

Chapter 1

Introduction

1.1 Recommendation Systems

Recommendation Systems are designed to recommend relevant customized content or services to users based on different factors. The recommendations are predictions that would be most liked by user based on user-item data. The predictions could be based on user's likes, ratings or purchase history. Companies like Amazon, Netflix, YouTube utilize recommendation systems to keep their users engaged by providing with a customized product list. This has increased the scope of what user can buy since they are exposed to more products through recommendations. Since last decade, numerous firms have impacted the retail market by providing customized products to consumers based on their inputs such as StitchFix, Function of Beauty, Proven Skincare etc. All of these companies have carved a new path by addressing the gap in the market while helping users understand what is right for them. [1].

1.2 Working of a Recommendation System

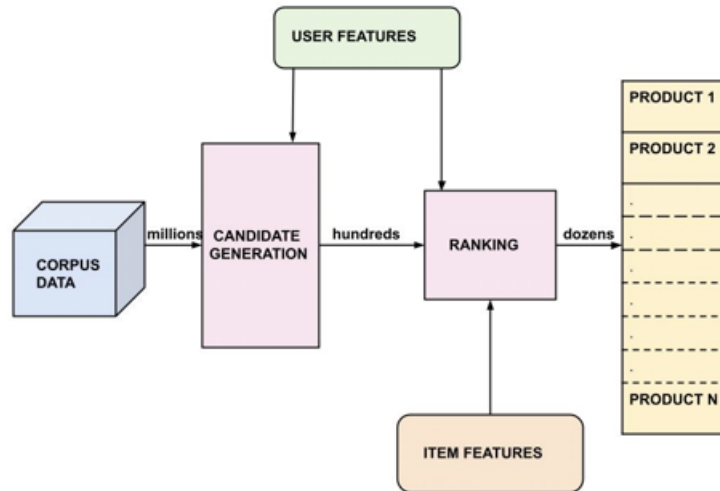


Figure 1.1: Recommendation System Model

1. Corpus Data

Consists of collection of all possible item as well as user data available at the respective websites. The data can be in millions which is input into the candidate generation block.

2. Candidate Generation

Candidate generation filters out the content or products to hundreds based on the user's features such as type of product, amount of money spent on an average by a user, frequency of buying a particular product. For example, the candidate generator in YouTube reduces billions of videos down to hundreds or thousands based on user's video preferences. [6]

3. Ranking

Ranking is used as a final filter to recommend the closest predictions based on user's ratings, sentiments or dislikes. The ranking is also dependent on user as well as item features such as price of the product, brand of the product, ratings for that product. The final list is the list of products that are shown to the user as a recommended list as shown in fig 1.2.

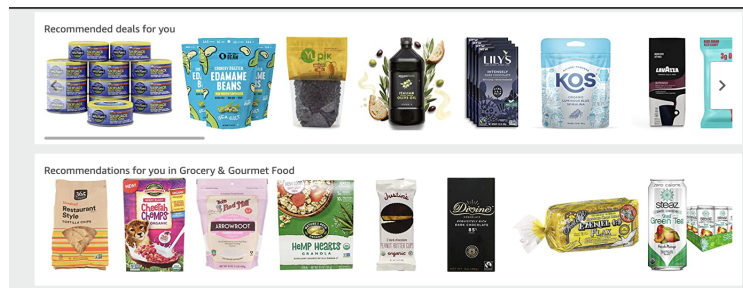


Figure 1.2: Recommendation based on Amazon purchase history

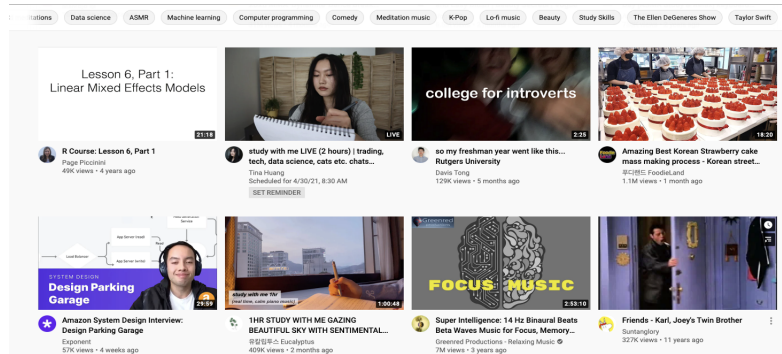


Figure 1.3: Recommendation based on YouTube purchase history

1.3 Types of Recommendation Systems

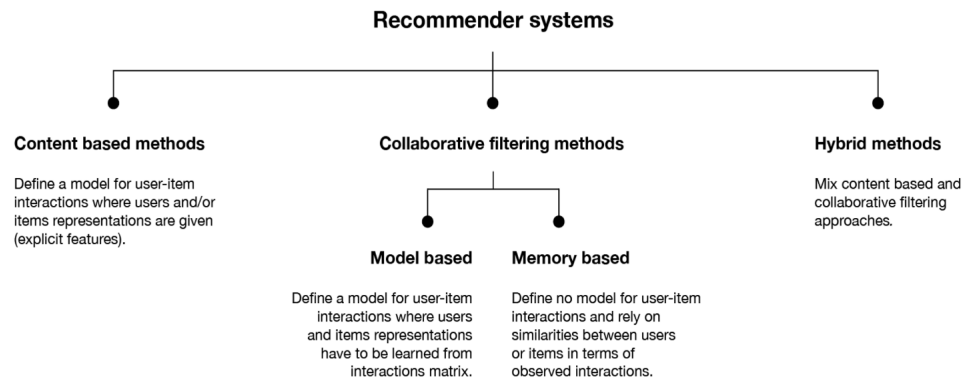


Figure 1.4: Types of Recommendation Systems [8]

1. Content based filtering

	HORROR	THRILLER	COMEDY	ROMCOM	ACTION	DRAMA	DOCUMENTARIES
MOVIE 1	1	0	0	0	0	0	1
MOVIE 2	0	1	0	0	0	1	0
MOVIE 3	0	0	1	0	1	0	0
MOVIE 4	0	0	0	1	0	0	0

Figure 1.5: Content Based Filtering

When it is just user based it is known as Content based filtering. We can see in fig 1.5 and calculate the dot product of the movies for each genre for this

given user. Movie 1 scored the highest after calculating the dot products like so

$$mov1 = 1.1 + 0.0 + 0.0 + 0.0 + 0.0 + 0.1 + 1.1 = 2$$

This means movies that are horror and are documentaries are going to be the best recommendation for this particular user based on the information of user's watch history.

2. Collaborative filtering

When user-item interaction is considered, it is known as Collaborative filtering. We can see in the fig 1.6, the matrix on the right hand side shows the ratings given by 4 different users for 5 different items. In practical situations, recommendation systems help us fill in the missing values in this matrix, by predicting based on user-item interaction.

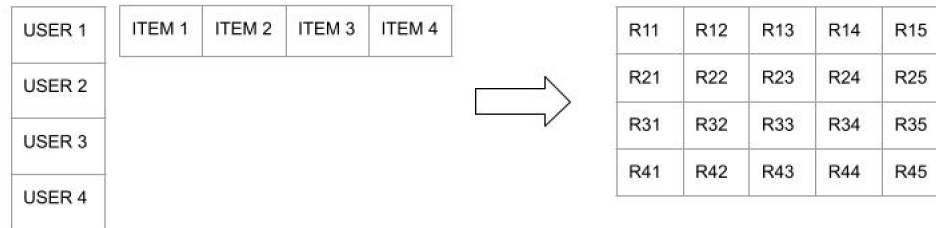


Figure 1.6: Collaborative Filtering

3. Hybrid filtering

Hybrid filtering is a combination collaborative and content based filtering together. In this report, we get a good result with Hybrid method using KNN and Cosine Similarity.

Chapter 2

Related Work

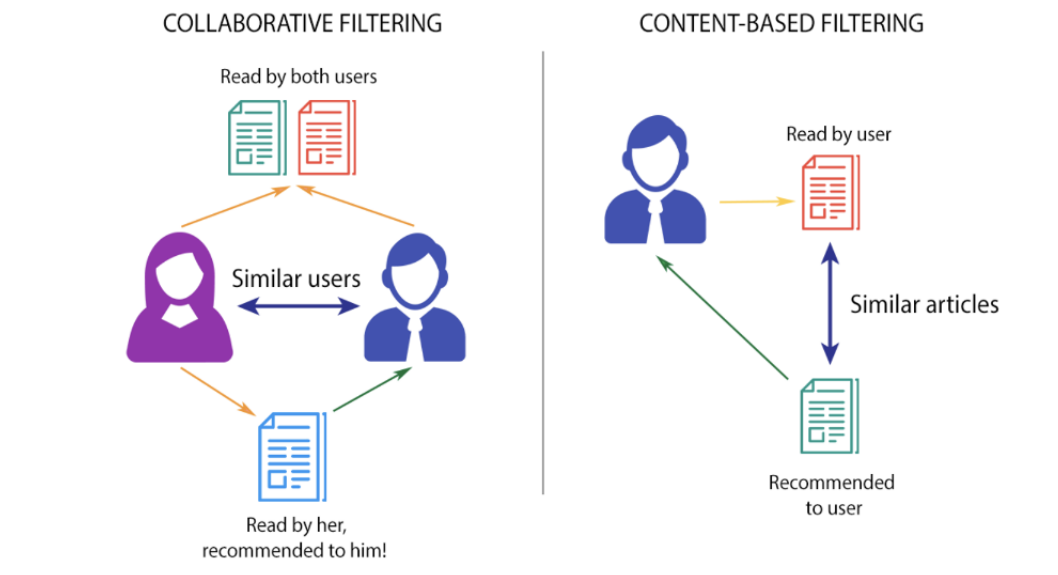


Figure 2.1: Content V/S Collaborative Filtering [3]

Content based filtering seems to be good for cold start problems where we do not have history for a new user while Collaborative filtering relies on user's history,

where the similarity of a user is implied based on the similar likes for a particular product as we can see in fig 2.1. Based on this information, the recommendation system can recommend a new product based on the other's user's likes, since these two are noted down as similar users.

There are three classic collaborative filtering approaches that are used in recommendation systems such as nearest neighbor, matrix factorization and/or neural networks. We have implemented all three of them but we do not have enough research that caters to the under serving community in the beauty community. The approach is to find a similar products used by users in the past or find similar users who have liked the same products. The issue we will face, as we progress through the report is, how these approaches alone do not take user's demographics into consideration and can end up providing a recommendation based on highest similarity but they might not fit the skin tone or skintype of the user.

In our scenario, we have to take care of two things:

1. Cold Start problem

The idea is to help users with zero to no user-item interaction history by utilizing the information user put as input. In our case, we would take down the skin tone, skin type and budget as their input. This would help us find similar users of the same demographics using nearest neighbor.

2. Products suitable for under serving skin tones

Once the products are found based on user's input, we need to see whether the similar users liked those products or not. This will help us in identifying whether this would be good recommendation or not.

Chapter 3

Data Scraping and Preparation

3.1 Data acquisition

The dataset has 3749 entries and around 291 classes. A class for a product is defined as (Brandname, Productname, Shade of the product) here, which is known as PID in our table 3.1. This means we have 291 unique products based on PID(Product ID). Data acquisition was the biggest challenge in this project and took up most of the time as there were multiple web drivers that were not compatible with the Sephora website, if using python. There were no available APIs at the time and the content on website was changed twice during the course of these 8 months. We wanted information on these features from the website to be able to apply any machine learning or recommendation models on it: Brand, Product, Ingredients, Skin type, Skin tone, Reviews.

The first three were easily accessible on the main page of the website but skin

tone and skin type were not readily available. These two features were user based hence we had to find user data. On Sephora website, the reviews have all of the user's data input before the actual review like skin tone, skin type, age, shade of the product and so on. Hence, the last three attributes had to be extracted from the review data set. Unfortunately, there were no readily available data sets for this market and had to be extracted in some certain steps. The fig 3.1 shows the idea of how we wanted our data set to finally look like.

The idea:

The data was supposed to be scraped in certain fashion and there were multiple ways of scraping the data. The following web drivers were considered before using the Selenium webdriver:

1. Scrapy

It extracts data from the websites in a structured manner. This webdriver was chosen first because it is written in Python, so there are high chances of it being compatible with code written in Python. The ideal dataset was supposed to be structured, making it an enticing option. But in this project, scrapy did not end up working out. Hence, this was not the ideal scraper for this project.

2. BeautifulSoup

Beautiful Soup is another python library and is usually ideal for web scraping using XML path. BeautifulSoup returned an empty list while scraping on the Sephora website. After researching on why it returned an empty list, it was found that beautiful soup does not work well on Sephora. It worked well on Ulta and Target website but not on Sephora. So, this web scraper could not

be used either.

3. Selenium

Selenium finally worked on Sephora website and returned the outputs exactly as needed. However, there were some challenges faced with the use of this webdriver. Selenium is a dynamic webdriver while Sephora is a dynamic website. Selenium also does not have a good history with Python, so that posed its own set of challenges. Selenium works best with Java. But since, Selenium was the only webdriver that returned the outputs exactly as needed, this ended up being the ideal web scraper for this project.

3.2 Data Acquisition : Challenges

1. `button.click()`

The reviews needed to be loaded by clicking on the button, after the 6 reviews on that page were recorded. Hence, the button functionality was inculcated, so that it would save the manual work and time.

2. `time.sleep(10)`

While scraping, the driver would skip some data from the current page, and move on to the next one. This problem is commonly known as Lazyloading. The solution to this problem was to inculcate `time.sleep(5)`

3. Computation issues

There were 120 brands for blush on the website but computationally, it was

not possible for the system to extract over 100 reviews per product and extract 120,000 rows of data. Hence, I decided to limit the data to 35 brands and have 3500 rows of data.

4. Limited dataset

There were 120 brands on the website for blushes. The idea was to acquire 100 reviews per brand and have in total of 12,000 rows of data. However, due to multiple time.sleep(10), it took 5 hours of continuous sitting to extract data for 33 brands.

5. Changes made on the website

The website has changed since March and there are a few new functionalities that have been added, that are not captured in the current dataset.

After researching and analyzing Selenium webdrivers, it is safe to say that the desired dataset was acquired but it ultimately did not entirely eliminate the manual work like scrolling through the webpages. This was done so that all the data is acquired from the pages.

	PID	Price	Ingred	Skintone	Skintype	Review
0	Rare Beauty..., Stay Vulnerable..., Rose	21	Isodecyl Isononanoate,...	Medium	Combination	Was super hesitant bc cream blushes ...!
1	Rare Beauty..., Stay Vulnerable..., Apricot	21	Isodecyl Isononanoate,...	Fair	Dry	This blush blends like a dream
2	Rare Beauty..., Stay Vulnerable..., Berry	21	Isodecyl Isononanoate,...	Tan	Combination	For natural looking make up looks...
3	NARS, Blush, Orgasm	30	Isododecane, Talc,...	Medium	Combination	Love everything about this blush!
4	NARS, Blush, X	30	Isododecane, Talc,...	Light	Combination	Too bright red...
5	NARS, Blush, Throat	30	Isododecane, Talc,...	Medium	Dry	A keeper for sure!
6	Rare Beauty..., Soft Pinch..., Happy	20	Hydrogenated Polyisobutene,...	Light	Dry	the best cream blush ever!
7	Rare Beauty..., Soft Pinch..., Grace	20	Hydrogenated Polyisobutene,...	Fair	Normal	An everyday product!!
8	Rare Beauty..., Soft Pinch..., Bliss	20	Hydrogenated Polyisobutene,...	Fair	Combination	Best blush out there!
9	Rare Beauty..., Soft Pinch..., Joy	20	Hydrogenated Polyisobutene,...	Fair	Combination	As the other reviews have stated...
10	Benefit Cosmetics, Hoola Matte..., Caramel	30	Talc, Iron Oxides...	Light	Normal	LOVE THIS FOR OLIVE SKIN
11	Benefit Cosmetics, Hoola Matte..., Hoola	30	Talc, Iron Oxides...	Medium	Normal	FAVVVV
12	Benefit Cosmetics, Hoola Matte..., Hoola	30	Talc, Iron Oxides...	Light	Dry	Beautiful bronzer
13	Benefit Cosmetics, Hoola Matte..., Hoola	30	Talc, Iron Oxides...	Medium	Combination	I love this bronzer...
14	Benefit Cosmetics, Hoola Matte..., Hoola	30	Talc, Iron Oxides...	Light	Combination	love it, but a little orange
15	Benefit Cosmetics, Hoola Matte..., Hoola	30	Talc, Iron Oxides...	Light	Combination	Best bronzer. Matte. Good for all year round.
16	MILK MAKEUP,Lip+Cheek,Perk	28	-Mango Butter, Avocado Oil, and Apricot Oil...	Light	Combination	I wanted to love it so much!
17	MILK MAKEUP,Lip+Cheek,Work	28	-Mango Butter, Avocado Oil, and Apricot Oil...).	Medium	Combination	BUY ITTTT!!!
18	MILK MAKEUP,Lip+Cheek,Work	28	-Mango Butter, Avocado Oil, and Apricot Oil...).	Light	Combination	Pretty, but Pimply

Table 3.1: Data extracted from the website

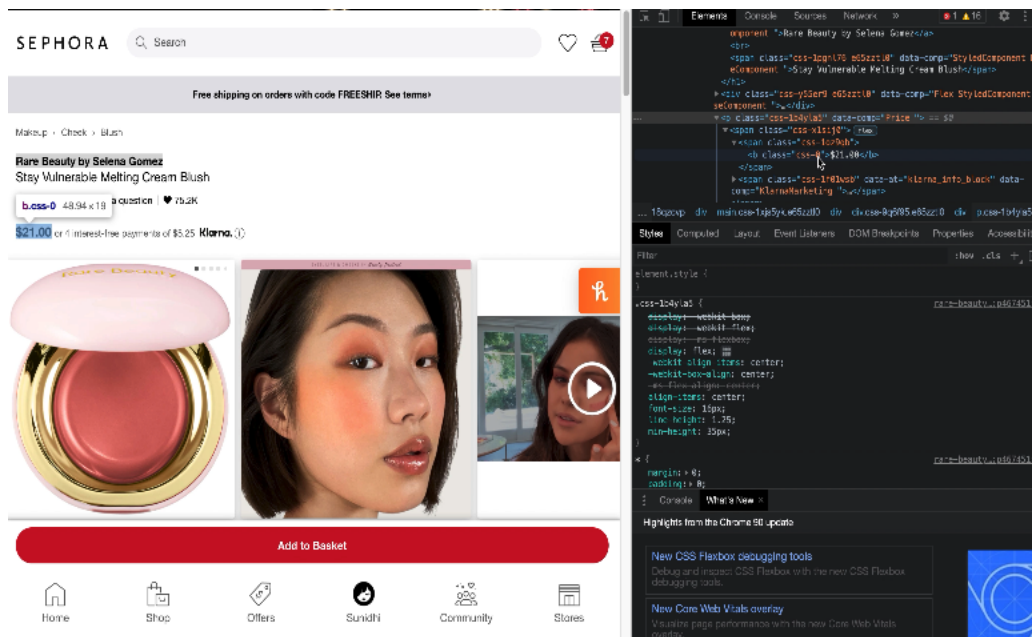


Figure 3.1: Data Scraping from Sephora Website

3.3 Data Preparation

3.3.1 Exploratory Data Analysis

In fig 3.2, there are users who have purchased more than blush. For example, user jen101 has bought 20 products. This will be quite beneficial if we require to train our model on users who have bought the most items.

As seen in the table 3.2, product of Product ID = (Anastasia Beverly Hills, Contour Kit Medium) has been used by 66 people from our data set. Same goes for (Tarte, Amazonian Clay Waterproof Bronzer Princess), (MILK MAKEUP, Mini Matte Cream Bronzer Baked) etc. We have also illustrated the number of products that have been used by the same number of users in the fig 3.3. For example, there

Product ID	Count of common products
Anastasia Beverly Hills: Contour Kit: Medium	66
tarte: Amazonian Clay Waterproof Bronzer: Princess™	66
MILK MAKEUP: Mini Matte Cream Bronzer: Baked	66
Tower 28 Beauty: Bronzino Illuminating Bronzer: Coast	66
Tower 28 Beauty: BeachPlease Tinted Lip+Cheek Balm: Hour	66
Too Faced: Sun Bunny Natural Bronzer: Bunny	66
HUDA BEAUTY:3D Cream and Powder Highlighter Palette:Edition	66
Benefit Cosmetics: Hoola Matte Bronzer Jumbo: Bronze	66
Hourglass: Ambient® Lighting Bronzer: Light	66
Charlotte Tilbury:Filmstar Bronze & Glow Contour Duo:Fair/Medium	65
Benefit Cosmetics: Dandelion Baby-Pink Blush: Dandelion	62
Natasha Denona: Diamond & Blush Palette: Darya	54
MILK MAKEUP: Matte Bronzer: Baked	54
tarte: Amazonian Clay Matte Waterproof Bronzer: Princess™	54
MILK MAKEUP: Lip + Cheek: Werk	50
Laura Mercier: Matte Radiance Baked Powder Compact: 01	47
Charlotte Tilbury: Cheek to Chic Blush - Pillow Talk Collection: Talk	47

Table 3.2: Common Product count

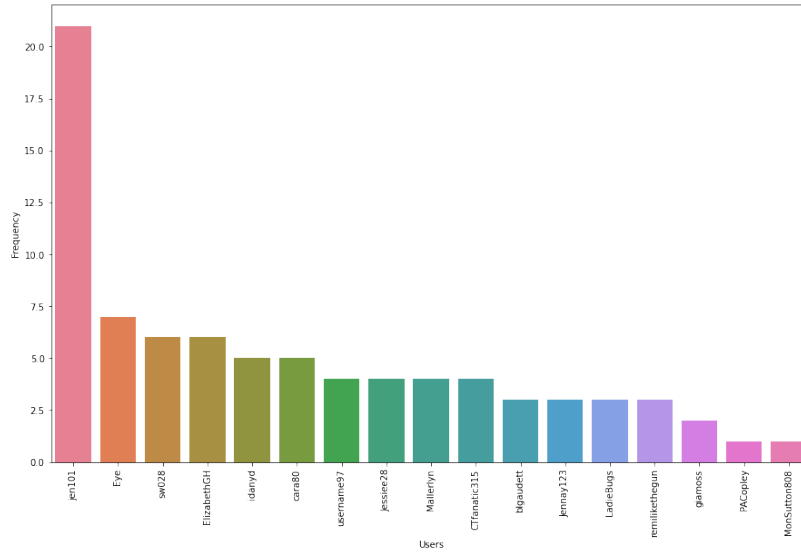


Figure 3.2: User purchase frequency

are 8 products that have been used by 66 number of users. These products could be used to check accuracy later. We shall discuss about it in the coming chapters.

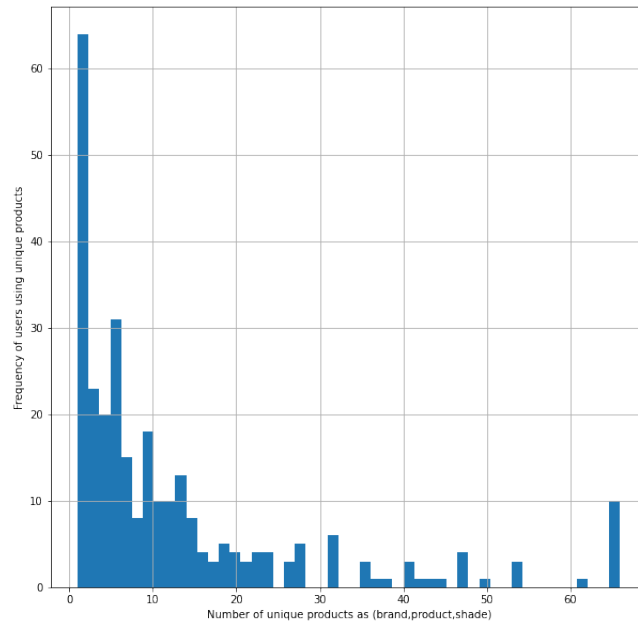


Figure 3.3: Frequency of products of same brand, product and shade

We can see that we do not have enough data points for ebony skin tone. While training, with very few ebony data points, it might give undesirable results or no results at all for a user input with ebony skin tone.

The price attribute has the most variation as we can see in the following figure, which would pose some problems as we are going to see further.

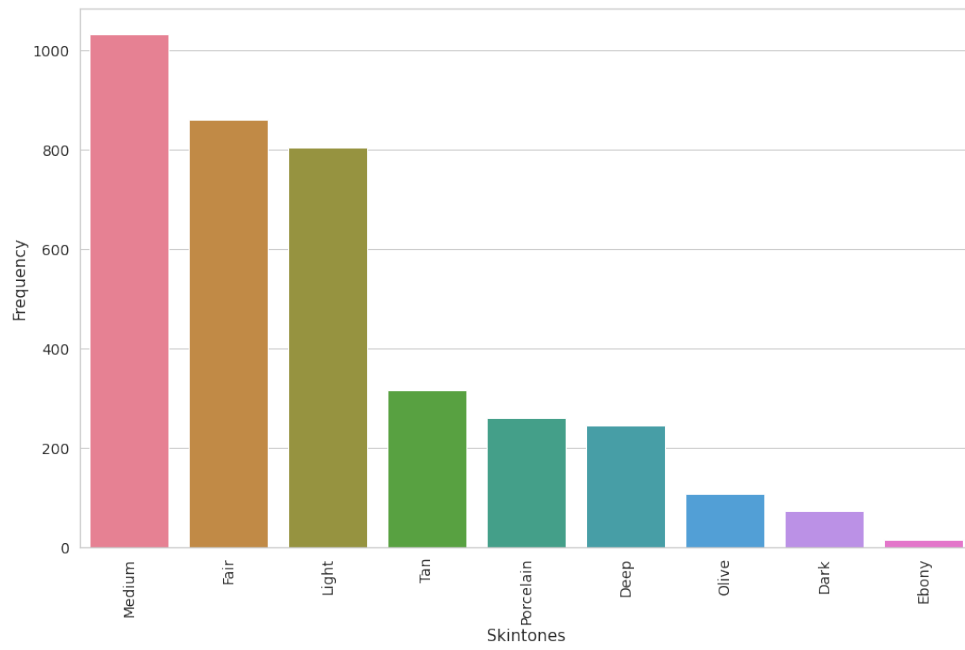


Figure 3.4: Skintone Frequency

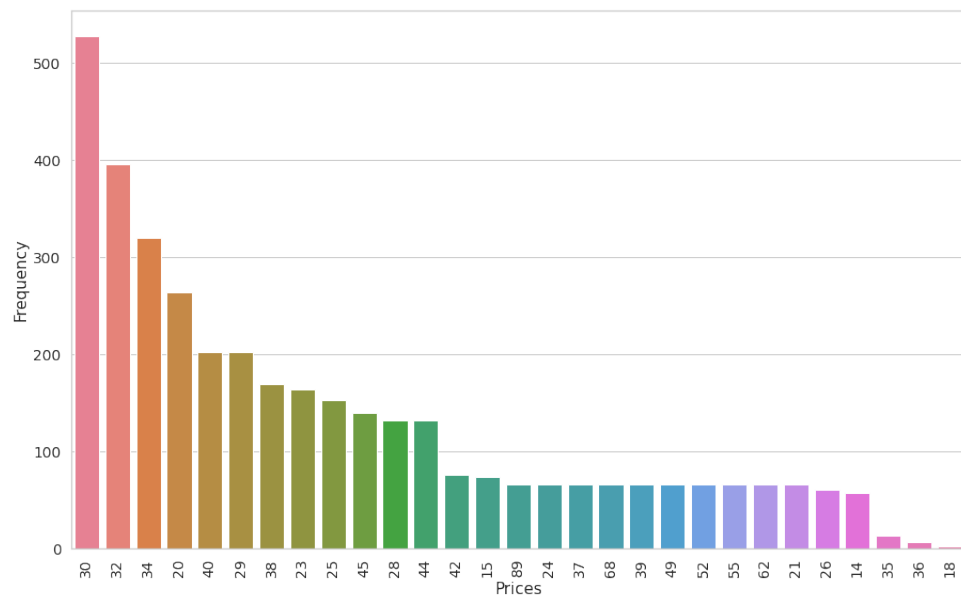


Figure 3.5: Price Frequency

3.3.2 Data Cleaning

The data set has to be cleaned and pre-processed before inputting it into the model. Hence, the reviews acquired from the website have to be cleaned by removing non-character, converting them into lower case, removing stopwords, stemming and then splitting text. The cleaned reviews looked something like shown in table 3.3.

Review	Cleaned Review
Was super hesitant bc cream...	was super hesitant bc cream...
This blush blends like a dream.	this blush blends like a dream i
It is the very first time I..	it is the very first time i..
For natural looking make up looks..	for natural looking make up..
User friendly and natural!	user friendly and natural does
Easy to Use Cream Blush	easy to use cream blush color payoff
Omg I cannot say enough good things	omg i cannot say enough
Easy to apply and lasting	easy to apply and lasting love so
Finally Found The Perfect Blush	finally found the perfect blush back
Love this. I've been looking for	love this i ve been looking
Beginner Friendly	beginner friendly
amazing blush	amazing blush this blush is so creamy and
One of my favorite	one of my favorite love this cream blush

Table 3.3: Reviews before and after cleaning

Chapter 4

User Demographics

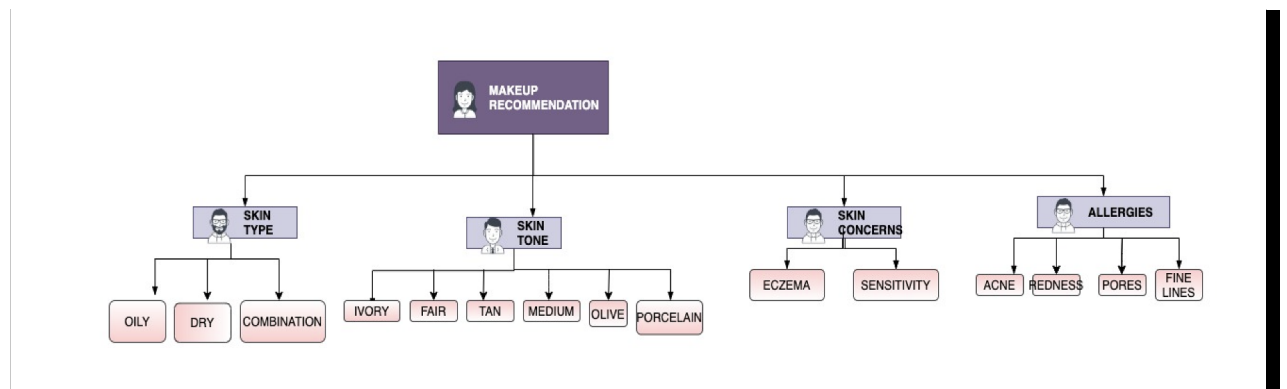
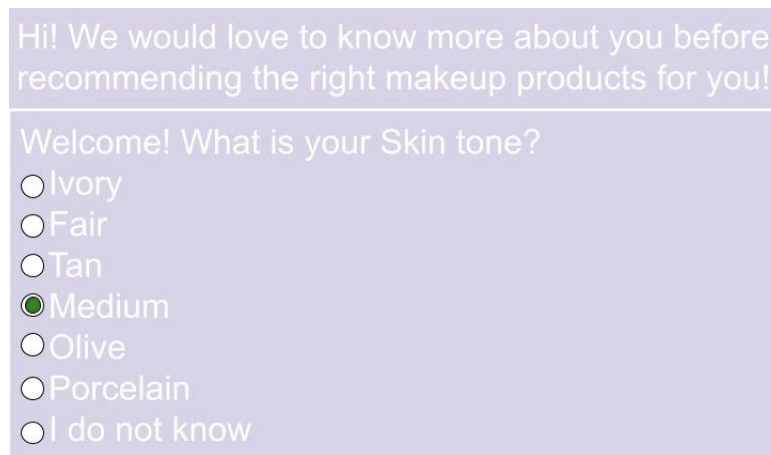


Figure 4.1: User Demographics Flowchart

We have to design a model in such a way that user's skin type, skin tone, skin concerns and budget are looked after. When the recommendations are made, our model should be able to consider all of these demographics. This means a user with medium skin tone and oily skin type should only be recommended products for medium skin tone and oily skin type.

4.0.1 Frontend perspective of the idea

The questions that would be asked would follow in the same format as shown in fig4.1 on the frontend once this project is deployed. The users would select the options that would suit them the best. These inputs would be encoded and then input into the model. The closest predictions would be shown as recommendations

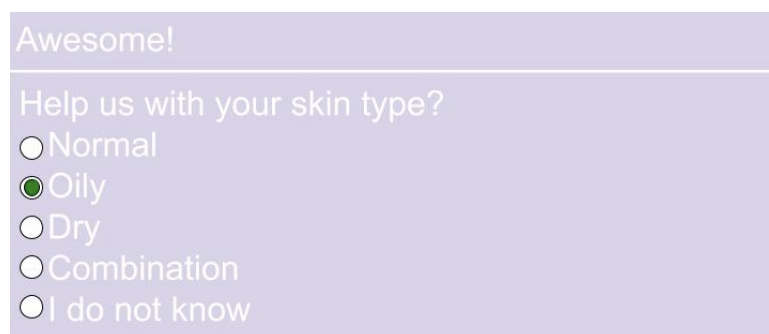


Hi! We would love to know more about you before recommending the right makeup products for you!

Welcome! What is your Skin tone?

- ☐ Ivory
- ☐ Fair
- ☐ Tan
- ☒ Medium
- ☐ Olive
- ☐ Porcelain
- ☐ I do not know

Figure 4.2: Frontend Prototype: Skin Tone



Awesome!

Help us with your skin type?

- ☐ Normal
- ☒ Oily
- ☐ Dry
- ☐ Combination
- ☐ I do not know

Figure 4.3: Frontend Prototype: Skin Type

Awesome! We have plenty of products for Medium Skin tone and Normal skin type!

Do you have any skin concerns?

- ☐ Eczema
- ☒ Sensitivity
- ☐ Acne
- ☐ Redness
- ☐ Enlarged Pores
- ☒ Fine Lines
- ☐ I do not know

Figure 4.4: Frontend Prototype: Skin Concerns

Okay Alex, we have these 5 blushes for you today. Happy shopping!

1. NARS Blush in Orgasm
<https://www.sephora.com/product/blush-P2855?skuld=2396422&icid2=skugrid:p2855>
2. Rare Beauty by Selena Gomez Soft Pinch Liquid Blush in Joy
<https://www.sephora.com/product/rare-beauty-by-selena-gomez-soft-pinch-liquid-blush-P97989778?skuld=2354140&icid2=skugrid:p97989778>
3. Tower 28 Beauty BeachPlease Lip + Cheek Cream Blush in Rush Hour
<https://www.sephora.com/product/beachplease-tinted-balm-blush-P449342?skuld=2284875&icid2=skugrid:p449342>
4. Dior BACKSTAGE Rosy Glow Blush in Coral
<https://www.sephora.com/product/dior-rosy-glow-blush-P454762?skuld=2328383>
5. Tarte Amazonian Clay 12-Hour Blush in Captivating
<https://www.sephora.com/product/amazonian-clay-12-hour-blush-P278610?skuld=1604917&icid2=skugrid:p278610>

Figure 4.5: Frontend Prototype: Recommendations for the user

Chapter 5

Proposed Model

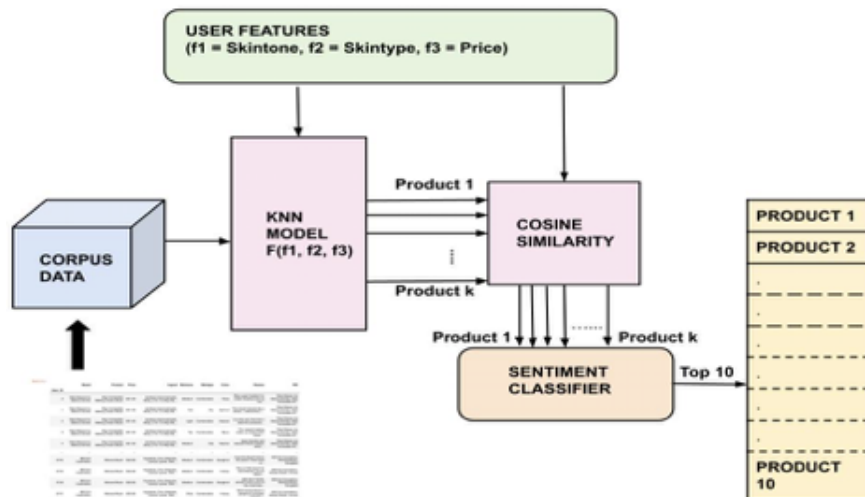


Figure 5.1: Recommendation Model

Our method involves Hybrid Filtering which is Collaborative + Content Filtering using K-Nearest Neighbours and cosine similarity.

KNN known as K-Nearest Neighbours, is an approach that is used to predict a

class of one user based on the data points that are closest to this particular user.

Cosine similarity is used to calculate how similar items are based on the distance to each other in the feature space. In our case, we are using cosine similarity as an added filter to get a closer prediction than KNN.

Sentiment Classifier is utilized to finally filter out the products from recommended list that have a good review using the sentiment score.

In basic terms, KNN predicts what class a new input belongs to, or returns the nearest neighbours to that input. In the fig 5.2, the red diamond is the new input, and KNN will help determine the class of this input. Based on the picture, it can be concluded that, it would belong to the yellow class. If the nearest neighbors were to be calculated, it would return the 2 yellow pentagons along with a white triangle first and in ascending order would follow the rest of them, based on how near they are to the red diamond.

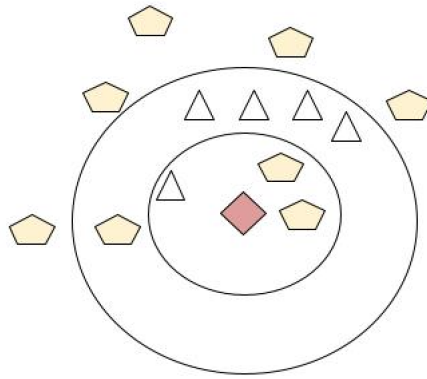


Figure 5.2: K-Nearest Neighbours

Cosine Similarity is given by:

$$\cos(x, y) = \frac{x \cdot y}{||x|| * ||y||}$$

where x and y are two vectors in the feature space and we are trying to find a point that is closest to the user's input.[10]

1. **Step 1 : Collaborative Filtering:**

KNN is used to get output products nearest to the class mapped in the feature space. The output products are acquired based on past interactions of users with the product with similar demographics as the current user. The demographics of the user are acquired based on the quiz we mentioned before which gives us input in the form of (Skin tone, Skin type, Budget). This helps with initial cold start problem, which can be taken care of once the user reviews the recommendation and utilized in a feedback loop to gauge the accuracy.

2. **Step 2 : Content Filtering**

Once the products are acquired, we use cosine similarity to find similar products based on our output. This is calculated by finding the similarity between different vectors in the feature space. The intent is to be able to cater the results more towards the particular skin tone and skin type of the user.

3. **Step 3 : Sentiment Classifier**

Since the reviews were not labelled, the sentiment is calculated using TextBlob.

The Sentiment of the reviews are classified from 0-5 as 1: -1 to -0.6, 2: -0.6 to -0, 3: -0.2 to 0.2, 4: 0.2 to 0.6, 5: 0.6 to 1. 1 is the most negative, 2 is negative, 3 is Neutral, 4 is positive and 5 is the most positive sentiment. TextBlob gives us both polarity and subjectivity scores. The final sentiment score is calculated by multiplying both polarity and subjectivity, where subjectivity here is used like a weight as to how subjective or objective a review is and polarity is determined on how negative or positive the sentiment was.

Following are the recommendations for a user input of (Medium, Normal, 30) which represents Medium skin tone, Normal skin type and 32 as the budget. The important thing to notice here is that the output still shows products that are for Olive skin tone or tan skin tone and oily skin type. The reason behind this is price is holding the highest weight in the recommendations because Price attribute is varying the most in the data set. Hence, some of the outputs are closer to price 30 than the others which caused other skin tones and types to enter too. The cosine similarity simply acts as a filter here. When the output from KNN is input into CNN, it is necessarily finding the closest products to user's demographics from this data, which is the output to KNN. Hence, when we look at our observation, the unnecessary skin tones and types are eliminated. Final filtering is done using the sentiment classifier, which finally ejects the products that are mentioned for user's demographics but might be not be well rated.

Chapter 6

Ingredients Study

The skin concerns are directly related to what a user puts on their skin. The ingredients in the products react differently to different skin types and skin concerns. It is important to bear in mind how potent a product is when it comes to skin concerns. For example, for an acne prone skin, a product that has acne aggravating ingredients such as talc or any alcohol related ingredients, they can make a user's skin condition worse. This in turn, would mean our model is not doing a good job after all. Hence, a study on ingredients has been done to understand how could we address skin concerns better and inculcate it in our model.

In this experiment, only first 6 ingredients have been considered per product as through research it was found that the first 6 ingredients are the most important and potent ones. The potency of the ingredients goes in a decreasing order, the first ingredient being the most potent one and so on.

The ingredients that we see in the following table has been prepared manually,

by making a bag of words of first 6 ingredients per product for all the products and then calculating the safety of the ingredient with respect to each skin concern. '1' means the product is not safe while '0' means the product is safe for the skin.[2][5]

	INGREDIENTS	ACNE	FINE LINES	SKIN TEXTURE	ECZEMA	SENSITIVITY	Enlarged Pores
0	Isododecane	0	0	0	0	0	0
1	Talc	0	1	0	0	1	1
2	Dimethicone	1	0	0	1	0	0
3	Caprylic/Capric Triglyceride	0	0	0	0	0	0
4	Phenoxyethanol	1	1	0	1	0	0
5	Bismuth Oxychloride	0	0	1	0	0	1
0	Hydrogenated Polyisobutene	0	0	0	0	0	0
1	Hydrogenated Poly(C6-14 Olefin)	0	0	0	0	0	0
2	Mica	0	0	0	0	0	0
3	Octyldodecanol	0	0	0	0	0	0
4	Ethylene/Propylene/Styrene Copolymer	0	0	0	0	0	0
5	Trimethylsiloxysilicate	0	0	0	0	0	0
0	Mango Butter	0	0	0	0	0	0
1	Avocado Oil	1	0	0	0	0	0
2	Ricinus Communis (Castor) Seed Oil	0	0	0	0	0	0
3	Persea Gratissima (Avocado) Oil	1	1	0	0	0	0
4	Ethylhexyl Palmitate	0	0	0	0	0	0
5	Helianthus Annuus (Sunflower) Seed Wax	0	0	0	0	0	0
0	Avocado Oil, Orange Peel Wax, Shea Butter, Helianthus Annuus (Sunflower) Seed Oil	0	1	0	0	0	0
1	Ricinus Communis (Castor) Seed Oil	0	0	0	0	0	0
2	Theobroma Cacao (Cocoa) Seed Butter	0	0	0	0	0	0
3	Cera Alba (Beeswax)	0	0	0	0	0	0
4	Euphorbia Cerifera Cera/Euphorbia Cerifera (Candelilla) Wax	0	0	0	0	0	0
5	Butyrospermum Parkii (Shea) Butter*	0	0	0	0	0	0
0	Amazonian Clay	0	0	0	0	0	0
1	Vitamin E	0	0	0	0	0	0
3	Polyethylene	0	0	0	0	0	1
4	Zinc Stearate	0	0	0	0	0	0
5	Isononyl Isononanoate	0	0	0	0	0	0
0	Oryza Sativa (Rice) Starch	0	0	0	0	0	0
1	Zea Mays (Corn) Starch	0	0	0	0	0	0
2	Silica	1	0	0	0	0	0
3	Ethylhexyl Palmitate	0	0	0	0	0	0
4	Nylon-12	0	0	0	0	0	0
0	Titanium Dioxide	0	0	0	0	0	0
1	Octyldodecyl Stearoyl Stearate	0	0	0	0	0	0
2	Boron Nitride	1	0	0	1	0	0
3	Synthetic Fluorophilopite	0	0	0	0	0	0
4	C30-45 Alkyl Dimethicone	1	0	0	0	0	0
5	Butyrospermum Parkii (Shea) Butter	0	0	0	0	0	0
0	Sorbitan Isostearate	1	0	0	0	0	0
1	C20-24 Alkyl Dimethicone	1	0	0	0	0	0
2	Hyaluronic Acid	0	0	0	0	0	0
3	Talc	0	1	0	0	1	0
4	Methyl Methacrylate Crosspolymer	0	0	0	0	0	0
5	Patented Biomimetic Pigment	0	0	0	0	0	0
1	Micronized Pigments	0	0	0	0	0	0
4	Mango Butter, Apricot Oil, Ricinus Communis (Castor) Seed Oil	0	0	0	0	0	0
5	C12-15 Alkyl Benzoate	0	0	0	0	0	0
0	Helianthus Annuus (Sunflower) Seed Oil	0	0	0	0	0	0
1	Coconut Alkanes	0	0	0	0	0	0
2	Disostearyl Malate	0	0	0	0	0	0
3	Polybutene	0	0	0	0	0	0
4	Caprylyl Methicone	0	0	0	0	0	0
5	Synthetic Wax	0	0	0	0	0	0
0	Vinyl Dimethicone/Methicone Silsesquioxane Crosspolymer	1	0	0	0	0	0

Table 6.1: Ingredient Potency per skin concern

This dataset is used to check whether ingredients that are finally returned as recommendations, hold a safe degree of threshold value ≤ 3 , for the skin concerns

mentioned by the user. This threshold value is calculated by simply adding the values of skin concerns, for the first 6 ingredients per product from the recommendation list. For example, a user mentions the skin concerns to be sensitivity and fine lines as mentioned in fig 4.4, the potency will be calculated by adding the values for sensitivity and fine lines of first 6 ingredients from the table 6.1. In this case, safe degree value for Isododecane, Talc, Dimethicone, Caprylic/Capric, Triglyceride, Phenoxyethanol, Bismuth Oxychloride ingredients

$$Safe\ degree = [0 + 1 + 0 + 0 + 0 + 0] + [0 + 1 + 0 + 0 + 1 + 0] = 3$$

Here, for Rare Beauty by Selena Gomez blush in Joy ended up being a safe product for sensitive and fine line prone skin as the blush holds a value of 3 which is within the threshold. The threshold of 3 was decided by trial and error method. The idea is to make sure users who have skin concerns buy products that will not aggravate the condition more. These products will not help fix the concerns, only help to accommodate them.

Chapter 7

Discussion

In this chapter, we would discuss about the other methods that were tried and tested, before concluding that KNN + cosine similarity combined gave better results. We already know the data set comprises of 3749 rows of data with 291 unique classes. The model needs to output a best match product for any user input of (Skin tone, Skin type, Budget). The returned output is supposed to be used to get similar products using similarity model. The models that we will discuss here would be multiclass Logistic Regression, Support Vector Machines and Alternating Least Squares method.

7.1 Classification Models

7.1.1 Multi-class Logistic Regression

The problem can be solved by fitting regression model to the data points and rounding off the predicted value to the closest class. Suppose we wish to predict a class which can be any of (Rare Beauty by Selena Gomez, Stay Vulnerable Melting Blush,Nearly Mauve),(Fenty Beauty, Cream Blush, Cool Berry),..... (LAWLESS, Make Me Blush Talc-Free Velvet Blush, Summer). The model can be interpreted as finding conditional probability of belonging to a class out of 291 classes. Mathematically, we can say:

$$P(y = k|x) = f(x) = \frac{1}{Z} \exp(\langle w_k, x \rangle)$$

where k is the total number of classes and Z is a constant such that all probabilities add up to 1. The above expression gives the likelihood of y_i given x_i Z is known as a partition function given by:

$$Z = \sum_{k=0}^{K-1} \exp(\langle w_k, x \rangle)$$

The class scores are given by vector Z. The regression variable w is a set of vectors w_0, w_1, \dots, w_{k-1} and are stacked in a matrix. In this case, we used multi-class logistic regression model and acquired an accuracy of 3.4% which is very low. The score is so low because:

1. We have limited data points with 291 unique classes, which makes it difficult for the model to predict the right product.

2. Feature weightage is also impacting the predicted output. After observing the results, price attribute was holding more weight than skin tone and skin type. We want our model to give equal weightage to all three attributes, that will help in getting the best match.
3. Score provides a mean accuracy of all the predicted classes, which becomes a strong metric as each data point should be correctly predicted [?]

So let us see if other models help with the issues we face with the above model.

7.1.2 Support Vector Machines

Support Vector Machines are one of the best machine learning methods when getting the correct answer is a higher priority. They work really well with relatively small data sets. Mathematically, the minimizer of the below equation is known as a Support Vector Machine:

$$L(w) = \sum_{i=1}^n \max(0, 1 - y_i \langle w_k, x \rangle) + \frac{\lambda}{2} \|w\|_2^2$$

In our case, Polynomial Kernel is used because the training data set has a lot of overlap. Hence, we were not able to find a satisfying Support Vector Classifier to separate different classes. Hence, a Support Vector Machine with a Polynomial Kernel with Support Vector Classifier(SVC) is used to train the model. The idea in kernel method is to map a given data to a higher dimensional feature space if there

is a lot of overlap in training data.

$$x \rightarrow \phi(x)$$

where $\phi(.)$ is some non-linear function.

In this case, we used Support Vector Machine and acquired an accuracy of 28% which is a lot better than the one acquired by Logistic Regression. But it is still not great because of feature weightage, and hence impacting the predicted output.

7.2 Matrix Factorization Algorithm

7.2.1 Alternating Least Squares

Alternating Least Squares is a Collaborative Filtering method that factorizes a user-item matrix into a user matrix and item matrix.

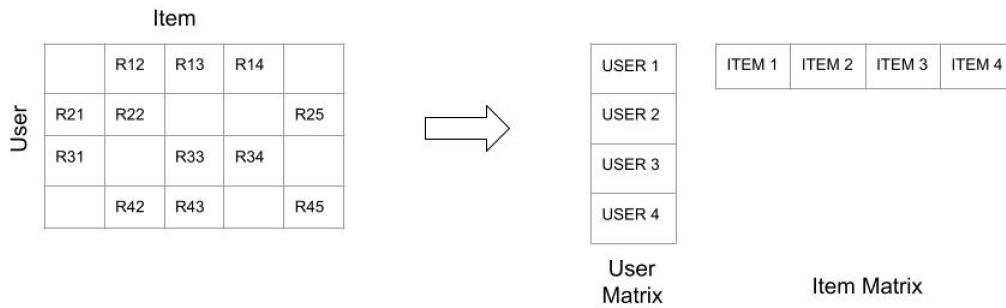


Figure 7.1: Factorization of a matrix

Matrix factorization framework is described as follows:

$$\operatorname{argmin} ||R - U^T V||^2 + \lambda(||U||_2^2 + ||V||_2^2)$$

ALS algorithm states: Fix user factor U , solve for V . Fix item factor V , solve for U .

$$V = (UU^T + \lambda I)^{-1}UR$$

$$U = (VV^T + \lambda I)^{-1}VR^T$$

where R is partially observed user-item preference matrix as seen in fig 7.1 Since the cost function becomes quadratic, it is easier to find an optimum and reduce the value of the cost function. In our case, Implicit library has been used. The factors hyperparameter was set to 7 because there are 7 features that we are training our model on : Skin tone, skin type, price, brand, product, shade and sentiment.

	User Name	Precision	Recall
1	jen101	0.006923	0.042641
2	Eye	0.12524	0.285714
3	ElizabethGH	0.100000	0.166667
4	sw028	0.100000	0.100000
5	ShiraChanel	0.23541	0.5000
6	idanyd	0.042578	0.100000
7	makeupgt5	0.058824	0.250000
8	username97	0.142857	0.250000
9	makeupfreak12	0.111111	0.500000
10	Mailerlyn	0.250000	0.250000

Table 7.1: ALS Precision and Recall score

This is a subset of the actual table. The average precision value came out to be 7.84% and a recall value of 28.44%. The values are not as good as acquired by our model as we will see in Results chapter. Hence, we decided to finalize our model to be of KNN+cosine similarity.

	User Name	Precision	Recall
1	jen101	0.076923	0.052632
2	Eye	0.117647	0.285714
3	ElizabethGH	0.100000	0.166667
4	sw028	0.100000	0.200000
5	ShiraChanel	0.444444	0.800000
6	idanyd	0.142857	0.200000
7	makeupgt5	0.058824	0.250000
8	username97	0.142857	0.250000
9	makeupfreak12	0.111111	0.500000
10	Mailerlyn	0.250000	0.250000

Table 7.2: Model's Precision and Recall score

7.3 Observations

As observed, the outputs from these classification models was not giving the best match due to feature weightage. All of the predictions were giving more weightage to price because price attribute varied the most, out of all of the attributes. These models would return only one product as the prediction which would look like this: (Brand, Product and Shade). This product should ideally fit the entered user demographics and be a best match. However, with these models, the accuracy score was not good, meaning the predicted product was not right for the user. Hence, we decided to use K-Nearest Neighbours that would output multiple neighbours using `KNeighborsClassifier($n_neighbors = 50$)` and then input these products into cosine similarity model to get a better, closer prediction.

Chapter 8

Results

Price	Ingred	Skintone	Skintype	Review
32	Silica, C12-15 Alkyl	Medium	Normal	My oily skin thanks this
32	Mica, Silica, Alkyl Be..	Medium	Normal	Best formula ever
32	Silica, C12-15 Alkyl...	Medium	Normal	Need this blush..
34	Silica, C12-15 Alkyl...	Medium	Normal	Beautiful...
29	Mica, Silica, Alkyl Be...	Olive	Normal	I don't know...
32	Mica, Silica, Alkyl Be...	Olive	Normal	Not good for olive
32	Silica, C12-15 Alkyl...	Medium	Oily	Dry, patchy blush...
32	Mica, Silica, Alkyl Be..	Olive	Dry	Best formula ever..
29	Silica, C12-15 Alkyl...	Medium	Normal	Good bronzer for...
29	Talc, Iron Oxides..	Tan	Normal	Maybe I will buy this..
29	Talc, Iron Oxides..	Tan	Normal	Hate it, so patchy...
32	Talc, Iron Oxides..	Tan	Normal	Gorgeous

Table 8.1: Output Products using KNN

Price	Ingred	Skintone	Skintype	Review
32	Silica, C12-15 Alkyl..	Medium	Normal	My oily skin thanks this
32	Silica, C12-15 Alkyl..	Medium	Normal	Best formula ever
32	Silica, C12-15 Alkyl..	Medium	Normal	Okay blush not...
32	Silica, C12-15 Alkyl..	Medium	Normal	Good bronzer for my..
29	Mica, Silica, Alkyl Be..	Olive	Normal	horrible fo..
32	Silica, C12-15 Alkyl..	Medium	Normal	This blew m..
32	Silica, C12-15 Alkyl..	Medium	Oily	Recommends...
34	Silica, C12-15 Alkyl..	Medium	Normal	Not sure if..
34	Talc, Iron Oxides..	Tan	Normal	Proud I bought it...

Table 8.2: Output Products using Cosine Similarity

Price	Ingred	Skintone	Skintype	Review	Sentiment
32	Silica, C12-15 Alkyl	Medium	Normal	Wow, thi	5
32	Mica, Silica, Alkyl Be..	Medium	Normal	Best formula ever	5
32	Isodecyl Isononanoate...	Medium	Normal	13+ hour wear...	5
32	Isodecyl Isononanoate...	Medium	Normal	Perfect Blush...	4
29	Isodecyl Isononanoate...	Medium	Normal	Easy to...	4

Table 8.3: Output Products using Sentiment Classifier

8.1 Accuracy Metrics

To calculate how accurate this final model is, Precision and Recall have been inculcated. Precision implies how many of the positive identifications are actually correct[?] and is given by following formula:

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

Recall calculates how many actual positives are identified correctly[?] and is given by following formula:

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

In this case, a subset of 25 users was selected from the database. These users have used at least more than 2 products and one product was picked per user as the training data. These user's attributes like Skin tone, skin type, price were taken into consideration for each product. When this data is passed into KNN, then cosine and then in sentiment classifier, we are acquiring a list of 19 products per user. We then find the intersection between the master database and the list that we just acquired, this intersection shows how many of the products used by user prior are actually present in the recommendation list.

let us take an example of the user Jessiee28 who has a True Positive value of 2. To calculate precision,

$$Precision = \frac{2}{2 + 17} = 0.105$$

To calculate recall,

$$Recall = \frac{2}{2 + 2} = 0.50$$

Hence, we have an average precision value of 11.96% and a recall value of 38.49%. The values are not too shabby given the amount of dataset and the ample number of classes.