Cold Pressed Avocado)...	1	1	1	1	1
Moisturizer	FARMACY	Honeymoon Glow AHA Resurfacing Night Serum with E...	58	46	Water, Lactic Acid, Propanediol, Jojoba Esters, Glycolic...	1	1	1	1	1
Moisturizer	DRUNK ELEPHANT	The Littles™	90	44	Beste™ No.9 Jelly Cleanser: Water, Sodium Lauroyl Met...	1	1	1	1	0
Moisturizer	FIRST AID BEAUTY	Ultra Repair® Cream Intense Hydration	30	46	Water, Stearic Acid, Glycerin, C12-15 Alkyl Benzoate, Ca...	1	1	1	1	1
Moisturizer	CLINIQUE	Moisture Surge 72-Hour Auto-Replenishing Hydrator	39	44	Water , Dimethicone , Butylene Glycol , Glycerin , Trisil...	1	1	1	1	1
Moisturizer	FRESH	Rose Deep Hydration Moisturizer	40	44	Water, Glycerin, Ethylhexylisononanoate, Butylene Glyc...	0	0	0	0	0
Moisturizer	SK-II	R.N.A. POWER Face Cream	230	43	Water, Glycerin, Galactomyces Ferment Filtrate*, Isohe...	0	1	1	0	1

The data in the preview has been truncated due to size limits.

Load | Transform Data | Cancel



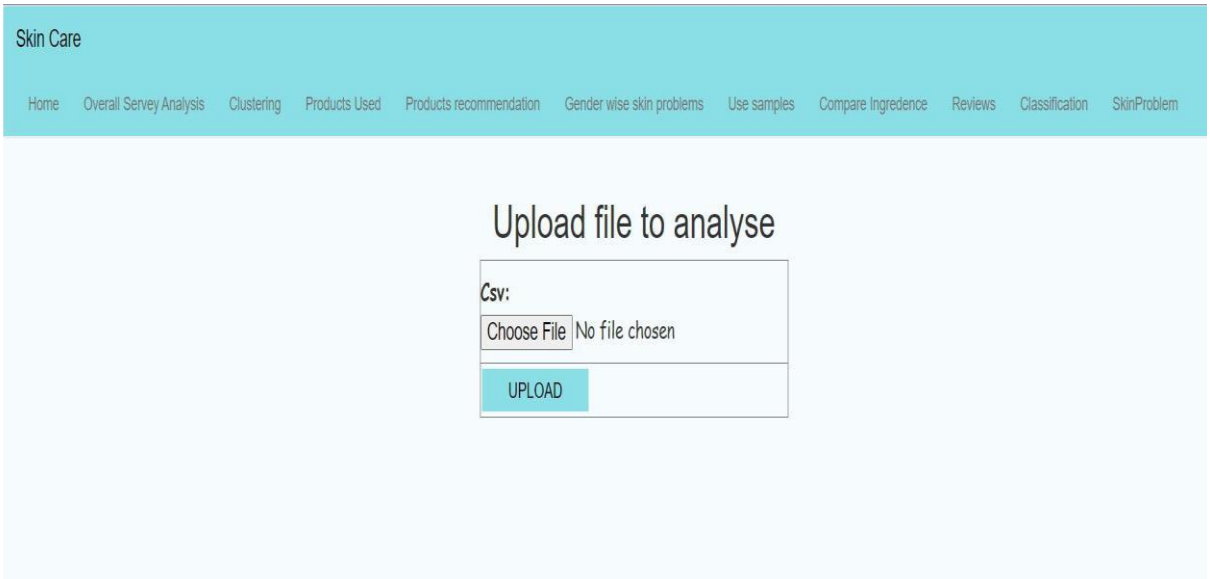
File Origin: 1252: Western European (Windows) | Delimiter: Comma | Data Type Detection: Based on first 200 rows

	price	rank	Combination	Dry	Normal	Oily	Sensitive	Reviews	
	175	41	1	1	1	1	1	ThisÂ isÂ aÂ dullÂ product.IÂ wouldÂ neverÂ recomme...	
	179	41	1	1	1	1	1	There are lot of similiar products online but what made...	a famed brand i
	68	44	1	1	1	1	0	This is one of those creams I wanted to love, but I think...	
	175	38	1	1	1	1	1	Best moisturizer Iâ€™ve ever used. I normally use estee...	but it has startir
	38	41	1	1	1	1	1	Best moisturizer Iâ€™ve ever used. I normally use estee...	but it has startir
	68	42	1	0	1	1	1	Very good product for my dry skin. Very expensive and...	not greasy or sti
	60	42	1	1	1	1	0	This is a dull product. I would never recommend it to a...	
	72	44	1	1	1	1	0	Marula Oil will always have a special place in my heart....	
	29	44	1	1	1	1	1	Amazing moisturiser. Honestly. Nothing feels as good a...	i suggest better
	325	50	0	0	0	0	0	ThisÂ isÂ aÂ dullÂ product.Â IÂ wouldÂ neverÂ recomm...	
	45	43	0	0	0	0	0	There are lot of similiar products online but what made...	a famed brand i
	47	44	1	1	1	1	1	This is one of those creams I wanted to love, but I think...	
	38	45	1	0	1	1	0	Best moisturizer Iâ€™ve ever used. I normally use estee...	but it has startir
	105	41	1	1	1	1	1	Best moisturizer Iâ€™ve ever used. I normally use estee...	but it has startir
E...	58	46	1	1	1	1	1	Very good product for my dry skin. Very expensive and...	not greasy or sti
	90	44	1	1	1	1	0	This is a dull product. I would never recommend it to a...	
	30	46	1	1	1	1	1	Marula Oil will always have a special place in my heart....	
	39	44	1	1	1	1	1	ThisÂ isÂ aÂ dullÂ product.IÂ wouldÂ neverÂ recomme...	
	40	44	0	0	0	0	0	There are lot of similiar products online but what made...	a famed brand i
	230	43	0	1	1	0	1	This is one of those creams I wanted to love, but I think...	

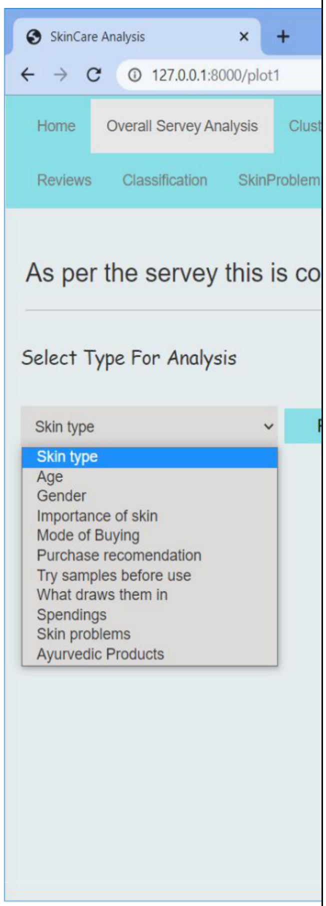
Load | Transform Data | Cancel

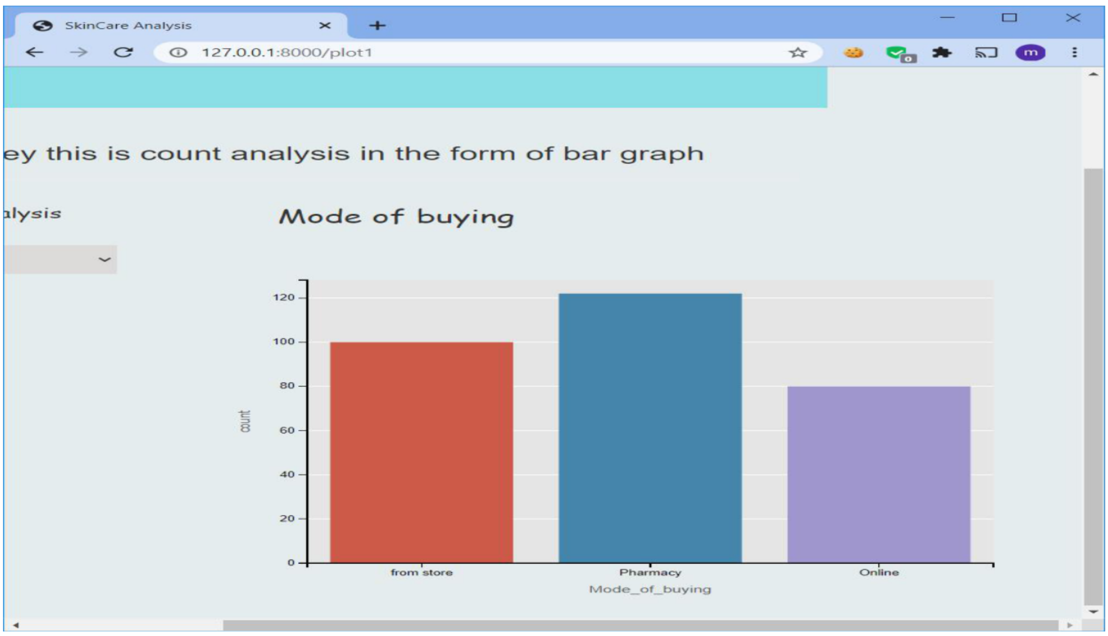
ANALYSIS

1. Upload file of survey data set.



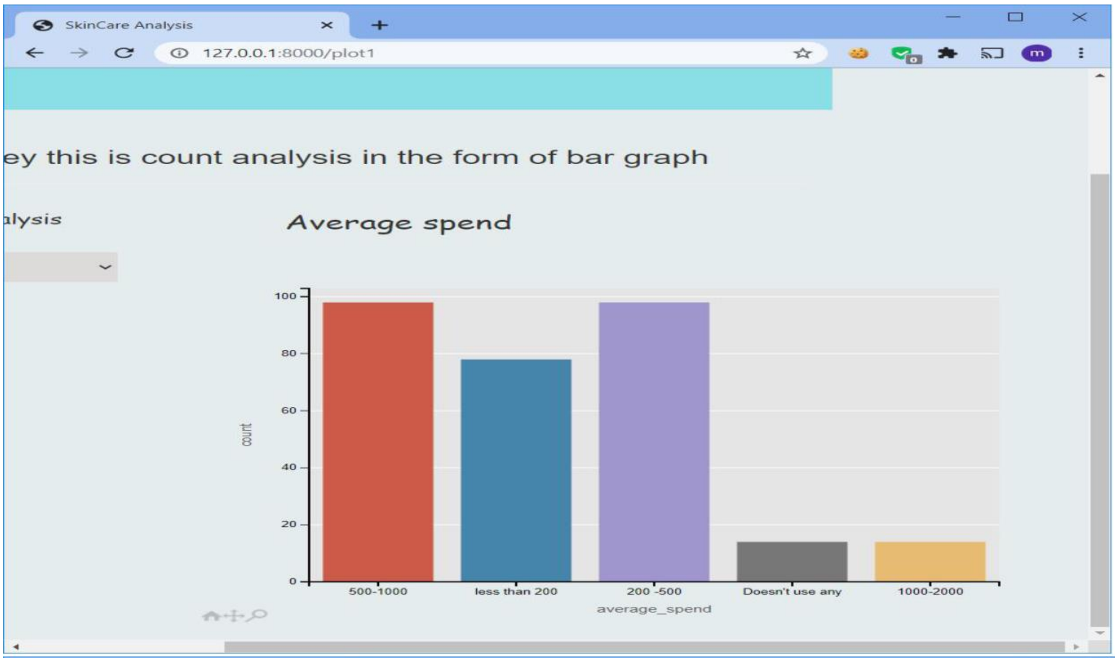
2. Getting initial analysis of dataset.

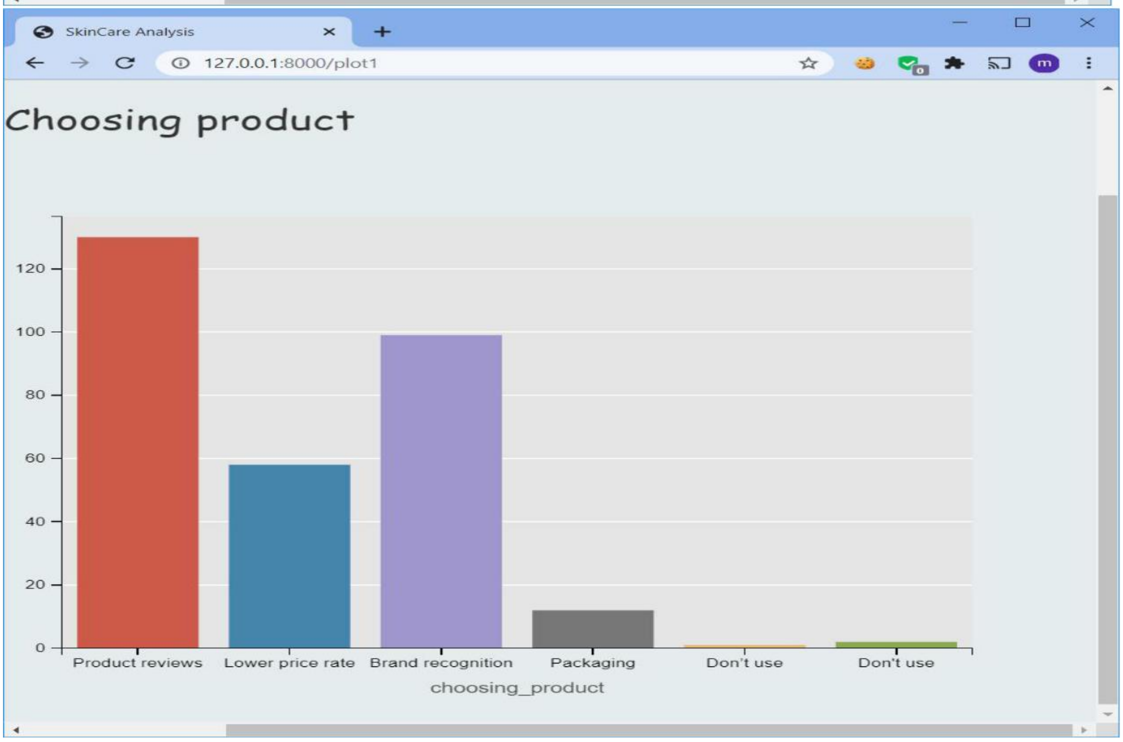




3. Clusters according to the importance of skin and annual spending on products as per the age

Active age group is between 20-30…

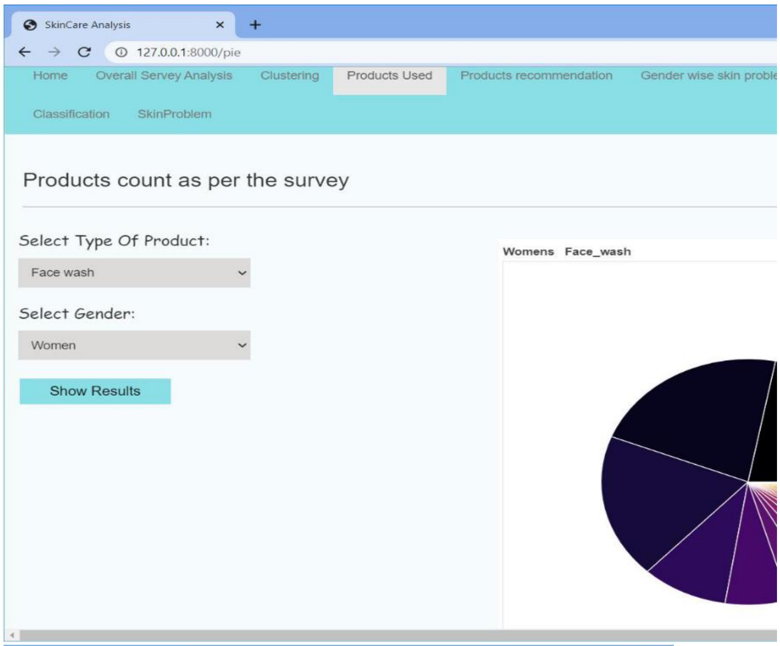


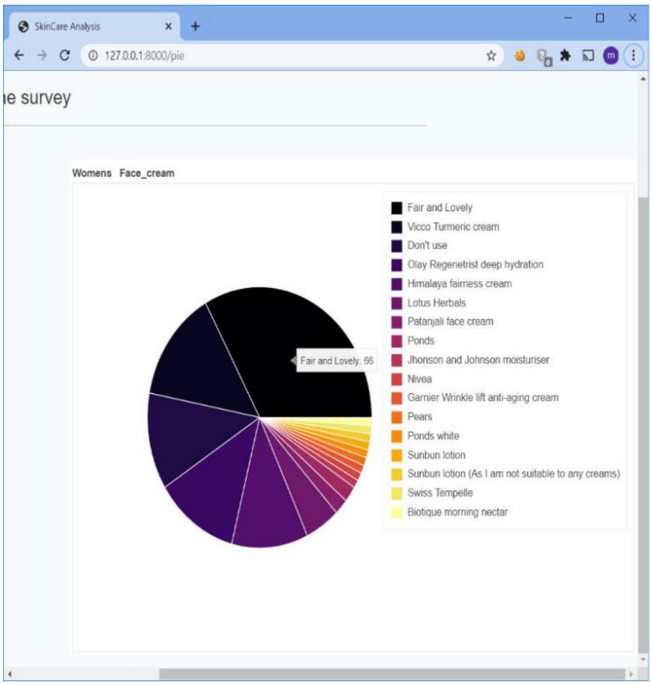


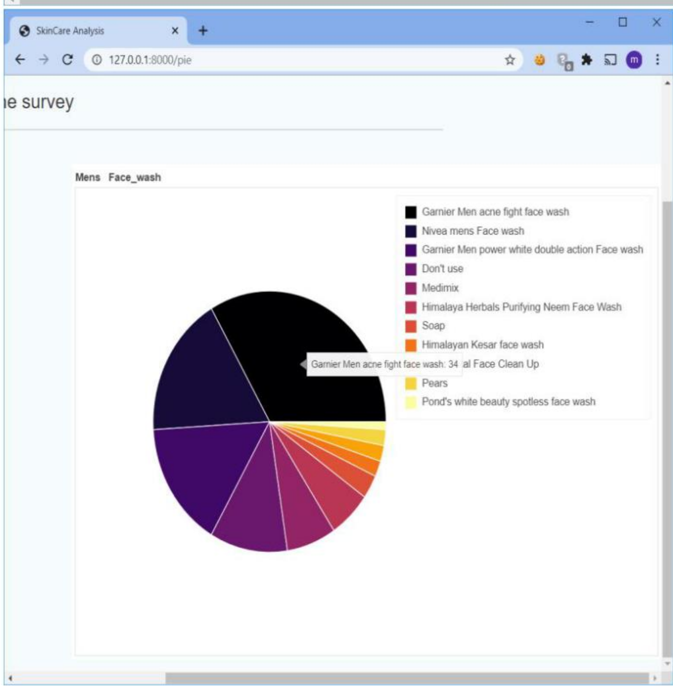
4. Which face wash, face cream, body lotion is used to most and the least for men and women

Women-fash wash- Himalaya Herbals Purifying Neem Face Wash, Face cream- Fair and lovely, body lotion- Vaseline Intensive Care

Men – fash wash- Garnier men acne fight, Face cream- Garnier men power white, body lotion – Don’t use



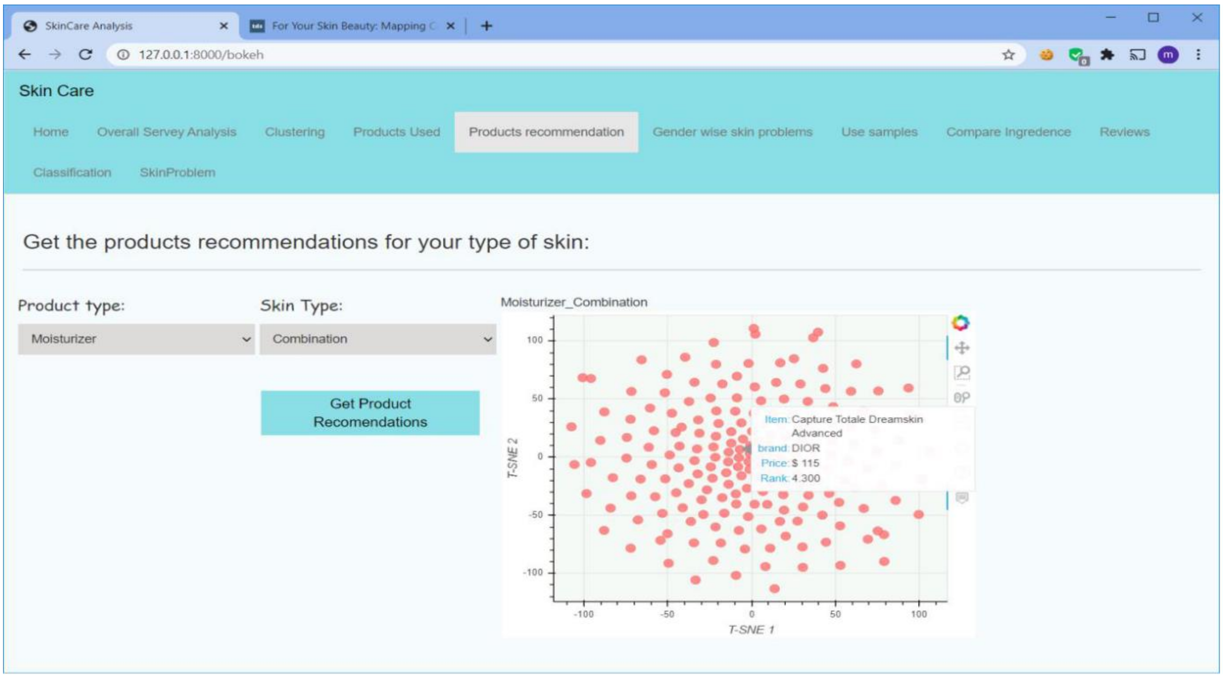


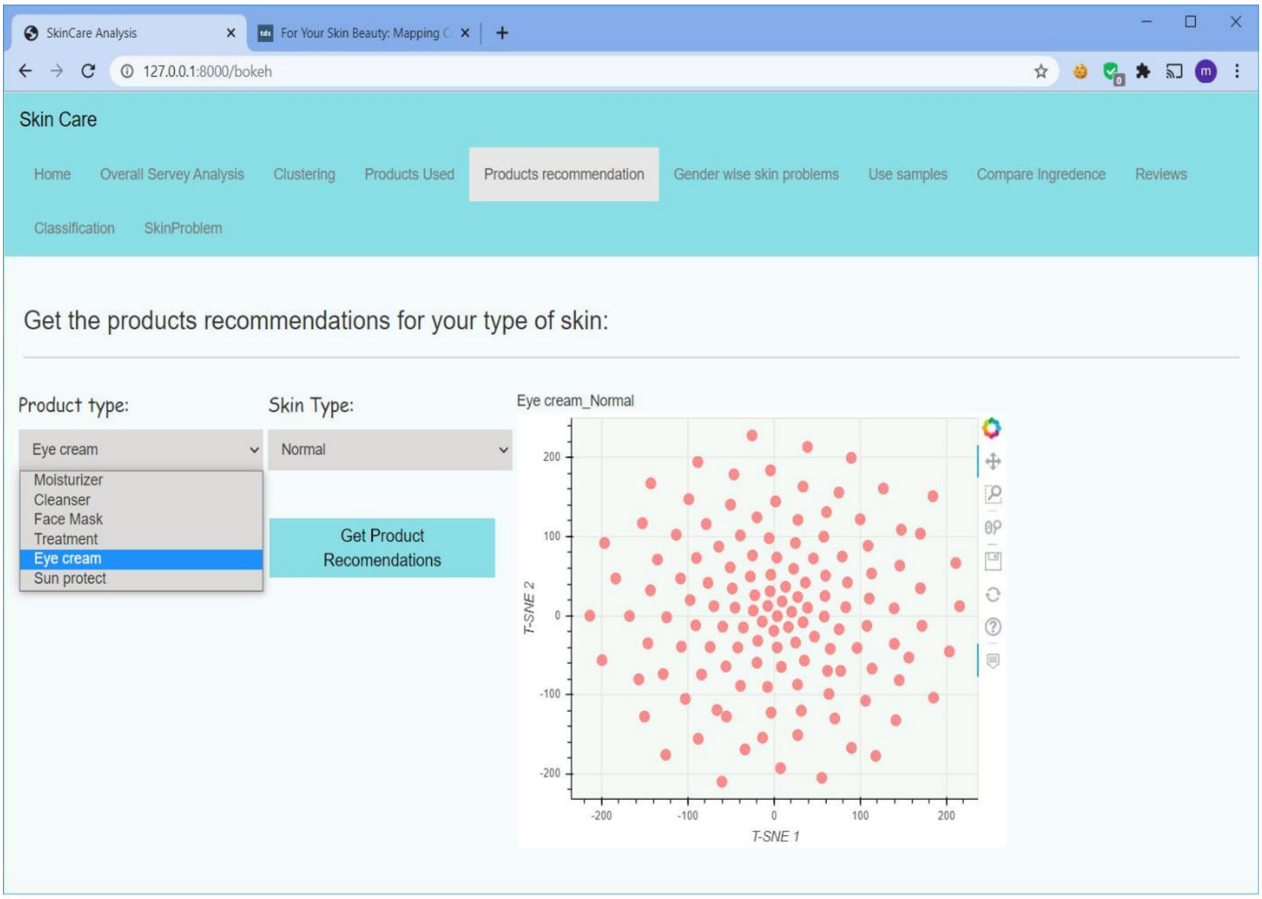


5. Recommendation system by comparing the ingredients among products: Skin type: Combination Skin, Normal Skin , Oily skin, Dry skin

Product type: Moisturizer, Cleanser, Face Mask, Eye cream, Treatment, Sun protection,

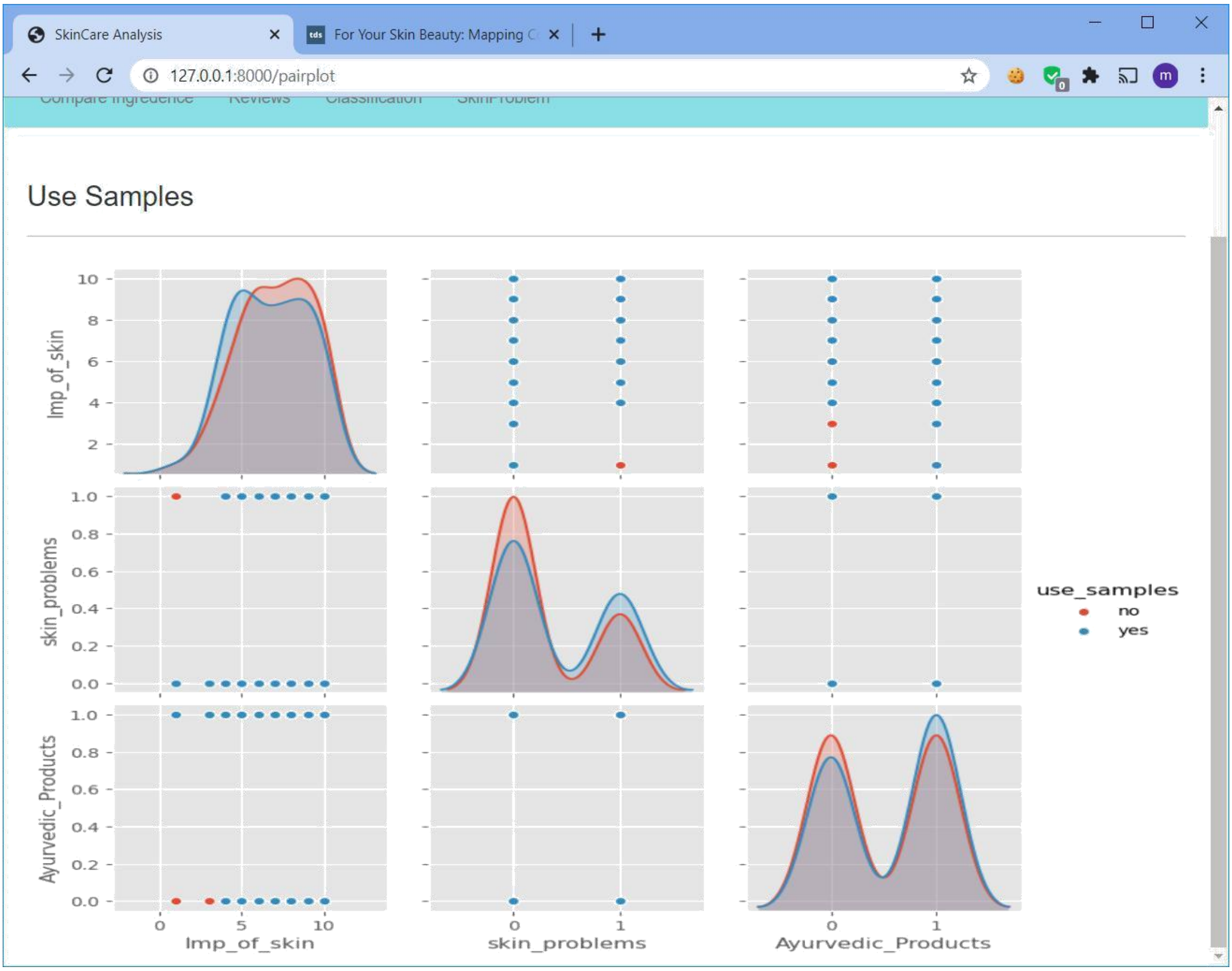
One in middle and close to many other products is best for combination skin type and moisturizer and vice versa.



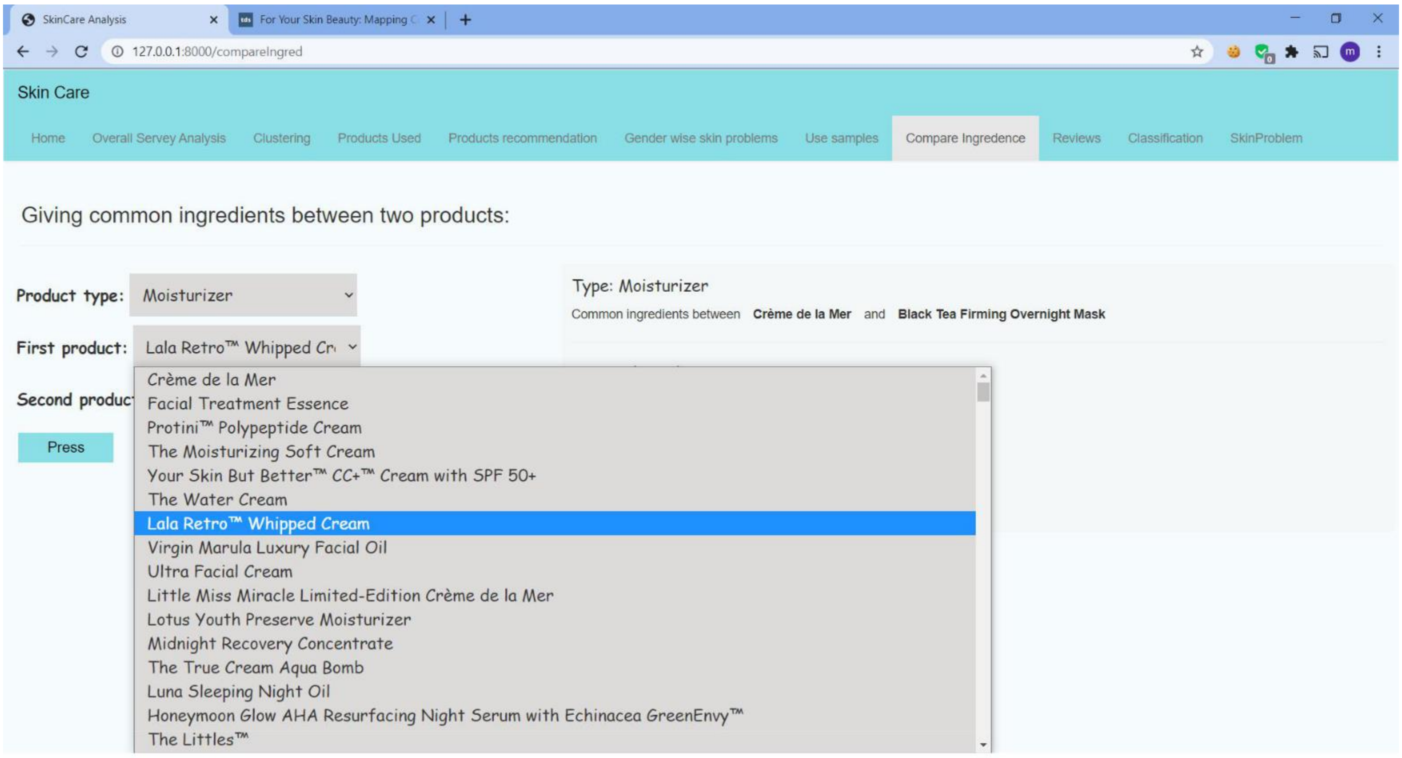


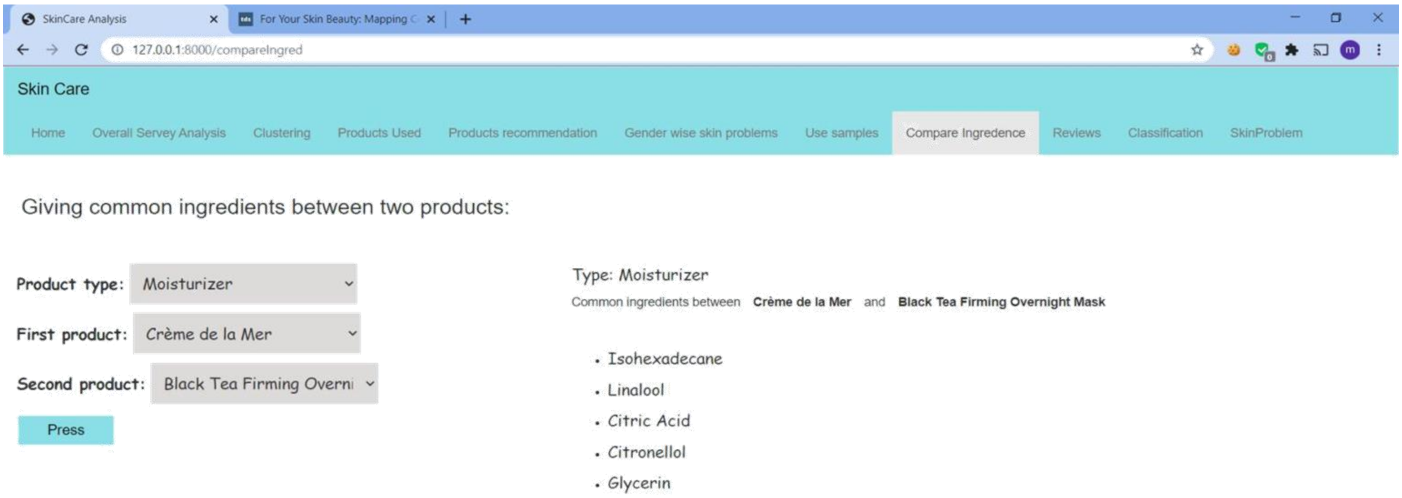
6. How people looks after their skin as per the gender and whether they prefer using samples before use and where they are facing any skin problem.

7. Visualization of skin problems and whether they prefer Ayurvedic products or not as per skin type

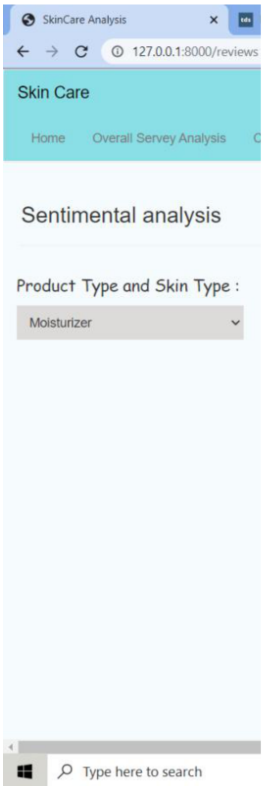


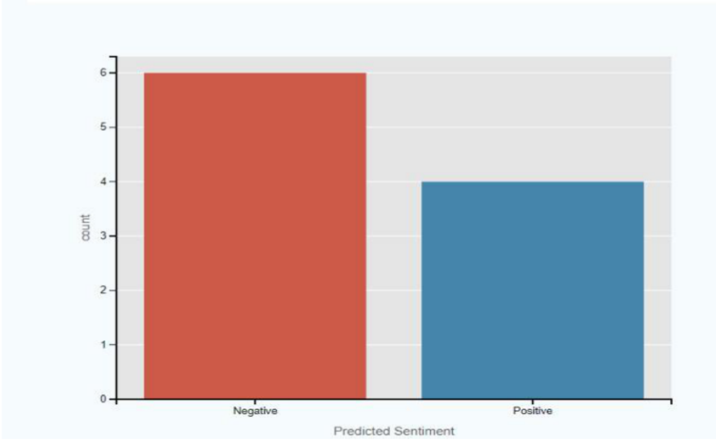
8. What are the common ingredients between two products? These are the common ingredients in these two products.





9. Sentimental analysis of problem selected in dropdown





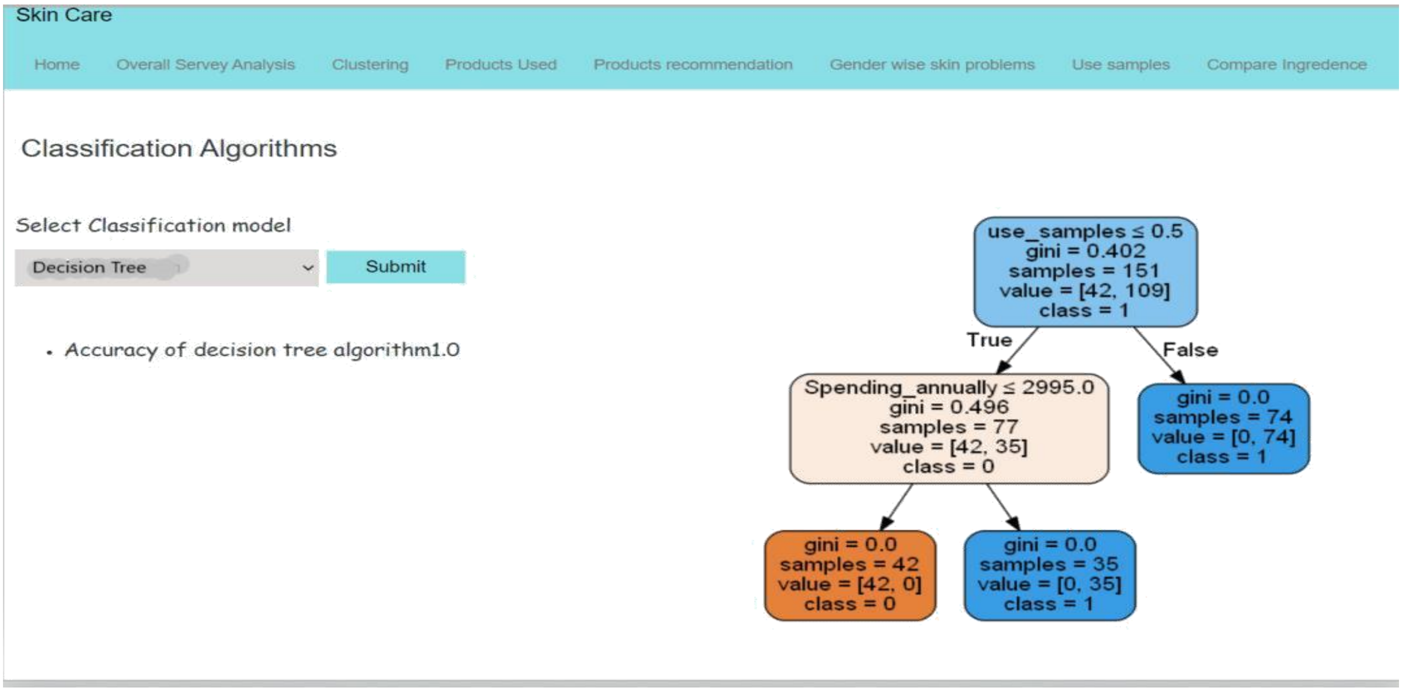
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL 1: python

```
dull product would never recommend anyon  -----  Negative
ingred pretti great felt effect  -----  Positive
product pathet  -----  Negative
face start glow amaz product  -----  Positive
worst product ever  -----  Negative
best cream dri skin  -----  Negative
skin start glow fantast  -----  Positive
would like recom sure  -----  Negative
read mani review product say receiv expir product want clarifi date written front side crimp tube expiri date manufactur date e
atch number written back side crimp tube pleas check properli give neg feedback hamper reput seller  -----  Negative
job well  -----  Positive
                                          Review Predicted Sentiment  Probability
0                            This is a dull product            Negative         0.85
1                   The ingredents are pretty great            Positive         0.85
2                          This product is pathetic            Negative         0.63
3                              Face started glowing            Positive         0.65
4                                worst product ever            Negative         0.86
5                             Best cream for dry skin          Negative         0.65
6                  my skin starting glowing fantastic          Positive         0.68
7               Would like to recomment this for sure          Negative         0.52
8  I have read many reviews of this product who s...           Negative         0.85
9                         But this does the job well           Positive         0.61
countt  Negative    6
Positive    4
```

10. Classification of the problem for predicting whether the skin problem is possible as per the gender, age ,skin type ,importance of skin, spending

It shows the false positive and true positive rate and here in this case it is giving the 78 percent of accuracy for the data

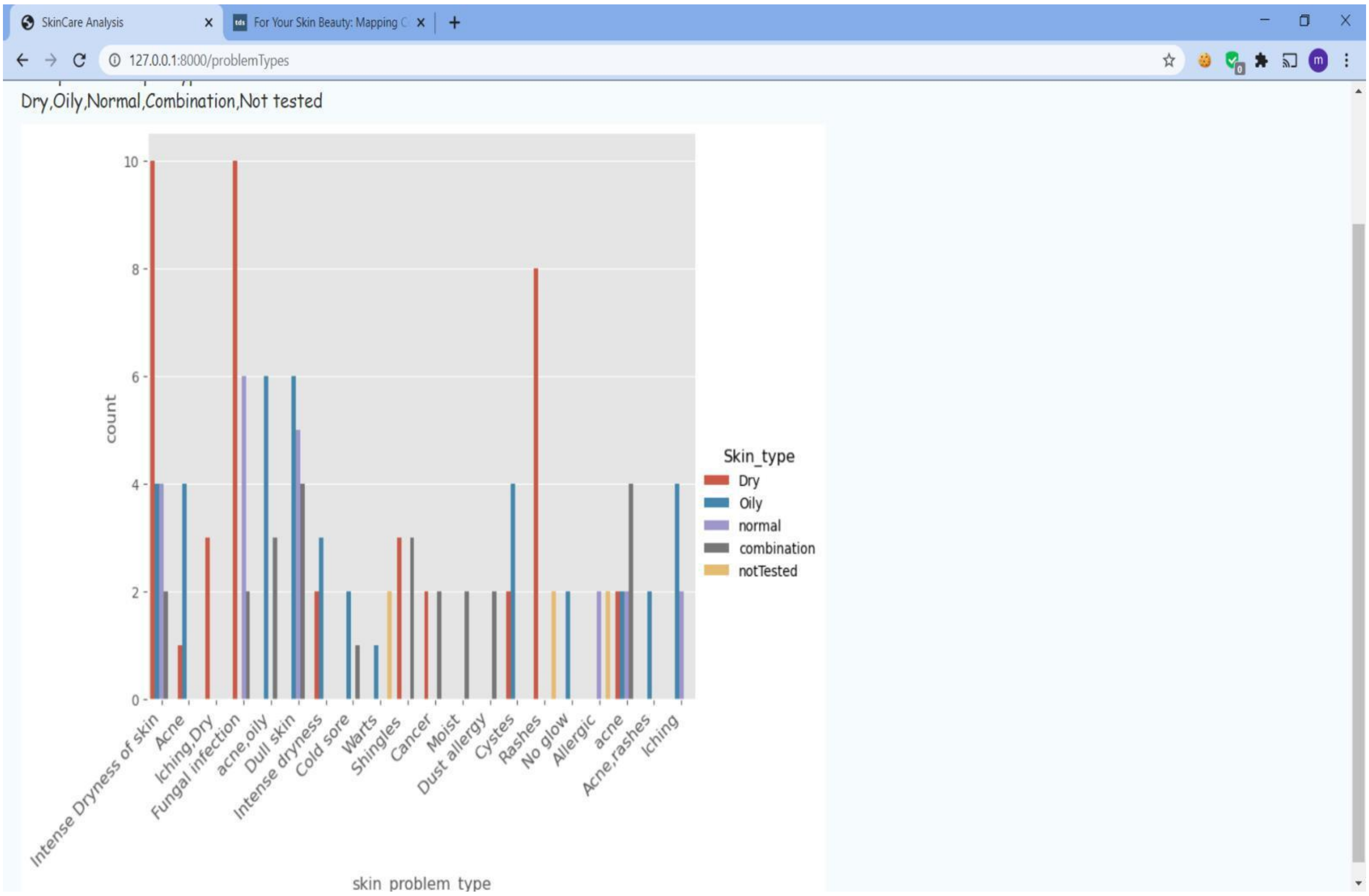
This concludes that the skin problems are accurate as per the spending and how good they handle their skin, whether they use samples or not and ratings.



It shows the accuracy is 100 %.

So the people under 30 years and the one who spends a lot and one who do not have skin problems and females are tend to buy new launch of products.

11. What are the types of skin problems people are facing as per the skin type.



This analysis shows that Dust allergy is suffered most by Combination skin people where Rashes by dry skin people and so on

CONCLUSION

Analyzed all the possibilities in skincare data. There were some null data's which can create barrier in analytics. So, data cleaning is an initial and important part for data analysis.

After applying some algorithms the accuracy can out good of the model. But if the dataset values are much biased accuracy will reduce.

Recommendation system is made possible. This would be helpful for the new product users.

Sentimental analysis on product reviews is done. In which many a times the review has false positive or true negative values even after removing all the stop words from the data, this is major challenging to find the exact prediction value, this is a drawback .

This sentimental analysis will help users who are looking for quick review of any product.

Applied classification algorithms to classify whether person will buy skin product or not buy.

Also, making a decision on it by using Decision tree algorithm.

FUTURE ENHANCEMENT

In future can work with Real-time online data and can also store data in aws s3.

There will be analysis on the various types of skin diseases. Suggest cure and medications recommendation.

According to that will also get connected to various dermatologists by using more high level packages in python

Also, can apply the deep learning concept in this which will automatically analyse what person wants to find with some data related to person.

The products can be recommended according to the users need. Also add E-commerce feature in this.

PROGRAM CODE

```
import mpld3
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
import seaborn as sns
import pandas as pd
from matplotlib.backends.backend_agg import
FigureCanvasAgg import base64
from io import BytesIO
from mpld3.plugins import PointLabelTooltip
from mpld3 import fig_to_html, plugins
from django.shortcuts import render, redirect
from django.http import HttpResponse
from django.core.files.storage import FileSystemStorage
from .forms import ProductForm, CompareIngredientsForm
from .models import product, CompareIngredients, get_product_names
import json
import matplotlib
import numpy as np
from bokeh.plotting import figure, output_file, show, save
from bokeh.models import ColumnDataSource, HoverTool, Select, Paragraph, TextInput
from bokeh.layouts import widgetbox, column, row matplotlib.use("Agg")

plt.style.use('ggplot')
prod = product.objects.all()
a = []
for p in prod:
  a.append(str(p.csv))
input = a.pop()


# Create your views here.


def home(request):
  if request.method == "POST":
    form = ProductForm(request.POST, request.FILES)
    if form.is_valid():
      form.save()
      return redirect("plot1")
  else:
    form = ProductForm()
  return render(request, "home.html", {"form": form})


def allplots(request):
  return render(request, 'allplots.html')


def addcsv(request):
```

```python
    if request.method == "POST":
        form = ProductForm(request.POST, request.FILES)
        if form.is_valid():
            form.save()
            return redirect("allplots")
    else:
        form = ProductForm()
    return render(request, "addcsv.html", {"form": form})


def plot1(request):
    """pii = product.objects.filter(csv="product/skinfinal.csv")
    a1 = []
    for p in pii:
        print("filtered csv", str(p.csv))
        # a1.append(str(p.csv))

    prod=product.objects.all()"""
    if request.method == "POST":

        # getting the values
        colm = request.POST["col"]
        type = colm.replace("_", " ").capitalize()

        prod = product.objects.all()
        a = []
        for p in prod:
            a.append(str(p.csv))
        input = a.pop()
        print("typee ", colm)
        path = "D:/Divya/DjangoProject/skincareproject/media/" + input
        df = pd.read_csv(path)
        fig, ax = plt.subplots()
        count = df[colm].value_counts()
        print("countt ", count)

        p = sns.countplot(x=colm, data=df)
        plt.style.use('ggplot')
        plot = fig.savefig(
            'D:\\Divya\\DjangoProject\\skincareproject\\static\\my_plot2.png')

        plot = fig_to_html(fig)

        return render(request, 'plot1.html', {'plot': [plot], "plot1_page": "active", "type": type})

    return render(request, 'plot1.html', {"plot1_page": "active"})


def cluster(request):
    inputs = {"title": "As per the servey this is count analysis in the form of bar graph",
              "subtitle": "Select Type For Analysis",
              "options": {"Imp_of_skin": "Importance of skin", "Spending_annually": "Spending Annually"}}
    #   "names": ["Imp_of_skin", "Spending_annually"], "values": ["Importance of skin", "Spending Annually"]}}
```

```
    # inputs = ['hello', 'world']
    if request.method == "POST":

        # getting the values
        colm = request.POST["type"]
        type = colm.replace("_", " ").capitalize()

        prod = product.objects.all()
        a = []
        for p in prod:
            a.append(str(p.csv))
        input = a.pop()

        path = "D:/Divya/DjangoProject/skincareproject/media/" + input
        data = pd.read_csv(path)
        df = pd.DataFrame(data, columns=['age', colm])
        kmeans = KMeans(n_clusters=5).fit(df)
        centroids = kmeans.cluster_centers_

        fig, ax = plt.subplots()
        ax.scatter(df['age'], df[colm],
                   c=kmeans.labels_.astype(float), s=50, alpha=0.5)
        ax.scatter(centroids[:, 0], centroids[:, 1], c='red', s=50)

        # plt.show()
        ax.set_xlabel('age')
        ax.set_ylabel(colm)
        ax.set_title('Clustering according to age group and '+colm, size=10)
        fig.savefig(
            'D:\\Divya\\DjangoProject\\skincareproject\\static\\cluster.png')
        # plugins.connect(fig, plugins.MousePosition(fontsize=14))
        # plot = mpld3.fig_to_html(fig)

        # plot = fig_to_html(fig)

        return render(request, 'cluster.html', {'plot': '/static/cluster.png', "cluster_page": "active", "type": type})
    return render(request, 'cluster.html', {"cluster_page": "active", "inputs": inputs})


def bokeh(request):
    if request.method == "POST":
        # cosmeticFileExist = True

        import os
        if os.path.isfile("D:\\Divya\\DjangoProject\\skincareproject\\media\\product\\cosmetic_TSNE.csv") =
= False:
            from sklearn.manifold import TSNE
            import pandas as pd
            import numpy as np
            # Load the data
            cosm_2 = pd.read_csv(
                'D:\\Divya\\DjangoProject\\Datasets\\cosmetic_p.csv')
```

```
# All possible combinations for the option choices
option_1 = cosm_2.Label.unique().tolist()
option_2 = cosm_2.columns[6:].tolist()

# defining a function embedding ingredients and decomposition at once
def my_recommender(op_1, op_2):

  df = cosm_2[cosm_2['Label'] == op_1][cosm_2[op_2] == 1]
  df = df.reset_index()

  # embedding each ingredients
  ingredient_idx = {}
  corpus =
  [] idx = 0

  for i in range(len(df)):
    ingreds = df['ingredients'][i]
    ingreds = ingreds.lower()
    tokens = ingreds.split(', ')
    corpus.append(tokens)
    for ingredient in tokens:
      if ingredient not in ingredient_idx:
        ingredient_idx[ingredient] = idx
        idx += 1

  # Get the number of items and tokens
  M = len(df)              # The number of the items
  N = len(ingredient_idx)    # The number of the ingredients

  # Initialize a matrix of zeros
  A = np.zeros(shape=(M, N))

  # Define the oh_encoder
  function def oh_encoder(tokens):
    x = np.zeros(N)
    for t in tokens:
      # Get the index for each ingredient
      idx = ingredient_idx[t]
      # Put 1 at the corresponding
      indices x[idx] = 1
    return x

  # Make a document-term
  matrix i = 0
  for tokens in corpus:
    A[i, :] = oh_encoder(tokens)
    i += 1

  # Dimension reduction with t-SNE
  model = TSNE(n_components=2, learning_rate=200)
  tsne_features = model.fit_transform(A)
```

```
        # Make X, Y columns
        df['X'] = tsne_features[:, 0]
        df['Y'] = tsne_features[:, 1]

        return df

    # Create the dataframe for all combinations
    df_all = pd.DataFrame()
    for op_1 in option_1:
        for op_2 in option_2:
            temp = my_recommender(op_1, op_2)
            temp['Label'] = op_1 + '_' + op_2
            df_all = pd.concat([df_all, temp])

    # Save the file
    df_all.to_csv('D:\\Divya\\DjangoProject\\skincareproject\\media\\product\\cosmetic_TSNE.csv',
                  encoding='utf-8-sig', index=False)

try:
    i1Type = request.POST["i1Type"]
    i1Skin = request.POST["i1Skin"]
    name = i1Type+"_"+i1Skin
    # import libraries
    import pandas as pd
    import numpy as np
    from sklearn.metrics.pairwise import cosine_similarity

    from bokeh.io import show, curdoc, output_notebook, push_notebook
    from bokeh.plotting import figure, output_file, show, save
    from bokeh.models import ColumnDataSource, HoverTool, Select, Paragraph, TextInput
    from bokeh.layouts import widgetbox, column, row from ipywidgets import interact


    output_file(
        'D:\\Divya\\DjangoProject\\skincareproject\\templates\\bokehOutput.html')
    path = "D:\\Divya\\DjangoProject\\skincareproject\\media\\product\\cosmetic_TSNE.csv"
    df = pd.read_csv(path)
    # cosmetic filtering options
    option_1 = ['Moisturizer', 'Cleanser', 'Treatment',
                'Face Mask', 'Eye cream', 'Sun protect']
    option_2 = ['Combination', 'Dry', 'Normal', 'Oily', 'Sensitive']
    # make a source and scatter bokeh plot
    source = ColumnDataSource(df)
    plot = figure(x_axis_label='T-SNE 1', y_axis_label='T-SNE 2',
                  width=500, height=400)
    plot.circle(x='X', y='Y', source=source,
                size=10, color='#FF7373', alpha=.8)

    plot.background_fill_color = "beige"
    plot.background_fill_alpha = 0.2
```

```
        # add hover tool
        hover = HoverTool(tooltips=[
            ('Item', '@name'),
            ('brand', '@brand'),
            ('Price', '$ @price'),
            ('Rank', '@rank')])
        plot.add_tools(hover)

        fig, ax = plt.subplots()

        # define the callback

        def update(op1=option_1[0], op2=option_2[1]):
            a_b = op1 + '_' + op2
            new_data = {
                'X': df[df['Label'] == a_b]['X'],
                'Y': df[df['Label'] == a_b]['Y'],
                'name': df[df['Label'] == a_b]['name'],
                'brand': df[df['Label'] == a_b]['brand'],
                'price': df[df['Label'] == a_b]['price'],
                'rank': df[df['Label'] == a_b]['rank'],
            }
            source.data = new_data

        # interact the plot with callback
        output_notebook()

        interact(update, op1=i1Type, op2=i1Skin)
        # show(plot, notebook_handle=True)
        save(plot, notebook_handle=True)
        fig.savefig(
            'D:\\Divya\\DjangoProject\\skincareproject\\static\\skin1.png')
        return render(request, 'bokeh.html', {'error': 'Successfully shown', "htmlPage": "bokehPage.html", 'bokeh': '
active', 'name': name})
    except:
        return render(request, 'bokeh.html', {'error': 'error', 'bokeh': 'active'})

  else:

      return render(request, 'bokeh.html', {'error': 'error', 'bokeh': 'active'})


def catplot(request):

  fig, ax = plt.subplots()

  prod = product.objects.all()
  a = []
  for p in prod:
      a.append(str(p.csv))
  input = a.pop()
  path = "D:/Divya/DjangoProject/skincareproject/media/" + input
```

```
    sc = pd.read_csv(path)
    g = sns.catplot(x="use_samples", hue="Gender", col="skin_problems", data=sc, kind="count",
                height=4, aspect=.7)

    # plt.plot(p)
    # ax.plot(p)
    # plot = fig_to_html(fig)
    g.savefig(
        'D:\\Divya\\DjangoProject\\skincareproject\\static\\catplot1.png')
    return render(request, 'catplot.html', {'plot': '/static/catplot1.png', 'catplot_page': 'active'})


def pairplot(request):

    fig, ax = plt.subplots()

    sc = pd.read_csv("D:\\Divya\\DjangoProject\\Datasets\\values.csv")
    sns_plot = sns.pairplot(sc, hue='use_samples', size=2.5)

    # plot = fig_to_html(fig)
    sns_plot.savefig(
        'D:\\Divya\\DjangoProject\\skincareproject\\static\\pairplot.png')
    # fig.savefig('C:\\Users\\Admin\\PycharmProjects\\djangoproject\\skincare\\assets\\pairplot.png')
    return render(request, 'pairplot.html', {'pairplot': "/static/pairplot.png", 'pairplot_page': 'active'})


def problemTypes(request):
    path = "D:/Divya/DjangoProject/skincareproject/media/" + input
    fig, ax = plt.subplots()
    fig.set_size_inches(20, 8.27)

    df = pd.read_csv(path)

    df['Skin_type'] = df.replace({'Skin_type': {'Normal Skin': "normal", 'Dry Skin': "Dry", 'Oily Skin': "Oily",
                                        'Combination Skin': "combination",
                                        'Not tested yet': "notTested"}})
    df.drop(df.loc[df['skin_problem_type'] == "no"].index, inplace=True)
    print(df.head())

    g = sns.catplot(x="skin_problem_type", hue="Skin_type", data=df, kind="count",
                height=6, aspect=1.2)
    g.set_xticklabels(rotation=45, horizontalalignment='right',
                 fontweight='light', fontsize='large')

    g.savefig(
        'D:\\Divya\\DjangoProject\\skincareproject\\static\\catplotnew.png')
    return render(request, 'problemTypes.html', {'plot': '/static/catplotnew.png', 'problemTypes': 'active'})


def compareIngred(request):

    import pandas as pd
    df = pd.read_csv(
```

```
    "D:\\Divya\\DjangoProject\\Datasets\\cosmetic_p.csv")
names = df.name.tolist()
context = {}
compareIngredientsform = CompareIngredientsForm()
context['compareIngredientsform'] = compareIngredientsform

productNames = get_product_names()

json_data = json.dumps(productNames)

context['compareIngred'] = 'active'
context['json_data'] = json_data

if request.method == "POST":

    form = CompareIngredientsForm(request.POST)

    if form.is_valid():
        form.save()

        product_type = form['product_type'].value()
        context['type'] = product_type
        first_product = form['first_product'].value()
        second_product = form['second_product'].value()
        import pandas as pd

        fieldsMatchingProduct1 = df.loc[(df['name'] == first_product)]

        print(fieldsMatchingProduct1.ingredients)
        ingredients1 = fieldsMatchingProduct1.iloc[0].ingredients
        context['product1'] = first_product

        fieldsMatchingProduct2 = df.loc[(df['name'] == second_product)]

        print(fieldsMatchingProduct2.ingredients)
        ingredients2 = fieldsMatchingProduct2.iloc[0].ingredients
        context['product2'] = second_product

        common_ingredients = []

        def get_jaccard_sim(str1, str2):

            a = set(str1.split(","))
            b = set(str2.split(","))

            c = a.intersection(b)
            for e in c:
                common_ingredients.append(e)

            return float(len(c)) / (len(a) + len(b) - len(c))

        p = get_jaccard_sim(ingredients1, ingredients2)
```

```
            print("common ingredients are", common_ingredients)

            if len(common_ingredients) != 0:
                context['common_ingredients'] = list(common_ingredients)
            else:
                context['no_common_ingredients'] = "No common Ingredients found"

            return render(request, 'compareIngred.html', context)

        return render(request, 'compareIngred.html', context)

    return render(request, 'compareIngred.html', context)


def reviews(request):
    df = pd.read_csv(
        "D:\\Divya\\DjangoProject\\Datasets\\finalreview.csv", error_bad_lines=False)
    names = df.name.tolist()

    if request.method == "POST":
        productname = request.POST['browser']

        df1 = df.loc[(df['name'] == productname)]

        # df1 = df.loc[(df['name'] == 'Lala Retro™ Whipped Cream') & (df['Label'] == 'Moisturizer_Combination')]
        df2 = df1['Reviews'].values

        reviewString = ""
        for i in df2:
            reviewString = i

        def ConvertToList(string):
            return list(string.split("|"))

        print("converted", ConvertToList(reviewString))
        input_review = ConvertToList(reviewString)

        import re
        import nltk
        from nltk.corpus import stopwords
        from nltk.stem.porter import PorterStemmer
        corpus = []

        def removeStopWords(corpus, input_review):
            for i in range(0, len(input_review)):
                review = re.sub('[^a-zA-Z]', ' ', input_review[i])
                review = review.lower()
                review = review.split()
                ps = PorterStemmer()
                review = [ps.stem(word) for word in review
                          if not word in set(stopwords.words('english'))]
                review = ' '.join(review)
```

```
        corpus.append(review)
    return corpus

print("courpus", removeStopWords(corpus, input_review))

from nltk.classify import NaiveBayesClassifier
from nltk.corpus import movie_reviews

# nltk.download('movie_reviews')

positive_fileids = movie_reviews.fileids('pos')
negative_fileids = movie_reviews.fileids('neg')

def extract_features(word_list):
    return dict([(word, True) for word in word_list])

features_positive = [(extract_features(movie_reviews.words(fileids=[f])), 'Positive') for f in
                positive_fileids]
features_negative = [(extract_features(movie_reviews.words(fileids=[f])), 'Negative') for f in
                negative_fileids]

threshold_factor = 0.8
threshold_positive = int(threshold_factor * len(features_positive))
threshold_negative = int(threshold_factor * len(features_negative))

feature_train = features_positive[:threshold_positive] + \
    features_negative[:threshold_negative]
feature_test = features_positive[:threshold_positive] + \
    features_negative[:threshold_negative]

classifier = NaiveBayesClassifier.train(feature_train)

for item in classifier.most_informative_features()[:20]:
    print(item[0])

# input_reviews = review

def getResult(reviews, input_review):
    main = []
    count = []
    for i in range(0, len(reviews)):
        eachWord = []
        eachWord.append(input_review[i].split(".")[0])
        probdist = classifier.prob_classify(
            extract_features(reviews[i].split()))
        pred_sentiment = probdist.max()

        eachWord.append(pred_sentiment)

        eachWord.append(round(probdist.prob(pred_sentiment), 2))
        main.append(eachWord)
        print(reviews[i], " ----- ", probdist.max())
```

```
        return main

    sentiments = getResult(corpus, input_review)

    df = pd.DataFrame(
        sentiments, columns=['Review', 'Predicted Sentiment', 'Probability'])
    print(df)

    fig, ax = plt.subplots()
    count = df["Predicted Sentiment"].value_counts()
    print("countt ", count)

    p = sns.countplot(x="Predicted Sentiment", data=df)
    plt.style.use('ggplot')
    plot = fig.savefig(
        'D:\\Divya\\DjangoProject\\skincareproject\\static\\review_count.png')

    plot = fig_to_html(fig)
    sns_plot = sns.catplot(x="Predicted Sentiment", y="Probability", hue="Review", data=df, kind="bar", aspect
=1,
                           sharex=True, sharey=True)
    sns_plot.savefig(
        'D:\\Divya\\DjangoProject\\skincareproject\\static\\reviews3.png')

    return render(request, 'reviews.html', {"p": "Successfull", 's': '/static/reviews3.png', 'reviews': 'active', 'names
': names, 'plot': [plot]})

  return render(request, 'reviews.html', {'reviews': 'active', 'names': names})


def classification(request):
  if request.method == "POST":
    # try:
    from sklearn.linear_model import LogisticRegression
    from sklearn.tree import DecisionTreeClassifier
    from sklearn.neighbors import KNeighborsClassifier
    from sklearn.naive_bayes import GaussianNB
    from sklearn.svm import SVC

    algorithmName = request.POST["ca"]
    print(type(algorithmName))
    print(algorithmName)
    prod = product.objects.all()
    a = []
    for p in prod:
        print("hello", str(p.csv))
        a.append(str(p.csv))
    input = a.pop()
    print(input)
    path = "D:/Divya/DjangoProject/skincareproject/media/" + input
    df = pd.read_csv(path)
```

```
print(df.dtypes)
skintype = pd.get_dummies(df['Skin_type'], drop_first=True)
gender = pd.get_dummies(df['Gender'], drop_first=True)
mode = pd.get_dummies(df['average_spend'], drop_first=True)
sample = pd.get_dummies(df['use_samples'], drop_first=True)
df.drop(['Skin_type', 'Gender', 'Mode_of_buying', 'Purchase_recommendation', 'use_samples', 'average_spe
nd',
        'Rate_Face_wash', 'Face_wash_avoid', 'Face_cream', 'Rate_Face_cream', 'Face_cream_avoid',
        'Body_lotion', 'choosing_product', 'Face_wash', 'skin_problem_type'], axis=1, inplace=True)


print(df.dtypes)
df = pd.concat([df, gender, skintype, mode, sample, mode], axis=1)
print(df.dtypes)
X = df.drop("skin_problems",
axis=1) y = df["skin_problems"]
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(
  X, y, test_size=0.5, random_state=0)
from sklearn.preprocessing import
MinMaxScaler scaler = MinMaxScaler()
X_train =
scaler.fit_transform(X_train) X_test =
scaler.transform(X_test) print(X_test)

accuracy_train = ""
accuracy_test = ""

def runAlgo(a):

  logmodel = a
  print(type(a))
  print(logmodel)
  logmodel.fit(X_train, y_train)

  accuracy_train = "Accuracy of " + algorithmName + \
    " on training set is " + \
    format(logmodel.score(X_train, y_train))
  print(accuracy_train)

  accuracy_test = "Accuracy of " + algorithmName + " on testing set " + format(
    logmodel.score(X_test, y_test))
  print(accuracy_test)
  predictions = logmodel.predict(X_test)
  from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
  # global classificationReport
  print("classification report",
     classification_report(y_test, predictions))
  print(accuracy_score(y_test, predictions)) report
  = classification_report(y_test, predictions)

  # global confusionMatrix
```

```
        print(confusion_matrix(y_test, predictions))

        # plot

        fig, ax = plt.subplots()

        from sklearn.metrics import roc_auc_score, roc_curve
        logit_roc_auc = roc_auc_score(y_test, logmodel.predict(X_test))
        fpr, tpr, thresholds = roc_curve(
            y_test, logmodel.predict_proba(X_test)[:, 1])
        plt.figure()
        plt.plot(fpr, tpr, label=algorithmName +
                 ' (area = %0.2f)' % logit_roc_auc)
        plt.plot([0, 1], [0, 1], 'r--') plt.xlim([0.0,
        1.0])
        plt.ylim([0.0, 1.05])

        plt.xlabel('False Positive Rate')
        plt.ylabel('True Positive Rate')
        plt.title('Receiver operating characteristic')
        plt.legend(loc="lower right")
        plt.savefig(
            'D:\\Divya\\DjangoProject\\skincareproject\\static\\classify.png')

        return accuracy_train, accuracy_test

    acctrain = ""
    acctest = ""
    if algorithmName == "Logistic Regression":
        a = LogisticRegression()
        imgurl = "/static/classify.png"
        acctrain, acctest = runAlgo(a)

    elif algorithmName == "Decision Tree Classifier":
        dtData = pd.read_csv(
            "D:\\Divya\\DjangoProject\\Datasets\\desisonTree.csv")
        from sklearn import metrics
        feature_cols = ['Gender', 'use_samples', 'Spending_annually',
                        'skin_problems', 'Ayurvedic_Products', 'age', 'Rate_Face_wash']
        X = dtData[feature_cols] # Features
        y = dtData.Buy
        X_train, X_test, y_train, y_test = train_test_split(
            X, y, test_size=0.5, random_state=0)
        # Create Decision Tree classifer object
        clf = DecisionTreeClassifier()

        # Train Decision Tree Classifer
        clf = clf.fit(X_train, y_train)

        # Predict the response for test dataset
        y_pred = clf.predict(X_test)
        acctrain = "Accuracy of decision tree algorithm" + \
```

```
            str(metrics.accuracy_score(y_test, y_pred))

        from sklearn.tree import export_graphviz
        from sklearn.externals.six import StringIO
        from IPython.display import Image
        import pydotplus

        dot_data = StringIO()
        export_graphviz(clf, out_file=dot_data,
                    filled=True, rounded=True,
                    special_characters=True, feature_names=feature_cols, class_names=['0', '1'])
        graph = pydotplus.graph_from_dot_data(dot_data.getvalue()) graph.write_png(

            'D:\\Divya\\DjangoProject\\skincareproject\\static\\DTree.png')
        Image(graph.create_png())
        imgurl = "/static/DTree.png"
      elif algorithmName == "KNeighbors Classifier":
        a = KNeighborsClassifier()
        imgurl = "/static/classify.png"
        acctrain, acctest = runAlgo(a)
      elif algorithmName == "Gaussian Naive Bayes":
        a = GaussianNB()
        imgurl = "/static/classify.png"
        acctrain, acctest = runAlgo(a)
      elif algorithmName == "Support Vector Classifier":
        a = SVC()
        imgurl = "/static/classify.png"
        acctrain, acctest = runAlgo(a)

      return render(request, 'classification.html', {"p": imgurl, "atrain": acctrain, "atest": acctest, 'classification': 'acti
ve'})

      # except:
      #     return render(request, 'classification.html', {"p1": "No such product name"})

   return render(request, 'classification.html', {'classification': 'active'})


def pie(request):

   if request.method == "POST":

      # getting the values

      from math import pi
      from bokeh.io import output_file, save,
      output_notebook from bokeh.palettes import inferno
      from bokeh.plotting import figure
      from bokeh.transform import cumsum

      colm = request.POST["type"]
      gender = request.POST["gender"]
```

```
path = "D:/Divya/DjangoProject/skincareproject/media/" +
input df = pd.read_csv(path)
df = df.replace(to_replace=["no", "No"],
            value="Don't use")
genderWiseProducts = df.loc[df['Gender'] == gender]

genderWiseProducts = genderWiseProducts.groupby(
   by=colm, as_index=False).agg({'ID':
   pd.Series.nunique})

genderWiseProducts.sort_values(
   by=['ID'], inplace=True, ascending=False,)

x = {}

for i in genderWiseProducts.index:
   x[genderWiseProducts[colm][i]] = genderWiseProducts['ID'][i]

print("map created", x)

data = pd.Series(x).reset_index(name='value').rename(
   columns={'index': 'country'})
data['angle'] = data['value']/data['value'].sum()
* 2*pi data['color'] = inferno(len(x)) output_file(

   'D:\\Divya\\DjangoProject\\skincareproject\\templates\\productCount.html')
      if gender ==
         "Female":
          gender =
        "Womens"
else:
   gender = "Mens"

p = figure(plot_height=600, plot_width=800, title=gender+" "+colm,
        toolbar_location=None, tools="hover", tooltips="@country: @value", x_range=(-
        0.5, 1.0))

p.wedge(x=0, y=1, radius=0.3,
      start_angle=cumsum('angle', include_zero=True),
      end_angle=cumsum('angle'), line_color="white", fill_color='color',
      legend='country', source=data)

p.axis.axis_label = None
p.axis.visible = False
p.grid.grid_line_color = None

output_notebook()
```

```
    save(p, notebook_handle=True)
    return render(request, 'pie.html', {"htmlPage": 'productCount.html', 'pie_page': 'active', "type":
    type})

return render(request, 'pie.html', {'pie_page': 'active'})
```