*DS2500: Final Project Repor*t

Cosmetics Recommendations and Price Analysis

Using Machine Learning Techniques

Devyanshi Chandra– chandra.d@northeastern.edu

Kirti Magam– magam.k@northeastern.edu

Problem Statement and Background

In our project, we intended to dive into what happens behind the scenes of recommendation systems and price prediction, and how we can use these systems and emulate recommendation algorithms that are currently used by websites. We specifically focused on cosmetics and makeup since we both are avid users of various types of users, as well as the fact that makeup is incredibly personal and varies on the user's preferences. We aimed to use data on products available online to create a system that would use that information to give suitable suggested products, as well as using that data to estimate the price ranges of certain products. In addition, we planned for our system to be interaction based – meaning that the user would receive products to try based on their inputs into our system.

As previously mentioned, makeup is used by millions of people across the world with different preferences and different backgrounds, cultures, genders, and perceptions of beauty. Due to this, using makeup and cosmetic items is an incredibly intimate activity and is entirely reliant on what an individual intends to see from this product. This means that seemingly small details, such as the colors, the product's expected results, the recommended skin type, or the ingredients, are vital for the consumer. Oftentimes, current recommendation systems may not account for these variables, and may inaccurately promote products, which causes more inconvenience for the user. Since cosmetic items are so widely utilized, we wanted to recreate a system that would find similarities within each other's descriptions, as well as a system that would be able to predict the price ranges of the input product to understand how they're implemented in currently used algorithms for recommendations and potentially further study these algorithms and how they can be integrated.

The Data and its Characteristics

The dataset we used for the prediction and recommendation algorithms was posted online on Kaggle[1] and was directly web-scraped from the Sephora website. It contained over 9,000 products that were available in 2020 and included several different aspects such as the product's marketed name, brand, category (as in the specific kind of cosmetic product it was marketed as), price, ingredients, etc. They also included information on discounts available, average ratings, descriptions, and other information that could be used to create more elaborate systems in the future. For our systems, we primarily focused on the brands, categories, sizes, prices, and descriptions and filtered out any products that didn't have information within these columns, leaving about 5,700 products from over 300 brands to work with. Our data was primarily diverse, and included products from all different subcategories of cosmetics, including lipsticks, face makeup, eye makeup, skincare products, fragrances, gift sets, etc. This allowed us to diversify what kinds of recommendations we could provide, as well as expand the data we use to aim for a more precise price prediction.

Data Science Approaches

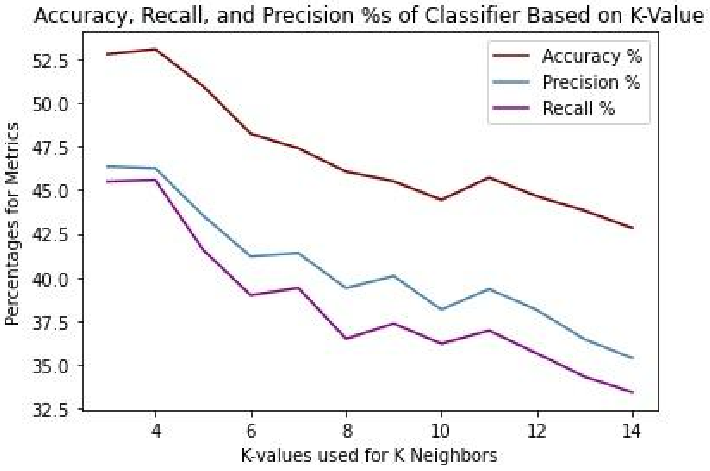
To complete this project, we had to utilize numerous data science techniques for analysis of our data. The first component of our project was using supervised machine learning to complete a price prediction analysis. We used a KNN classifier to classify products into price categories based on given features. Given the features of brand, product category, and product size, the price was predicted. The data was randomly split into a training and testing set so that the model could be trained and then classify the test set based on how its closest neighbors were sorted into price categories.

[1] Kaggle (2020), https://www.kaggle.com/datasets/raghadalharbi/all-products-available-on-sephora-website

For the second portion of the project, we used natural language processing techniques to build a cosmetics recommendation system based on the similarity of product descriptions. This used vectorization and cosine similarity to create suggestions. Vectorization turns these product descriptions into a vector[2]. Cosine similarity then calculates the cosine of the angle formed between two vectors so that it can mathematically compute how related these descriptions are. The last data science approach used was for the visualization of how our recommendation system performed. We used a simulation of fifty trials to see how it performed with accuracy in a large sample.

Results and Conclusions

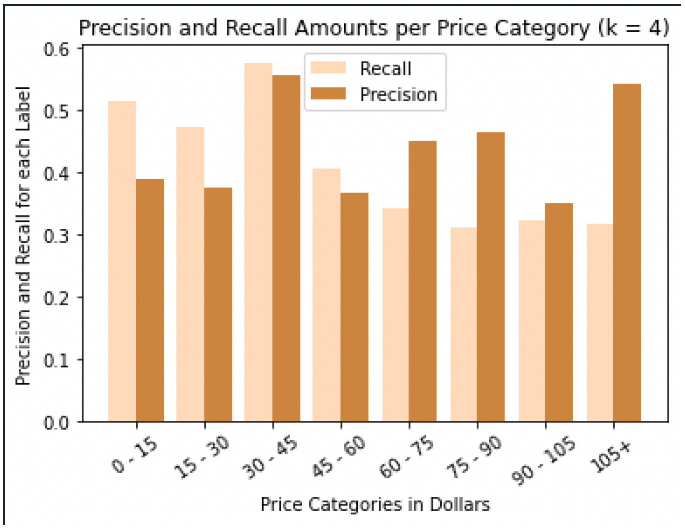
Once we used our data to complete both systems, we wanted to determine how accurate our results were for our price prediction system, as well as for our recommendation system. To do so, we first wanted to test how our price prediction system would differ based on the value of K that our KNN Classifier would take in. To do so, we ran a few trials where we tested the accuracy metrics of our price prediction system as K increased between the ranges of 3 and 15.



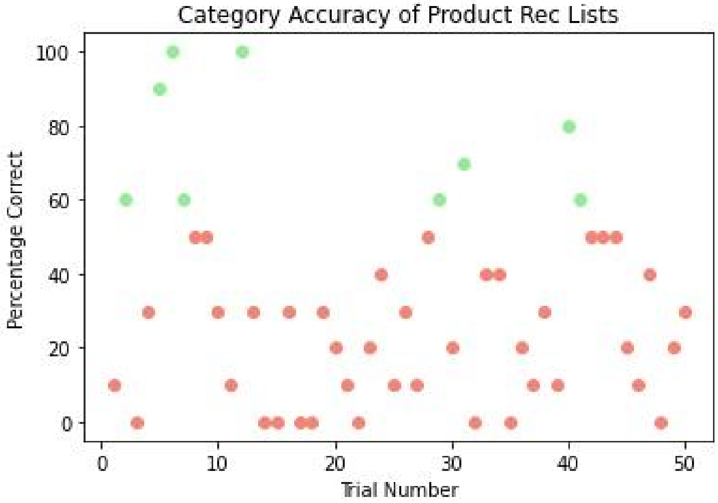
In this graph, we portrayed the accuracy, precision, and recall percentages that our metrics report returned once we'd run our simulations for the K values used in our KNN Classifier. While it's important to note that the KNN Classifier splits the data at random each time it's been called, whether between each time the graph is portrayed or even between each K

[2] Towards Data Science (2019): https://towardsdatascience.com/understanding-nlp-word-embeddings-text-vectorization-1a23744f7223

value's trial, overall the results remained fairly similar. The K values that received the better accuracy metrics were primarily between the K value ranges of 3 to 5. This can be explained by our set classifications of the prices. Because our categories are divided into price ranges of $15 until it hits $105, as the K value increases, it may include values that fall within more than one price range as the input point's neighbors, and therefore cause difficulties in determining which price range the input should fall within.



In the second graph, we portrayed the accuracy metrics for each individual price category, while K = 4. As mentioned before, we divided our price ranges into 8 categories, starting from 0 and increasing by $15 increments until it hit $105, as our data was primarily left-skewed and the prices generally fell under $100. We extracted the precision and recall metrics for each price category while running the KNN Classifier and plotted their metrics/ Based on our second graph, it seems like products within the price range of $30-45 and above the price of $105 were the most accurate ranges in terms of precision. This means that they generally had better ratios of true positives to total positives assigned, or that it was most accurate in assigning the products in the categories they were meant to be allotted to. Furthermore, for recall, the price categories of $30-45, and $0-15 were generally the ranges that had the best ratios of true positives allotted to the total number of true positives that were meant to be allotted. This means that those ranges specifically had better accuracy than other ranges in discarding products that weren't supposed to be included within those categories. Again, while the graphs change each time the program is run, the results primarily stay similar. This can be explained by the data itself, from the graph it seems that the products within the price range of $30-45 had better accuracy metrics out of all ranges we'd included. In our dataset, most products fell into those categories, meaning that while the data was randomly split in the KNN Classifier, it still had access to more information on that price range.



Moving onto our product recommendation system, we decided to determine our algorithm's accuracy based on whether the items we suggested for the user were the same as the user's input product (or category, if they chose to use the brand and category as a way to get a recommendation). We chose to run 50 trials, where we randomly chose products to input into our algorithm and took the list of recommended products, and analyzed the percentage of products that had the same category as the input. Through this, we calculated the percentage of correct categories and plotted them in the graph for each trial on the left, with the green being all trials with accuracy percentages above 50%, and red below 50%. Based on its graph, it seems like our recommendation system didn't pass our initial goal of the system having an accuracy rate of over 50%. However, as we inquired further on why our accuracy rate was lower than expected, we realized that the categories provided by the dataset had significant overlaps. For example, certain products fell into categories that were somewhat synonymous with each other, such as "Fragrances", "Fragrances and Perfumes", "Body Mists", etc. This caused our evaluation metric to give accuracy rates that were significantly lower than what we'd expected and should be further analyzed in order to precisely determine how accurate our recommendation system is.

Our price prediction analysis demonstrated how to find the best K value when using classification. We also learned that since we created seven separate price categories, it increased the difficulty of correctly classifying each product name. This explained why the accuracy, precision, and recall percentages were mostly below 50%. This helped us understand how the classification process changes accuracy metrics as we change different factors in our data. Through our recommendation system, we were able to gain knowledge on incorporating user interaction with natural language processing techniques so that data could be filtered and organized to give a user their desired results. Beyond this, it demonstrates the importance of computers understanding human language so that they can perform these language-based tasks that aid users. In general, knowing how to process real world data and form connections between different data points is helpful for others to access and gain further insight.

Future Work

As we continued to work on this project and implemented the various types of machine learning techniques for our systems, we ran into several obstacles and stop-points, which led us to fall short of what we'd intended to complete in our proposal. Firstly, in order to continue to work on the recommendation system for future usage, a more precise evaluation metric should be used. As we mentioned previously, the similar types of categories included in our dataset disinhibited us from properly evaluating our accuracy metrics for the recommendation system. Further analysis on that and potentially including aspects, such as accuracy in our Cosine Similarity and Count Vectorizator techniques for our descriptions, to determine the recommendation system's overall accuracy in several trials could allow us to understand how our simulation may have flaws. Once we have a concise accuracy system, we can expand our recommendations to other categories, such as ingredients, and to specific keywords in the product’s descriptions to cater further to the user’s needs as we’d initially planned in our proposal. This and the techniques we’ve used for the product recommendation system could be utilized to study how certain ingredients and price may be related to each other for particular kinds of products. Furthermore, they can be used to study the marketing strategies of brands and determine whether or not there’s a relationship between the prices of products and the usage of specific keywords in the descriptions of products.