E-Commerce Clothing Data Analysis and Recommender

Siqi Chen **Wen-Hsuan Hung** **Yuefeng Mao** **Xinyu Lou**

Abstract

In this project, we conduct analysis on the rating and fit of customers based on the e-commerce clothing data collected from *RentTheRunway.com*, and then develop a recommender to return similar items given the user inputs. From the model results, we conclude that review texts could largely improve the performance of classifiers which only includes customer demographics and body measurements. Other than that, word embedding models such as Skip-gram and CBOW which focus on learning from context perform best among a series of text mining models.

1 Introduction & Literature Survey

1.1 Introduction

In the year 2020, market research estimated that the online clothing rental market had reached a size of over 1.19 billion USD, and the market was expected to grow and reach 2.29 billion USD by 2028.

With more potential in the growing market size, and the fact that clients cannot have physical access to the rented clothing items, rating and clothing fit predictions estimated based on users' body measurements have become an increasingly important feature for the business. More accurate predictions will improve the customers' satisfaction rate and therefore increase the profit for the business.

In this project, we will use different word embedding techniques to process the review texts in the RentTheRunway dataset, and try to conduct exploratory analysis on distributions of customer information, and predictive analysis on customer ratings and fits based on the processed text data and other body measurements data. We will compare results using Logistic Regression on 50,000 randomly selected entries using different text processing techniques: A,B,C,D…. In each model, we will also compare the results of just using body measurements with also implementing review texts to see how text processing elevates the results. Our prediction can serve as an improvement measurement for different text processing models. In our project, we will also implement an interactive feature for existing customers. We will analyze past customer and item interactions, and generate the top 10 recommended items, corresponding categories, and 'rented for' occasions, given any existing customer ID.

1.2 Literature Survey

Misra, Wan and McAuley conducted related research regarding clothing fit and body measurements in their paper Decomposing Fit Semantics for Product Size Recommendation in Metric Spaces in 2018. In this research, they applied Latent Factor Model to analyze the sentiments in customers' review texts, and adopted a technique called "metric learning" to address the imbalance between different labels. According to their study result, Latent Factor Model with K dimensional latent variables showed a better outcome than the method with only one latent variable. The study used complex models with neural network methods and compared outcomes on multiple datasets. However, the study only processed the review text in a certain way, while we want to see how different word representation techniques can affect the model performance.

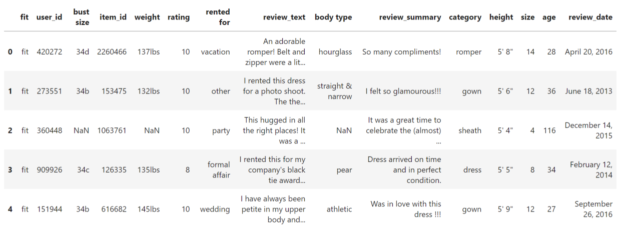
Chen, Yen and Tian also studied the relationship between online business review content, user ratings, and user recommender system. In this study, they also adopted several machine learning models including K-Nearest-Neighbor, Quadratic Discriminant Analysis, Support Vector Machine and Decision Tree to compare the prediction performance on their testing data. However, their study did not demonstrate the possible correlation between different features and user ratings. Also, when predicting possible recommendations from customers, they introduced a rating variable to generate a better-fitting model on the testing data, which the rating variable was already a strong indicator of rating preferences, thus overlooking other customer measurement data like heights, weights, and body sizes.

2 The Dataset

2.1 Data Preprocessing

In this project, we use open-sourced clothing fit data which is collected from *RentTheRunway*. The RentTheRunway dataset contains 192,544 records from 105,508 users regarding 5,850 items.

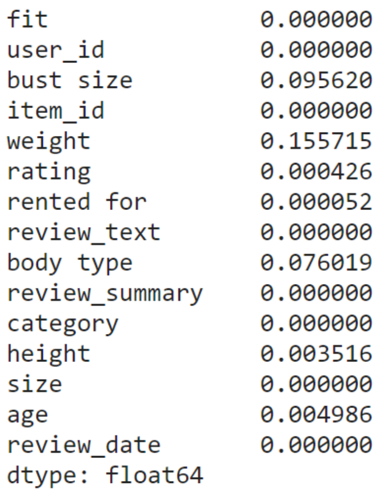
Our original source of data is in json format. To better manipulate data and conduct further analysis, we convert it to a pandas dataframe. As Figure 1 shows, the raw dataset contains missing values and some of its body measurement indexes are not in numeric formats.



	fit	user_id	bust size	item_id	weight	rating	rented for	review_text	body type	review_summary	category	height	size	age	review_date
0	fit	420272	34d	2260466	137lbs	10	vacation	An adorable romper! Belt and zipper were a lit...	hourglass	So many compliments!	romper	5' 8"	14	28	April 20, 2016
1	fit	273551	34b	153475	132lbs	10	other	I rented this dress for a photo shoot. The the...	straight & narrow	I felt so glamourous!!!	gown	5' 6"	12	36	June 18, 2013
2	fit	360448	NaN	1063761	NaN	10	party	This hugged in all the right places! It was a ...	NaN	It was a great time to celebrate the (almost) ...	sheath	5' 4"	4	116	December 14, 2015
3	fit	909926	34c	126335	135lbs	8	formal affair	I rented this for my company's black tie award...	pear	Dress arrived on time and in perfect condition.	dress	5' 5"	8	34	February 12, 2014
4	fit	151944	34b	616682	145lbs	10	wedding	I have always been petite in my upper body and...	athletic	Was in love with this dress !!!	gown	5' 9"	12	27	September 26, 2016

Figure 1: First 5 rows of raw dataset

To handle these issues, first of all, we check the proportion of missing values in each column.



```
fit               0.000000
user_id           0.000000
bust size         0.095620
item_id           0.000000
weight            0.155715
rating            0.000426
rented for        0.000052
review_text       0.000000
body type         0.076019
review_summary    0.000000
category          0.000000
height            0.003516
size              0.000000
age               0.004986
review_date       0.000000
dtype: float64
```

Figure 2: Proportion of missing values

From Figure 2, we can see that 6 features contain missing values. For these columns, we fill in their modes to replace missing values. And for feature columns which we expect to have numeric, we transform them to their corresponding units. For *bust size* and *weight*, we extract the number part as feature values, and create a new variable cup to store the letters after the number in *bust size*. And for *height*, since the original values are in feet and inches, we extract the numbers of feet and inches, convert 1 feet to 12 inches, and recalculate these values in inches to obtain the height data in numeric format.

And for text variables, we use nltk package to conduct tokenization and stemming operations for them, then we convert tokens to lower cases and remove the stop words. Specifically, we find the list of stop words in nltk.corpus package does not contain some frequent stop words in review text, so we manually added about 30 words into this list to keep it consistent with our data.

2.2 Exploratory Data Analysis

To explore the characteristics of each column, we conduct univariate analysis for features and response variables.

For our 4 response variables *fit, rating, rented for,* and *category*, since *rating* only have 5 levels of response value and the other 3 are all categorical variables, we create some barplot to explore their distributions.

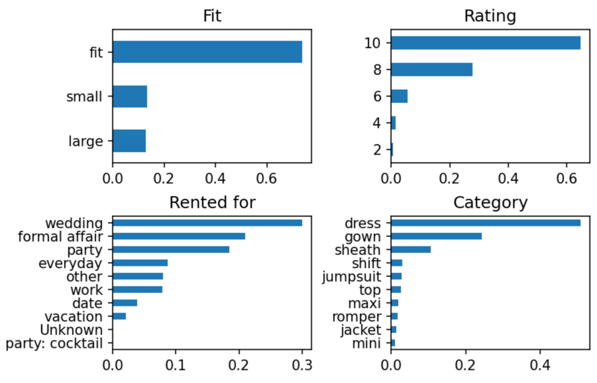


Figure 3: Proportion of variables

From Figure 3 we find that for *fit*, more than 70% of customers think the clothes that they get fits all, and the probabilities of being small or being large are about 10% for each level.

For *rating*, 10 (the highest rating) has the highest proportion of about 60%, and as rating decreases, the proportion also decreases. These 2 variables indicate that most customers are satisfied with the products and services offered by *RentTheRunway.*

And from the distribution of variables *rented for* and *category*, we find that most popular categories are formal dresses which are required for some public affairs. It makes sense since *RentTheRunway* offers clothes sharing services. For daily dressing which has higher re-use probability, people are more likely to make purchases instead of renting.

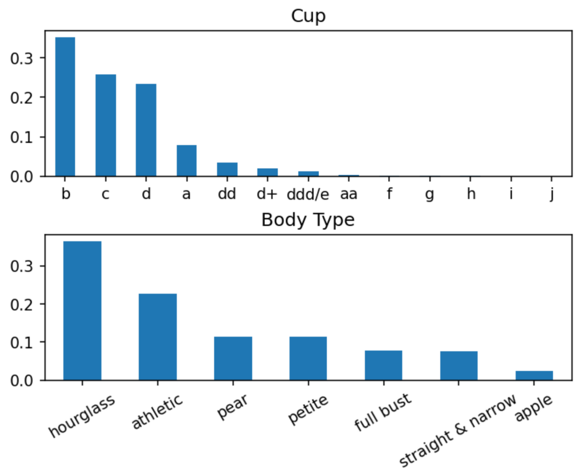


Figure 4: Proportion of categorical features

Then for features, we use barplot for categorical features and boxplot for numeric features. In Figure 4, we can see that most customers have medium body shapes. It may be because these kinds of people could find more suitable clothes on a cloth-sharing website and the website also has a willingness to offer more clothes in medium sizes since they will have higher turnover rate and generate more profit.

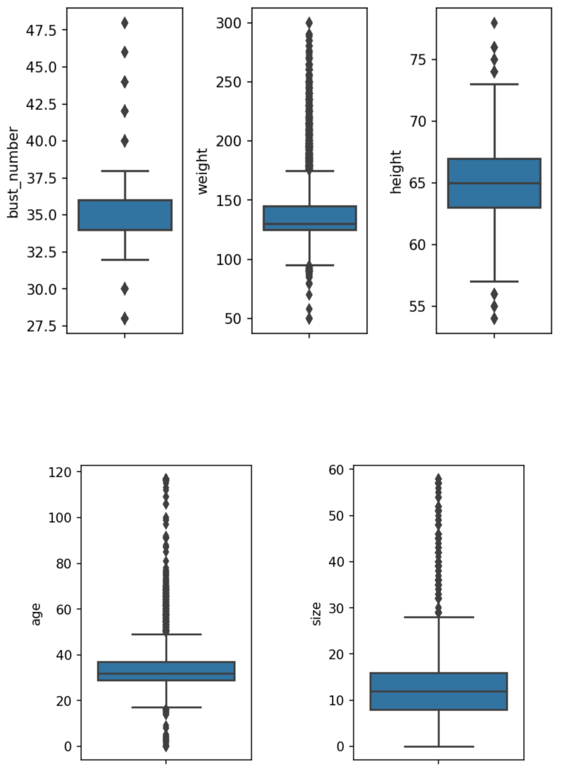


Figure 5: Distribution of numeric features

In Figure 5, we use boxplot to explore and examine the distribution of numeric features. Specifically, what we should pay attention to is the distribution of *age*. For variable *age*, we find there are many outliers and the range of *age* is [0,120], which is obviously abnormal. We assume that some customers are unwilling to offer their actual ages and inputs below 15 or above 70 may be fake ages. We replace these values with mode to make the variable more likely to obey a convincing distribution.

Besides univariate analysis, we also generate pairplot for bivariate analysis.

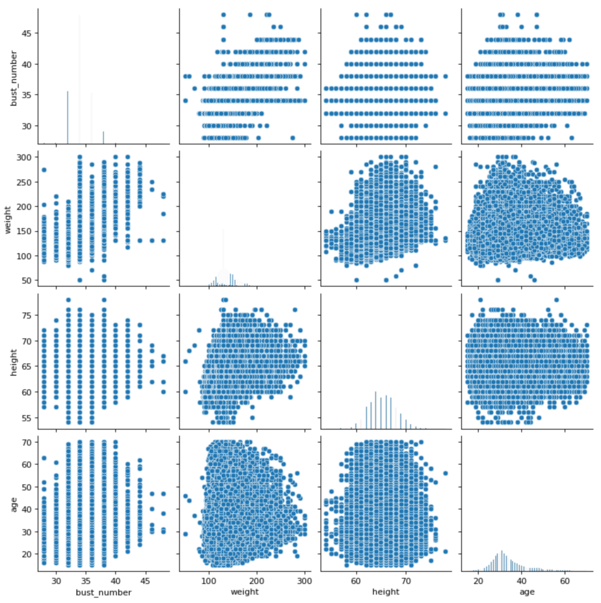


Figure 6: Pairplot of numeric features

From Figure 6, we find that among 4 features *bust_number, weight, height*, and *age*, variable *weight* shows some positive correlations with *height* and *bust_number* from the scatter plot. To further examine the correlations, we build a correlation matrix of these variables. In the correlation matrix, all pairs of variables show positive correlations, which is consistent with the actual fact that people who are taller or have larger bust numbers will have higher weights. However, the correlation between bust_number and weight is noteworthy since it is rather high. In our modeling part, we will keep track of these 2 variables to check if we should do some adjustments.

Then, we perform the VIF test on all variables as a whole, and the obtained VIF value is 2.28, indicating that there is no significant multicollinearity among the variables. Therefore, we keep all the current features.

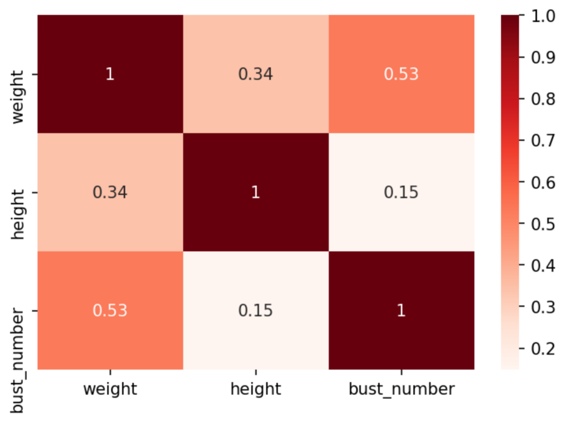


Figure 7: Correlation matrix

The *RentTheRunway* dataset also contains 2 text variables, *review_summary* and *review_text*. Previewing the raw data, we find *review_summary* contains phrases extracted from *review_text* which include some key information such as cloth category and customer sentiments, but other customer features are omitted. Since we need text to generate word vectors and build our model, we choose *review_text* as our text variable since it could provide more comprehensive information. After tokenization, we build a word cloud to demonstrate the frequent words in our text.

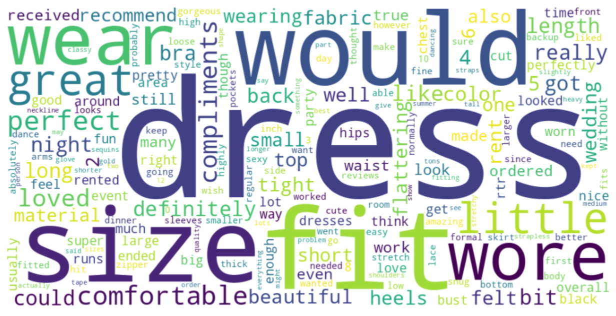


Figure 8: Word cloud of review_text

From our word cloud, “dress” is the most mentioned word, followed by words related to customer experience such as “fit” and “size”. Among these words, while “fit” is mentioned the most, “little” is also frequently brought up, which might be an indication of less satisfied customers. Also there are some frequent positive words like “great”, “perfect” and “loved”, which is consistent with the previous barplot that customers overall feedback in rating are rather satisfactory.

3 Research Problem

There are three goals of our research. First, we want to know whether adding review would increase model accuracy when predicting whether a clothes fit a person or not. Second, whether adding reviews variable would increase model performance when predicting customer rating. Third, we build a recommendation model which recommends clothing categories based on user input.

3.1 Input variables

For predictive models which predict *fit* and *rating*, the input variables include *bust number, weight, height, size, age, cup, body type, review*. For the recommender, the input is the item which a user rated the highest in the past.

For the reviews in predictive models, we try different preprocessing methods to convert text. These include converting reviews to vectors applying word2vec (Skip-gram & CBOW), GloVe, and Tf-Idf.

3.2 Output variables

The output variables for predictive models are *fit* and *rating*. For the recommender, the output variable is *item_id*, and its corresponding *category* and *rent for,* which means that the model would recommend top 5 items which have the highest similarity with the user input.

3.3 Baseline Model

The goal for our project is to see whether adding "text" would improve model accuracy, thus the baseline model is adding all variables except review. The variables include bust number, weight, height, size, age, cup, body type. The reason that the model adding text could outperforms the baseline is that first, we only keep important words in each row, and deleted pronouns, prepositions, and other common words, such as "I, you, he, she, when, where, what, and, also, to, oh, of", thus reduced the noise caused by not related words. Second, language models learn patterns from corpus, thus they are able to increase characteristics for each data entry.

3.4 Methods to compare and evaluate model

For *fit* predicting model and *rating* predicting model, we compare the performance of model using different NLP techniques including word2vec (Skip-gram & CBOW), GloVe, and Tf-Idf. To evaluate model performance, we compare AUC score, F1 score, and accuracy of each model. The higher the AUC, the better the performance of the model at distinguishing classes.

4 Proposed Solution

Since our purpose is to classify variables *fit* and *rating*, we try logistic regression without and with review text to clarify whether text is useful in this task.

To compare the effect of text on the model, we try different bags of words and word embedding methods to process review text. Bag of words can build a vocabulary from a corpus of documents and counts how many times the words appear in each document. To put it another way, each word in the vocabulary becomes a feature and a document is represented by a vector with the same length of the vocabulary. This approach causes a significant dimensionality problem: the more documents you have the larger is the vocabulary, so the feature matrix will be a huge sparse matrix. Therefore, in order to use this way more efficiently, we do a lot of important preprocessing like word cleaning, stop word removal, stemming to reduce the dimensionality problem.

Word embeddings are methods which look for a latent space to preserve the semantics as much as possible. Words from the vocabulary are mapped to vectors of real numbers. These vectors are calculated from the probability distribution for each word appearing before or after another. These methods are able to capture many linguistic regularities of words as well as taking their context into consideration.

To be more specific, we try binary, frequency, TF-IDF, word to vector, GloVe and CBOW to process data and retrain the classification models. Then we compare the models' accuracy, auroc_rate, F1 score(both macro and micro) and choose the best one as the with-text result. Then we compare it with the model's result without text and to find the improvement of adding text.

Finally, to make our project more useful, we design a recommender system based on existing information. The recommendation system is constructed based on the ratings given to the item by existing users and the Jaccard similarity. When existing users enter their id, they can see the top 10 products recommended by the system based on their ratings. In addition, we also output the categories of these ten products and the occurrences used, so that customers can choose according to their own needs.

5 Experimental Results

5.1 Prediction Model Comparison

To reduce the cost of training and tuning, we randomly shuffle the original dataset and pick 50,000 records as our dataset for model comparison. Among these 50,000 records, we split them into a training set with 40,000 records and a testing set with 10,000 records. Table 1 shows the evaluation scores of different fitting models on the testing set.

Table 1 Results comparison of fitting model

	AUROC	F1-Micro	F1-Macro	Accuracy
Baseline	0.5866	0.7339	0.2846	0.7339
TF-IDF	0.7905	0.7731	0.6031	0.7731
Skipgram	0.7853	0.7669	0.4842	0.7669
CBOW	0.7948	0.7708	0.5082	0.7708
Glove	0.7501	0.7553	0.4331	0.7553

As we can see from Table 1, all the 4 models show improvement from baseline, which indicates that review texts do provide more latent information for us to predict user fit. The performance of these models is not very different. However, in this project, we believe that AUROC can better reflect the pros and cons of the model in a balanced manner, so we finally choose the AUROC indicator to select the model. And among the 4 text models, CBOW has the best AUROC score.

Table 2 Results comparison of rating model

	AUROC	F1-Micro	F1-Macro	Accuracy
Baseline	0.5456	0.6506	0.1577	0.6506
TF-IDF	0.7883	0.6721	0.3477	0.6721
Skipgram	0.8226	0.6763	0.2688	0.6763
CBOW	0.8103	0.6646	0.2554	0.6646
Glove	0.7549	0.6496	0.1999	0.6496

And from Table 2, when predicting rating, text models also outperform the baseline model. Among these models, Skip Gram has the best AUROC score.

From above results, in general, word2v (CBOW and Skip Gram) has the best performance. Then, because the model effect of CBOW and skip-gram is not very different, and in this task, whether the clothes fit the customer is more important, finally we chose the CBOW model. We re-estimate this model on the whole dataset with 192,544 records and get the following results.

Table 3 Results of CBOW model on full dataset

	AUROC	F1-Micro	F1-Macro	Accuracy
Fit Model	0.7880	0.7684	0.5024	0.7684
Rating Model	0.8208	0.6740	0.2780	0.6740

As Table 3, the performances of CBOW model are similar on partial datasets as well as on the full dataset, which means our model is stable.

5.2 Recommender System Examples

In our recommender, based on cosine similarity, for each user within the dataset, we find 10 most similar items for them and return the categories and rented_for reasons as well. In Table 4, we randomly choose a user and show the recommendation for them.

Table 4 Recommendation Demos

User id	Recommended item id	Item category	Item rented_for reason
420272	2732613	shirt	date, everyday, party, other, work
	925104	dress	wedding, everyday, party, vacation, other
	1006590	dress	date, party, wedding
	940419	dress	date, wedding, everyday, party, formal affair
	962489	dress	date, everyday, party, vacation, other, work
	1601603	dress	date, wedding, party, formal affair, other
	2375866	skirt	vacation, party, everyday
	2363191	jumpsuit	date, everyday, party, formal affair, vacation
	1635675	dress	date, wedding, everyday, party, formal affair.
	2211789	romper	vacation, party, everyday, other

6 Conclusion and Outlook

In conclusion, we have the following findings from our analysis.

First, from the data collected from *RentTheRunway*, the majority of its customers have high evaluation of the services and products. The main reason that customers would like to use cloth sharing service is to prepare for some formal affairs, and most of the customers are of medium figure, which may be because this kind of customers are exactly whom *RentTheRunway* wants to target.

Second, baseline models of logistic regression which excluded text variables show poor performance in predicting user fit and rating. In Particular, the macro F1 scores of baseline models are extremely low, which suggests that the classifiers are affected by imbalanced data and have low accuracy in predicting the minority levels.

Third, Skip-gram and CBOW models show the best performance among all the text models involved in this project. Possible reason is that under this e-commerce situation, a user's fit and overall evaluation are much more likely to be mentioned in user comments, so these models which generate word embeddings based on the study of context outperform other models.

Also, our project still has some weaknesses which suggest the potential space for improvement.

First, when examining the impact of text compared with other features, we use only logistic regression models. Though models with text features added do perform better than logistic regression, if we parse original features into more complex machine learning models, they may generate better performance. In the future, we will consider comparing text models with other models to further study their impacts.

Secondly, when building text models, we only develop some simple frequency-based models and word embedding models. During the experiment, we tried to build a Bert model, however it failed to return expected outputs in time since it had higher requirements for devices. In the future, if possible, we are going to try more language models with advanced devices.

And for the recommender system we build, it returns recommendations based on existing records of certain users. However it does not solve the cold start problem that we cannot apply it to generate recommendations for new customers. To address this issue, we have proposed plans including introducing random numbers, returning average, or asking for more features from new customers that we can use KNN to find some similar users in our records and then apply our recommender system to them.

And lastly, for the deliverable, we are planning to create some more applicable forms to better demonstrate our models and let them be accessible to amateurs as well. In the future, we are going to build an interactive tool which allows inputs of body features and desired categories/situations to return customized recommendations for users.

7 Reference

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

[2] Hsin-Yu Chen, Huang-Chin Yen, Tian Tian (2020) *Female Clothing E-commerce Review and Rating*.