Building a Recommender System with Basic Text Models

Kegan Wong
A14933874
kmw037@ucsd.edu

I. Introduction

COVID-19 has resulted in a heavy reliance on technology. Since in person gatherings have become dangerous, many have transitioned to digital means, such as zoom calls for school and online shopping. This has given rise to the importance of communication through telephone or computer. Given that communication is not in person anymore, text has been widely used as a primary substitute. In this paper, I want to explore how textual data can be utilized to build an effective recommender system.

II. Dataset Utilized

The dataset studied in this paper is a collection of multiple reviews on clothing that was rented from the company Rent the Runway. Each entry in this dataset includes the following information:

- Fit - categorical size
- User ID - a unique ID given to a user
- Item ID - a unique ID given to an item
- Bust Size - clothing size measurement
- Weight - user weight in pounds
- Rating - measure of satisfaction
- Reason - why the clothing was rented
- Review - text description of product
- Body Type - anatomy of body
- Summary - summarized review
- Height - user height in feet/inches
- Age- user age in years
- Date - time stamped date of review

The dataset can be found at the link cseweb.ucsd.edu/~jmcauley/datasets.html. There are 192,544 reviews in this dataset. The demographics of each reviewer can be summarized in the following table.

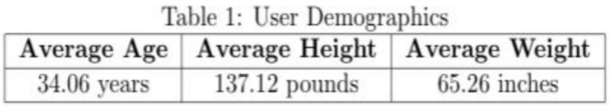
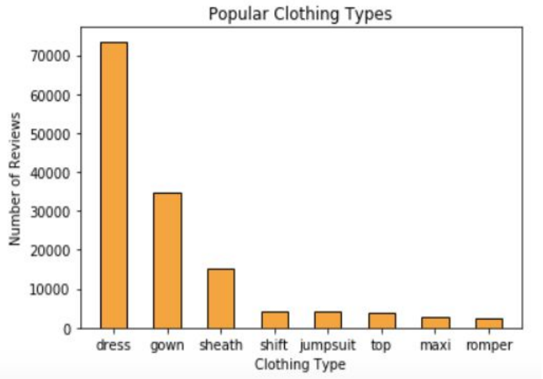
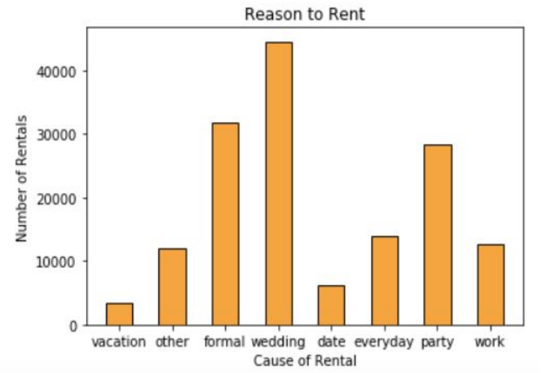


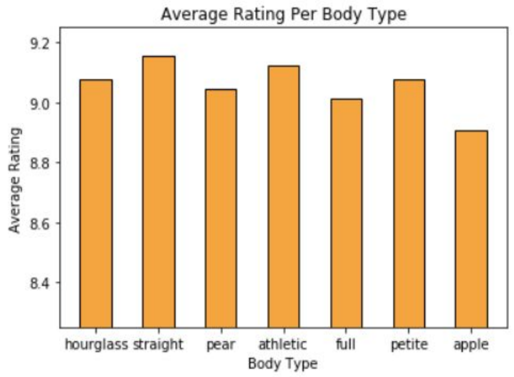
Table 1: User Demographics

Average Age	Average Height	Average Weight
34.06 years	137.12 pounds	65.26 inches

The dataset is mainly focused on different categories of clothing, which I will define as the type of clothing; examples include jeans, dresses, and more. To summarize the basic characteristics of the dataset, I utilized bar graphs. The bar graphs visualize how different categories of clothes compare in terms of popularity (Popular Clothing Types, in particular the top 8), the popular reasons for renting clothing (Reason to Rent), and how people of different body types rated differently (Average Rating Per Body Type).



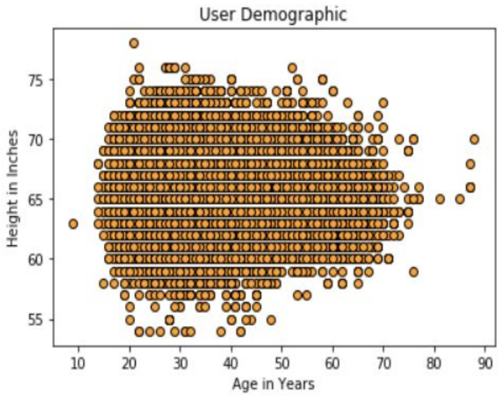


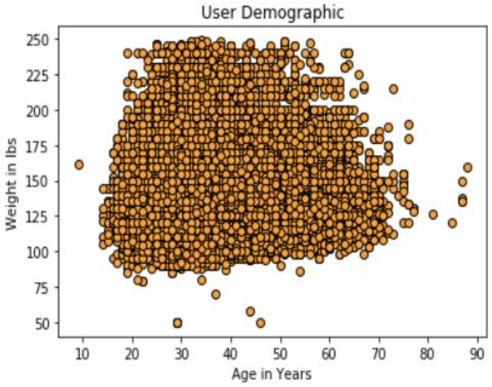


Building a Recommender System with Basic Text Models

Kegan Wong
A14933874
kmw037@ucsd.edu

Some interesting findings that I discovered was how the average rating for the apple body type provided a noticeably lower average rating than the rest of the body types. Another interesting discovery came when visualizing the demographics of the dataset, as I noticed that the weights and heights of each user appeared to be relatively uniform for each age, as shown in the scatterplots below. This contradicts my assumption that there should be a positive correlation between such variables.





However, some of my expectations were met, such as how weddings were the number one reason for renting clothing, and how dresses were the most commonly reviewed item as opposed to less used items such as the romper.

III. Predictive Task

Given that the qualities of the reviews appear to be lengthy and descriptive, my aim of this project is to find out how these textual reviews can be utilized to predict the reason for clothing rental. The models that I will use will be a bag of words model, followed by an n-grams model. This model seemed most appropriate because the dataset is heavily text based, and I want to explore basic natural language processing models. Most of the numerical data that was provided are just basic demographic information. This likely has less predictive capabilities than a lengthy, descriptive text on the product, although this is a bold assumption. Furthermore, the other instance variables on the properties of clothing are not categorical tasks, so I will not place my focus on such instances.

In order to evaluate the effectiveness of my chosen models, I will compare my performance to two intuitive baseline models. The first baseline model predicts the reason for rental based on the category of clothing, and what that category of clothing is commonly used in. An example would be predicting "wedding" if the category of clothing was "blazer", since blazers are commonly worn at weddings. For my second baseline model, I will predict the reason for rental if the reason appears in the review. If the word does not appear, I will predict the most popular reason for rental.

Along with both baseline models, I will assess an error metric by keeping track of the precision, recall and f-score for each reason of rental. By definition, the precision, recall and f-score can be calculated by the following:

1. $precision = \frac{|\{relevant\ reason\} \cap \{total\ retrieved\}|}{|\{total\ retrieved\}|}$
2. $recall = \frac{|\{relevant\ reason\} \cap \{total\ retrieved\}|}{|\{relevant\ reason\}|}$
3. $f_{score} = 2 \times \frac{precision \times recall}{precision + recall}$

For the task that I am studying, a higher recall is more important, since I am more interested in classifying all the labels correctly from the ones I predict. However, a very low precision would help me debug and fine-tune because it would suggest that my model is not predicting a certain reason for rental. Lastly, the f-score helps me see the balance between the recall and precision.

The primary features that I will use in this dataset include the text review, category, and reason for rental. The majority of work came through processing the textual data, experimenting with the structure of the text, and fine-tuning parameters. This includes basic text processing, and obtaining optimal lambda values. Achieving such values were achieved through grid search. I am using an n-grams model, so I vectorized words of varying lengths, and compared them to my bag of words and baseline model in the results section.

Since the dataset contains inconsistent JSON values, I removed entries that had null values, and any data that had potentially dangerous effects on the results, such as extremely short reviews, or information that contained false information (exceptionally high ages, weights, heights). The number of reviews I will be working with are 152,670.

IV. Model

The model that I used is a bag of words model and an n-grams model. These are effective models for exploratory text analysis. Since this dataset is heavily text based, I thought it would be a good starting point because these models account for individual words that comprise the reviews, which can easily be overlooked if one were to compute basic properties such as length of a review.

Both models use maximum likelihood estimation, and due to my unfamiliarity with Markovian Chains, which are used to minimize the loss function in multi-logistic regression, I failed to derive a loss function and run gradient descent on it. Thus, in order to optimize my model, I resorted to reducing the dictionary sizes with meaningful words, and fine-tuned parameters. I had to take a naive, brute force approach through grid search, and experimented the effects of removing different words on accuracy.

While implementing these models, I had trouble in fine tuning the parameters. Since these models are relatively expensive operations, each iteration took a couple minutes, and this quickly added up when I performed grid search to find near-optimal lambda values and dictionary sizes. Furthermore, I noticed that with relatively small lambda values, I was overfitting the data, since the model was not sufficiently being penalized for its complexity.

Even though this dataset is heavily text-based, it would be reckless to neglect the other information in this dataset. Because of this, I also used regular logistic regression, and incorporated the vector obtained from the bag of words or n-gram model as one parameter in the logistic regression vector. A benefit of logistic regression is how I can combine information, such as the category of the clothing in an integer format along with other variables. This is more realistic to real world explanations, because a combination of variables usually explain a certain phenomenon, as opposed to just one variable, such as text.

Given that the categorical baseline model achieved relatively acceptable accuracy, and the bag of words and n-grams achieved even greater accuracy (refer to results section), combining both together would likely yield a better model. A downfall of this combined model is its computational expense. Finding a unique combination of variables requires extensive processing power, and given my limited and slow computational resources, I evaluated a single, combined model in the results section to demonstrate how multiple variables in conjunction can yield to better results.

Below are some feasible models that I thought would be worth exploring for this task, in comparison to the model that I focused on for this paper.

Table 2: Bag of Words and N-Grams

Pros (+)	**Cons (-)**
Effectively utilizes all textual data.	Computationally expensive.
Can be used with other models.	Ignores other potential predictive factors.

Table 3: Naive Bayes

Pros (+)	**Cons (-)**
Easy and fast to implement, with basic probability calculations.	Underlying assumption that features are conditionally independent, which in this dataset is highly unlikely.
When the assumption is met, speed and accuracy are significant.	Can potentially double count instances.

Table 4: Logistic Regression

Pros (+)	**Cons (-)**
Can use a combination of variables, and incorporate other models, including the model being studied in this paper.	Computationally expensive, takes a while to run and fine-tune.
Easy to implement, general purpose recommender system for categorical data, and can obtain a level of confidence for each prediction.	By itself, cannot meaningfully dive into textual data without using a model like a bag of words or n-grams.

Table 5: Similarity Based Metric (Jaccard)

Pros (+)	**Cons (-)**
Little machine learning involved, simple to compute.	An optimized similarity metric is still computationally expensive.
User pairs, item pairs, or user item pairs can be very predictive, especially in shopping data.	Variables can be further utilized, and similarity by itself can be insufficient.

V. Related Literature

Clothing datasets that I have explored usually take one of two forms. The first is data relating to a clothing product, including basic information about the user and basic properties of the clothing such as the size, category, price, and some form of review text (Ref. [1], Ref.[4]). The other form usually includes a vector of pixels used to describe an image (Ref. [2]). With the advancements of computer vision and deep learning algorithms, the dataset of pixels becomes more relevant in learning patterns in these images, and using these previously seen patterns as recommendations. The idea is that neural networks are utilized to learn what combinations of outfits go together, and based on extensive training, will suggest popular, previously seen patterns.

For ref. [1] and ref. [4], the data that comes with the clothing can be studied to create its unique set of recommendations, such as using previous reviews to learn about user sentiment and preferences, and utilize this information to recommend another article of clothing. Furthermore, looking into user demographic can reveal potentially correlated variables that influence an effective prediction, or reveal innate user biases.

The dataset that I utilized was used to create a recommender system on the fit of the clothing (“small” , “fit”, “large”). However, ref. [3] addresses the complexities in the dataset, such as imbalances in labels and user biases in what defines a good fit. As a result, a latent factor model was developed to account for such biases, including sensitivity to fit, and the inherent clothing features that likely result in such descriptions.

Ref. [3] further discusses two classification approaches used with the latent factor model, including logistic regression and a modified nearest neighbor approach (LMNN) which attempted to address the label imbalance issues. The LMNN approach with the single latent factor model yielded better results , but the logistic regression worked well with both a k-latent factor model and a single latent factor model. This paper reveals some gaps in my model that I did not consider, such as scrutinizing the data for potential bias, and using this to drive or add onto my model. However, the paper illustrates the effectiveness in integrating multiple models, as seen by combining the latent model with logistic regression. This is a concrete example of how using one model is very limited, and how using multiple models represent the complexities of real phenomena.

Most scholarly articles that I read focus on the fit of the clothing, as well as effectively recommending categories of clothing to the customer. Ref. [5] and ref. [6] discuss various techniques that have been deployed in the field, including image processing to extract features such as color and the position of the person in the picture, or modifications on more advanced natural language processing methods such as BERT, which use neural networks and deep learning to understand the context of the word by processing the entire sequence of words at once. This is different from the directional n-gram model that I am using which only processes the left and right word. Such a model is limited because it only briefly touches the importance of context. BERT models, on the other hand, try to fully understand context, which likely increases its predictive capabilities. In summary, these papers emphasize how effective and careful data collection, meta-data analysis, and clustering can be integrated to generate an effective and improved model, as opposed to focusing on just one.

Conclusions from the existing work suggest that any effective recommender system often goes beyond a single model, explores different variables in combination, organizes the information cleverly, and accounts for dataset and user biases. Upon reading these articles, the articles served as a concrete example of the limitations in focusing on just one model or one instance variable, and serves as an extension to improve my findings. Simply put, the power does not come just from the models. The power comes from how I cleverly use them, and make decisions based on their strengths and limitations. Furthermore, such a model would be useless if the data is not fully understood from the beginning, which illustrates the importance of understanding the user demographic, and hypothesizing about how certain data was generated.

VI. Results

These results were yielded from a dictionary size of 4000, and a lambda value of 10^{-1}. Basic text processing removed common, meaningless words and punctuation for the bag of words, and just punctuation for the n-grams to maximize the accuracy.

There are a total of eight categories to predict. I approximately used an 80:10:10 split for my training, validation, and test set. After fine-tuning on my validation set, I re-trained on the combined training and validation set. Below are results from my experiment on the test set, rounded to the nearest tenth. To understand the percentages better, I tested on 10,000 testing points.

Table 6: Baseline Accuracy

Baseline Model One	**Baseline Model Two**
31.3 Percent	37.7 Percent

Table 7: Text Models

Bag of Words	**N-gram, n=2**	**N-gram, n=3**
51.7 Percent	47.1 Percent	41.2 Percent

Table 8: Combined Model (Category + Text)

Bag of Words	**N-gram, n=2**	**N-gram, n=3**
53.1 Percent	47.9 Percent	42.8 Percent

Based on the above tables, my models had relatively good performance compared to the baselines. Considering that random chance results in a 12.5 percent accuracy, and the stronger baseline uses category as a prediction factor, it really suggests how utilizing the text can be a powerful tool in predicting the reason for rental.

The combined model tied the stronger of the baselines with the text models. This model studied the variable of category in combination with text. As seen by Table 8, the performance yielded better results than the baselines and the text models, revealing how the rental for reason is defined beyond just the text.

Below are the error metrics for each model. Generally, I was really satisfied with the recall on the popular reasons for rental, but unsatisfied with the lower recall and precision on the less popular reasons. Due to the label imbalances, there was more data training on popular instances such as “wedding”, than on less popular instances such as “vacation”. This resulted in poorer predictions, recall and precision for the less popular reasons.

Table 9: Error Metric for Bag of Words

Reason For Rental	Precision	Recall	F1	Total Instances
Wedding	0.221	0.765	0.344	2895
Formal	0.117	0.545	0.193	2152
Party	0.078	0.427	0.132	1827
Everyday	0.049	0.557	0.090	879
Work	0.038	0.451	0.070	847
Other	0.012	0.153	0.023	805
Date	0.007	0.167	0.013	389
Vacation	0.003	0.160	0.006	206

Table 10: Error Metric for N-gram, n=2

Reason For Rental	Precision	Recall	F1	Total Instances
Wedding	0.210	0.734	0.326	2974
Formal	0.100	0.474	0.164	2063
Party	0.072	0.385	0.121	1829
Everyday	0.053	0.556	0.096	931
Work	0.030	0.357	.056	812
Other	0.009	0.119	.0169	746
Date	0.006	0.135	.0106	420
Vacation	0.004	0.018	.0008	225

Table 11: Error Metric for N-gram, n=3

Reason For Rental	Precision	Recall	F1	Total Instances
Wedding	0.213	0.716	0.328	2853
Formal	0.091	0.442	0.151	2098
Party	0.051	0.281	0.087	1860
Everyday	0.046	0.496	0.084	946
Work	0.0178	0.219	0.329	843
Other	0.006	0.075	0.010	765
Date	0.003	0.062	0.005	408
Vacation	0.001	0.004	.0002	227

A future extension can be rewarding certain training points, such as adding weights to each word in the review if the review's reason for rental is less popular. Furthermore, given that my n-grams model was not as effective as I wanted it to be, I can utilize an n-grams model for sentiment analysis. I can explore whether there are different sentiments for each reason of rental, and use this as a predictive factor in my model.

Overall, representing the text reviews in grams of size two and three resulted in mediocre performance. While removing common words helped my bag of words model, this action resulted in worse performance in comparison to the n-gram model that was not stripped of common words (table 7). The feature representation that worked best were vectors of the following form: vector(bias term,count of seen words, encoded category). Feature representations that did not include the encoded categories, and instead counted the seen grams performed noticeably worse.

Comparing the Bag of Words model to the N-gram model, I was relatively surprised how the structure of the text led to decreased performance. My hypothesis on why the bag of words model was more effective than n-grams is because each reason for rental is likely dominated by the presence of key words, such as "bride" for wedding. This information is lost in the n-gram model, which can tie keywords with less predictive words. As a result, the parameters that yield the best result are often heavily dominated by the presence of correlated words and categories. As stated by Occam's Razor, oftentimes the "simplest model is the best model."

To state that my model was successful or unsuccessful is nuanced. My text model was successful in illustrating how text is a very powerful feature, since focusing on purely text models resulted in a significant increase in accuracy. However, my text models were also unsuccessful since it was relatively limited by just looking at one instance variable, as opposed to looking at others in combination (illustrated by my combined model). It was also unsuccessful because I did not carefully deal with label imbalances, which likely hindered the model from reaching its full potential. In general, the text models that I studied here are a great way to develop a strong, baseline recommender system. Fine-tuning and expanding ideas presented in this paper will result in a significantly better recommender system.

VII. Conclusion

In this paper, I explored how text can be utilized in making an effective recommender system. The results were significantly better than the baselines, yet had much room for improvement. Through the experiments and the related literature, I learned the limitations of relying on one model, and how models can be improved by utilizing other variables and models in combination. Furthermore, without a proper and careful analysis on the initial dataset, I neglect potential feature biases that can develop a strong recommender system. A lack of careful analysis can also jeopardize my overall performance, as seen by not compensating for the unbalanced labels. This ultimately affected the training process, which is where all the learning occurs.

VIII. References

[1] Project, UCSD CSE Research. “Rent the Runway.” *Recommender Systems Datasets*, cseweb.ucsd.edu/~jmcauley/datasets.html.

[2] Research, Zalando. “Fashion MNIST.” *Kaggle*, MIT License, 7 Dec. 2017, www.kaggle.com/zalando-research/fashionmnist.

[3] McAuley, Julian, et al. *Decomposing Fit Semantics for Product Size Recommendation in Metric Spaces*. Oct. 2018, cseweb.ucsd.edu/~jmcauley/pdfs/recsys18e.pdf.

[4] Project, UCSD CSE Research. “Amazon Review Data .” *Amazon Review Data*, 2018, nijianmo.github.io/amazon/index.html.

[5] Kotouza M.T., Tsarouchis S., Kyprianidis AC., Chrysopoulos A.C., Mitkas P.A. (2020) Towards Fashion Recommendation: An AI System for Clothing Data Retrieval and Analysis. In: Maglogiannis I., Iliadis L., Pimenidis E. (eds) Artificial Intelligence Applications and Innovations. AIAI 2020. IFIP Advances in Information and Communication Technology, vol 584. Springer, Cham. https://doi.org/10.1007/978-3-030-49186-4_36

[6] Corona, Humberto. “The State of Recommender Systems for Fashion in 2020.” *Towards Data Science*, 1 Oct. 2020, towardsdatascience.com/the-state-of-recommender-systems-for-fashion-in-2020-180b3ddb392f.