Clothing Type Classification from Description

Anna Dinov
UC San Diego
adinov@ucsd.edu

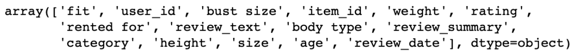
Christopher Hughes
UC San Diego
clhughes@ucsd.edu

Abel Seyoum
UC San Diego
aseyoum@ucsd.edu

Exploratory Data Analysis

The dataset that we had decided to use for this assignment was a clothing fit dataset used in the "Decomposing fit semantics for product size recommendation in metric spaces" paper written by Rishabh Misra, Mengting Wan, and Julian McAuley. In their paper, they had proposed a new way to better predict an online consumer's product fit in order to increase customer satisfaction and reduce a product's return rate. The data set they had used was collected from large clothing retailer Rent the Runway, and includes various information on clothing that people had rented from the retailer. In total, there are a total of 192544 rows and 15 columns (data.shape = (192544, 15)), making for a total 2888160 elements in the dataset.

As mentioned previously, there were a total of 15 different columns within this dataset, with information on fit, user id, bust size, item id, and a couple others.



```
array(['fit', 'user_id', 'bust size', 'item_id', 'weight', 'rating',
       'rented for', 'review_text', 'body type', 'review_summary',
       'category', 'height', 'size', 'age', 'review_date'], dtype=object)
```

List of Columns in Dataset

The first thing we investigated within this dataset was to investigate for any null values contained within the dataset. Fortunately for us, the columns we were interested in were already fairly clean, and resulted in us not having to do much work in regards to cleaning null values. The particular columns we were interested in were being able to predict the category of a clothing item using the review text column as a feature. In total, there are 68 unique categories of clothing items within this dataset, the most popular being dresses(~48.2%), gowns (~23.0%), and sheaths(~10.0%) (look at fig 1 for a visual of the 20 most popular categories of clothing to rent from Rent the Runway).

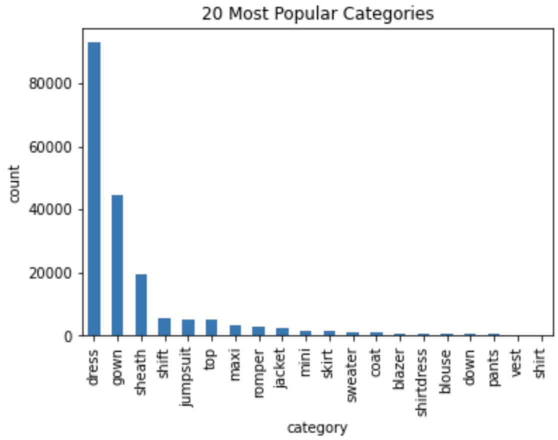
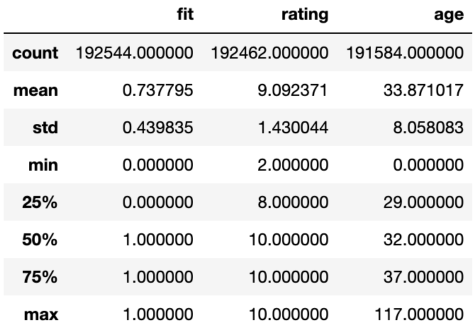


Figure 1

As can be seen from the visual, dresses are a very popular category of item for Rent the Runway, followed by gowns, sheaths, and shifts. Additional observations that were made from the dataset was that the average rating of an item from Rent the Runway per our dataset was 9.092371 / 10, the average age of a consumer from Rent the Runway was approximately 33 years of age, and the percentage of items that fit consumers from Rent the Runway was approximately 74% of items from our dataset, implying that approximately 26% of the items purchased did not fit the consumers. For more information on qualitative columns, please refer to table 1.



	fit	rating	age
count	192544.000000	192462.000000	191584.000000
mean	0.737795	9.092371	33.871017
std	0.439835	1.430044	8.058083
min	0.000000	2.000000	0.000000
25%	0.000000	8.000000	29.000000
50%	1.000000	10.000000	32.000000
75%	1.000000	10.000000	37.000000
max	1.000000	10.000000	117.000000

Table 1

For our next bit of exploration, we were curious into looking at the distribution of events that people utilized Rent the Runway for. We found that the highest event that people used this platform for was for weddings. For more information on this, please refer to figure 2.

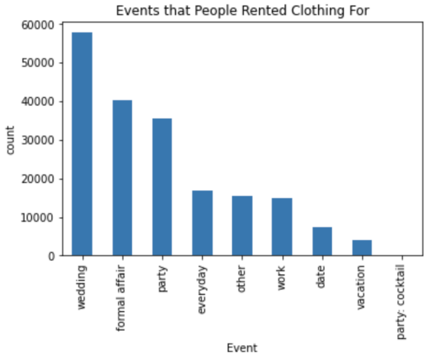


Figure 2

Another column we were interested in looking into was the review_tex column. Users provided a review about the product that they had rented from Rent the Runway. Much like the category column, there also were not any null values within the review text, thus the cleaning process was fairly simple for this dataset. The average length of a review per user for this dataset was approximately 301.6 characters. Ideally, our model would utilize this column in order to predict which category a piece of clothing would fall into based on its review. Upon completing our initial analysis of the dataset, we believe we are now ready to begin building a model in order to predict the category of an item.

Predictive task

We decided to attempt to predict the category of clothes based on the review of the clothing item. We really enjoyed the similar task given to us in assignment 1, where we predicted book genre based on the review of the book, so we decided to evaluate a similar predictive task. We decided to do this predictive task because we believe that the review of each item gives us a strong enough idea to build a predictive model off of. Our plan is to evaluate our model for the predictive task by checking the accuracy of our model by comparing our predicted categories to the true categories of each item which are located in the dataset. To do this, we will have our model predict the category of each review in the data set, and we will see which yield correct results when compared to a list of each correct category. We have come up with a couple varying baselines to compare for our model. The first baseline is a model that checks the review text for any words that are the same as a list of the categories in our data set. If the review contains any words that are in the list of categories, it outputs that category as the category for that item, and if no words are the same as any of the categories then it outputs a random category as that item's category. In order to get the random category we compiled a list of every category given in the dataset, by looping through the whole dataset and adding every category to the list while also avoiding duplicates. This baseline model yielded a 0.354 accuracy, which was very low as we expected. We made a second baseline which was very similar, except we changed the randomization factor to output the most frequent category instead. We looped through each item in the dataset to find the most frequent category, which happened to be a "dress" by a large margin. This second baseline yielded a 0.413 accuracy which did improve a lot from our first baseline. We expected a higher accuracy on this baseline because it is simply an improved version of the first baseline. The use of adding the most frequent category accounts for a lot of error that occur when picking a random category in the first baseline. The accuracy on these two baselines was still very low, giving us lots of room to improve in our actual model. The main feature we will use includes the description of the clothing item, and we have decided to use a bag of words model with the use of unigrams, bigrams, and trigrams. Because our main feature was the description text, we had to process the data by putting it into a list of dictionaries containing every bit of information from each row. The dictionaries hold all the features of each item, including its review text and category which we decided to use. While processing this data, we also had to remove all punctuation from each description in order to make it very easy for us to get every word in the description and use that to attempt to predict the category of the item.

Model

We decided to build a logistic regression model built using the bag of words technique with unigrams. We decided to use this model because we saw good results in assignment 1 using this model for the category prediction task. This base model gave a lot of room for improvement, so we decided to try to get the best accuracy we could by playing around with it. We will optimize the model by using a regularization pipeline in order to find the best regularization constant that yields the best results. We will also try and optimize this model by increasing the dictionary size to an extent while also finding out if unigrams, bigrams, or trigrams works best with our model. We ran into one issue of overfitting when it came to our dictionary size. At a certain point, increasing our dictionary size started to decrease the accuracy of our model, so we stuck with a dictionary size of 10000 which seemed to fit the dataset pretty well even when we increased the number of elements we were testing on. The other models we considered for comparison were the baselines we introduced in the previous section as well as the bag of words model using bigrams, trigrams, and various combinations of all three. We had many unsuccessful attempts throughout the process of testing our model and that came up when we started to use bigrams and trigrams. Overall, our accuracy went down when we included bigrams and/or trigrams. For reference, the model with only bigrams had an accuracy of 0.58, the model with only trigrams had an accuracy of 0.54, the model with all three had 0.61, and the model with unigrams and bigrams combined had an accuracy of 0.62. Our best model was using strictly unigrams where we were able to get our accuracy up to 0.65. The strengths of the baseline models that we compared to our model was that it was extremely accurate when the category of the clothing item was present in the review of the item. If the dataset had more reviews that state the name of the clothing item, this model would have done much better than our model. However, this is the only strength of this model because it relies solely on this. Therefore, the baseline models have many weaknesses which are evident in this particular dataset because there are an overweighing amount of reviews that do not state the category name in the review itself. Another limitation of these models is that it looks for the first time a category is stated in the review, so if the review states a wrong category (if for example it was comparing it to a clothing item), then the baseline models would predict the wrong category. The main strength of the bigram and trigram models are that they can gather more information about the context of the words in the review, but this creates a slight weakness in this particular dataset. The main weakness of the use of bigrams and trigrams is that they can overfit with different kinds of datasets because they cannot generalize as easily as a unigram model.

Literature

The dataset we used originated from Rishabh Misra, Mengting Wan, and Julian McAuley's "Decomposing fit semantics for product size recommendation in metric spaces" [1]. The goal of this paper was to study improvements for size recommendation from fit predictions in order to increase customer satisfaction from the clothing they order. The dataset we used for our clothing type predictor was obtained by the authors of this paper from the online clothing website, RentTheRunWay (https://www.renttherunway.com/). This dataset was used in this paper, along with another set of data from the Modcloth website (https://modcloth.com/), The Modcloth dataset is very similar to the dataset we used in our project, the only difference being the website that the data was pulled from, however in this paper it was used for the same purpose. In order to find the best way to predict the clothing size a customer should purchase based on their self-reported measurements and body characteristics. Their findings were that K-dimentional latent variable models performed the best. They also found that the RentTheRunWay dataset has much less cold products and customers (defined as being products or customers associated with a very small amount of purchases), in comparison to the ModCloth dataset, which caused the changes in performance to be reflected much more drastically on the RentTheRunway dataset. The relatively low number of cold products and customers in the RentTheRunway dataset, which is the dataset we used for our project, are probably what made the accuracy of our model fluctuate so greatly during our testing of different models. We predict that if we perform the same model tests on the ModCLoth dataset, our findings would have been the same, but the changes in performance of the different models would have been less drastic. The majority of the literature on the task we worked on for this project, clothing prediction, is done using images to be classified into clothing type. State-of-the-art technologies for using images to classify clothing include deep learning or neural network methods. The paper "Fashion and Apparel Classification using Convolutional Neural Networks" compared the accuracy of clothing classification from images using the following 5 renowned neural network techniques, Vgg16, Vgg19, InceptionV3, Custom CNN, and Vgg-like [2]. Their findings showed that the most accurate model for this kind of clothing classification was InceptionV3, which is a type of convolutional neural network model.

This paper, along with many others, used the DeepFashion dataset for clothing classification [3]. This dataset contains 800,000 plus images of clothing from online websites and photos from customers with 50 different categories and 1,000 other descriptive features. The DeepFashion dataset is one of the best datasets to use to create an accurate clothing classifier, which is clear from the extensive meta-data and the fact that it is used in so much literature on clothing classification.

In addition to clothing type classification, it is important to touch on some of the extensive literature covering the topic of fashion trend prediction. Predicting fashion trends can be extremely useful for clothing companies to continue producing products with high customer satisfaction. Future trends in fashion can depend on a multitude of features, including weather, seasons, pop culture and media, social media influencers, etc. This prediction topic often uses datasets in a similar way to how we used text from the RentTheRunway dataset to make predictions for the purpose of our project. Making predictions from text can be extremely useful for fashion trend prediction due to the fact that a lot of the popular media that may influence or inspire trends is written in words. The goal of the paper "Fashion Trend

Forecasting Using Machine Learning Techniques: A Review" was to evaluate and compare the state-of-the-art methods for fashion trend prediction [4]. The authors used the PRISMA checklist in order to evaluate what may be the best way to make these fashion trend predictions and they reviewed 73 publications on the topic. Big Data, Data Mining, Time Series, Greedy Method, and Machine Learning were the most common trend prediction methods that were encountered, with the vast majority covering machine learning for the predictive task. This indicates that the task of fashion trend prediction is heavily dominated by machine learning models. Specifically, the most common machine learning methods used for the prediction being deep learning and neural networks, as seen in much of the literature on clothes classification. Relating this back to our project, we used supervised machine learning classification in order to make predictions on clothing type. Variations on this approach that we took are seen in some of the literature reviewed in this article. Although we were not forecasting future trends in our project, it seems to be relatively backed up in this paper, and many others, that machine learning is an extremely useful predictive tool. ALthough we used a regression based model, as did a few of the reviewed papers, it is clear that if we were to continue to improve this project in the future, it may be useful to involve deep learning or neural network techniques for our predictive task since they seem to have performed so well for fashion trend prediction.

Results

We tested our model using many different attributes in order to see which one yielded the best results. The best accuracy we were able to achieve was about 0.65, which was significantly higher than the baselines, and considerably higher than the results we obtained with other attributes. We got this accuracy using only unigrams in our model, compared to our other models where we used only bigrams, only trigrams, both bigrams and unigrams, and all three at once. We were surprised that the unigrams model yielded the best results as we expected the model with unigrams, bigrams, and trigrams to result in the best accuracy. The other models had a decrease in accuracy when introducing bigrams, and they decreased even further when using trigrams as well. The significance of these results is that our model works best when only introduced to words by themselves rather than taking them in by chunks of two or more words. We also created a regularization pipeline in order to find the best c coefficient for our model, which came out to be 850. Representing our logistic regression model with a c coefficient of 850 as well as a significantly large dictionary size of 10000 yielded the best accuracy we could get. Our model could have improved slightly with a higher dictionary size, but it took too long to run our kernel to increase our dictionary size much more. We concluded that as dictionary size increased our accuracy would increase to an extent because by having access to a larger bank of words we could have more to predict off of. The only downside to this is the speed at which our model ran, which took about 25 minutes with a dictionary size of 10000. Our proposed model did succeed compared to our baselines and other representations of our model. It did better than both baselines and our model when using different parameters such as the introduction of bigrams and trigrams. We believe that this model succeeded because single instances of words in our specific review texts for each item tended to give more input about the specific category of that item compared to chunks of words in the forms of bigrams and trigrams. Because of this, our model yielded a higher accuracy when built off of unigrams alone, assuming that our constants and dictionary size were at their best respective values for our dataset. Overall, we have concluded that we were able to create a fairly accurate predictor of clothing category based on description of the item, however there is still a lot of room to grow in terms of creating the best possible predictor. Perhaps investigating other features in order to predict categories may have resulted in a higher accuracy in our model. In addition, perhaps another way in order to improve our accuracy significantly would have to be creating a whole new predicting model than from the techniques we have learned in this class.

References:

[1] Decomposing fit semantics for product size recommendation in metric spaces
Rishabh Misra, Mengting Wan, Julian McAuley
RecSys, 2018

[2] Fashion and Apparel Classification using Convolutional Neural Networks
Alexander Schindler, Thomas Lidy, Stephan Karner, Matthias Hecker
arXiv:1811.04374v1, 2018

[3] Large-scale Fashion (DeepFashion) Database
Ziwei Liu, Ping Luo, Shi Qiu, Xiaogang Wang, Xiaoou Tang
Multimedia Laboratory, The Chinese University of Hong Kong, 2016

[4] Fashion Trend Forecasting Using Machine Learning Techniques: A Review
A. Chang, et al., 2021