Rating Prediction for Users in RentTheRunWay

ZIKANG CHEN, University of California, San Diego
YILANG HE, University of California, San Diego
SHAOLONG LI, University of California, San Diego
XINGYIN XU, University of California, San Diego

Additional Key Words and Phrases: datasets, linear regression, latent factor model, collaborative filtering

ACM Reference Format:
Zikang Chen, Yilang He, Shaolong Li, and Xingyin Xu. 2022. Rating Prediction for Users in RentTheRunWay. *ACM Trans. Graph.* 37, 4, Article 111 (August 2022), 4 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 DATASET

1.1 Introduction

We used the clothing fit data from RentTheRunway[1] on Professor McAuley's website[2]. The dataset consists of customers' reviews and ratings of clothes and their body measurement, in addition to clothing fit, clothes size, and clothes category. The dataset contains 192544 data in total and contains 146,381 data after we removed the rows containing the null data.

Fig. 1. An Overview of Dataset

	fit	user_id	item_id	weight	rating	rented for	review_text	body type	review_summary	category	height	size	age	review_date	fit_sml
0	True	420272	2260466	137	5	vacation	adorable romper belt zipper little hard naviga...	hourglass	So many compliments!	romper	68	14	28	0.070136	1
1	True	273551	153475	132	5	other	rented dress photo shoot theme hollywood glam ...	straight & narrow	I felt so glamourous!!!	gown	66	12	36	-2.034953	1
2	True	909926	126335	135	4	formal affair	rented companys black tie awards banquet liked...	pear	Dress arrived on time and in perfect condition.	dress	65	8	34	-1.549788	1
3	True	151944	616682	145	5	wedding	always petite upper body extremely athletic im...	athletic	Was in love with this dress !!!	gown	69	12	27	0.392903	1

1.2 Data Cleaning

In this step, we first dropped the entries with null values. For numerical data, we got rid of the units in heights and weights, and converted them into numbers of inches and pounds. We also converted the ages from strings into integers. Since the ratings are 2, 4, 6, 8, and 10, we converted them into 1 to 5 for easier use. Next, we processed the reviews by converting them to lower case and removing punctuations and stop words. Since the data of bust size are inconsistent and presented in various formats, we decided to drop the bust size data. We used one-hot encoding to convert clothing fit into binary values of fit or not fit. If the clothes are either too big or too small for the user, we replace it with false.

1.3 Exploratory Data Analysis

After we performed an exploratory data analysis, we found that the average height is about 65 inches, the average weight is about 137 pounds, the average age is about 34, and the average size is about 11, the average rating is about 4.54. During the analysis, we found some extremely young people aged below 7 and extremely old people aged over 100, who might be too young or too old to give reasonable reviews or whose age might have been a typo. Because this data might become an outlier when fitting our model, we decided to only include people whose ages are between 10 and 80.

We also plotted the customers' heights per rating. We found that customers who give lower ratings on average have similar median height and median weight with customers who give higher ratings. Though the median looks similar, the distribution for five ratings has an overwhelming high count in median weight and height.

Furthermore, we noticed that about three quarters of customers think their clothes are fit, while others think the clothes are too large or too small.

We found that the most common words in reviews include "true size", "many compliments", and "loved dress". When we break down the reviews by clothing fitness, we found that people who think their clothes fit well usually write "true size" and "fit" in review, which is consistent with the most common words in general. However, it is inconsistent for people who think the piece of clothing is too small or too large because they only account for about 1/3 of the dataset. When we look at people who think their clothes fit small, they usually write "runs small" and "dress" in review and people who think their clothes fit large usually write "runs large" and "little big" in review. For those people who feel the piece of clothing does not fit, there is a clear indicator in their review that the piece of clothing is too big or small.

Lastly, we look at the rating distribution of the dataset. We found that about 64% of the ratings are 5, which is the highest grade. And 28% of people rate 4 and only 6% of people rate 3. The rest of the few people rate 2 and 1. This data shows that most people are satisfied with their orders, and only few are unsatisfied and give low ratings.

2 PREDICTIVE TASK

The predictive task we chose is to predict the users' rating. This task is useful because the rating prediction can give a guideline when the website recommend new items to users. At the same time, it can

[1]https://renttherunway.com/
[2]https://cseweb.ucsd.edu/ jmcauley/datasets.html

Authors' addresses: Zikang Chen, zic017@ucsd.edu, University of California, San Diego; Yilang He, University of California, San Diego, yih022@ucsd.edu; Shaolong Li, University of California, San Diego, shli@ucsd.edu; Xingyin Xu, University of California, San Diego, xix006@ucsd.edu.

0730-0301/2022/8-ART111 $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

Fig. 2. Distribution of Height

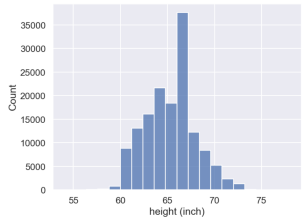


Fig. 3. Distribution of Weight

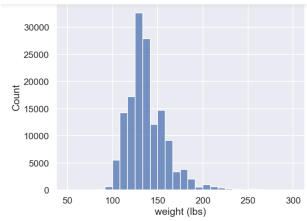


Fig. 4. Distribution of Size

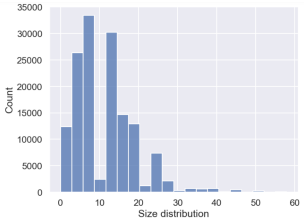
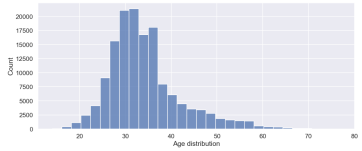


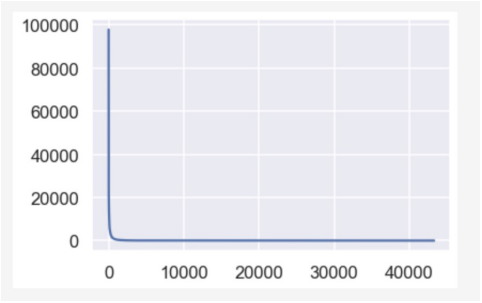
Fig. 5. Distribution of Age



help the platform to understand users' rating habits. The baseline model we created simply computes the average rating of each user and returns it. If the user hasn't rated anything yet, the model will return the global average rating of all users. The average rating is chosen because if the model predicts each rating as mean, it will provide us with the variance of the dataset, and the MSE tells us the variance of the error. So we should expect our model to have a lower value than this baseline. The mean squared error of our baseline model is roughly 0.624, the mean absolute error is about 0.581 and the R^2 is -0.199. To evaluate our model, we use mean squared error, mean absolute error and R^2 score. Here we will mainly focus on the MSE and R^2 score. Since we are predicting customers' rating which is a continuous value and directly reflects their attitude toward a product, we won't care too much about the small error but give large error great penalties. The MSE works for this purpose because it will ignore the small error, for example, the rating is 0.1 deviated from its true rating, the MSE will decrease this 0.1 into 0.01 by squaring it. MSE also magnifies the large difference, for example, if the rating is 1.1 deviated from its true rating, the MSE will magnify that 1.1 into 1.21 by squaring it. Thus, we choose the MSE as one of the metrics to measure the model. For our dataset, the rating is only on a 5-point scale. To remove the bias coming from the scale, we take a step further by using variance to normalize the MSE, which is R^2 score. Given the fact that our baseline simply computes the average rating of each user, it doesn't include any features.

In our improved model, we decide to use features including: height, weight, age, size, fitness and review. Since height, weight, age and size are numerical values, we can take these columns out from the data frame directly and convert them to arrays. These numeric values provide customers' body measurements. We then find the 1000 most popular words used by the users and use CountVectorizer to capture the occurrence of the 1000 most popular words in each review. This bag of words are important because they contain users' attitude that can directly reflect on their ratings. As we mentioned in the data cleaning section, punctuations and stopwords are removed because they contributed little information. First 1000 words with top frequency are chosen because they are more typical and representative due to the high appearance rate. At the same time, this bag of words contains words with relatively low frequency which are more distinctive. This helps to avoid every entry having too similar bag of words.

Fig. 6. Distribution of Word Counts(y) of Popular Word Index(x)



3 MODEL

3.1 Linear Regression

Given the fact that what we are trying to predict is a numerical value and our features are mostly numerical, linear regression is the first model we tried besides the baseline. In the first version of the linear regression model, our feature matrix includes all the numerical features including height, weight, age, size as well as boolean feature fit. We then use the train_test_split to split the features and labels into 50% data for training and 50% data for testing. After fitting the model on the training set and predicting the data in the test set, we got an MSE of about 0.476, an MAE of about 0.550 and an R^2 of about 0.068. All the errors are smaller compared with the baseline model. Though the scores are not very ideal, this model is much better than the baseline.

3.2 Optimization

To optimize the linear regression model, the main direction we are thinking is through feature engineering. In our first attempt, we only consider the user's numerical data and their objective feeling toward whether the clothes fit or not. In our second attempt, we take users' reviews into consideration. This is intuitive because users' reviews are the explanation and reasoning given to their ratings. Here we are using the bag of words with the first 1000 words with top frequency. The reasoning for this choice can be found in *2.Predictive Task* section. Through this change in features, we got an MSE of about 0.380, an MAE of about 0.468 and an R^2 of about 0.259. This is a huge improvement compared to the previous model without considering the users' reviews.

3.3 Issues

In addition, we considered some other combinations of features. For example, we include the maximum Jaccard similarity between the current item and the other items in the training set. This new feature doesn't improve the result compared to our previous model. This might be because the maximum similarity doesn't provide too much information about the current user's attitude. We may also need to include the users' attitude toward that item with the maximum similarity. However, even in that case, it may not help too much because different users have different body conditions. So others' attitudes contribute little to the current user's feeling because of the physical difference. And we will have an obvious risk of overfitting by adding too many features. This problem can be solved through adding regularization. This might be the new experiment we can consider if we have more time for this project.

Scalability is not a big issue for our models. The model can handle dataset with around 100,000 entries within a reasonable amount of time. Some improvements that we made is first dropping the entry with null value. This helps us to filter out more complete user information. Second, we use caching technique to improve the efficiency of our model. For example, when we go through the dataset to calculate items' similarity. We cache the result in a map and check if the result is stored in the map already to reduce duplicate calculation.

3.4 Other Models

We also tried other models, such as the latent factor model and the Jaccard similarity model. In the latent factor model, we went through the similar process as discussed in the lecture. We used the user and the clothes ID as features and let the rating be the data that the model will predict. We did not add any other features into the model because the unsupervised model would detect the characteristics of the person and clothes item by itself. For the loss function, we used the following equation:

$$\sum_u \beta_u^2 + \sum_i \beta_i^2$$

where i represents the cloth item and u represents the user.We choose our lambda to be 0.0001 by using a for loop to try different values of lambda. We also choose the best parameter k by using the for loop, which is 3. However, this latent factor model fails to solve the "cold start" problem, which means that if the user or the item never occurs in the training set before, the model fails to recognize any features of the user or item. By using the same training and test set for our best model, this model returns a MSE of 0.491, MAE of 0.561 and R^2 of 0.033.

In the Jaccard similarity model, we used collaborative filtering to predict the rating with the following equation:

$$r(u,i) = \frac{1}{Z} \sum_{j \in I_u \setminus i} r_{u,j} sim(i,j)$$

Specifically, we calculated the rating based on the ratings of user's past purchases and the corresponding Jaccard similarity of two items. If the user does not have any past purchases, the model will predict the global average rating. For this model, we got an MSE of 0.549, MAE of 0.558 and R^2 of -0.02. However, this model also fails to solve the "cold start" problem because we would only use the global mean to guess its rating.

3.5 Unsuccessful Attempts

We have some unsuccessful thoughts, especially in the starting phase. Initially, we are trying to predict the fit boolean and using that prediction to predict the final rating. Our reasoning is that user-provided data may not be accurate, considering the case that a user has small body measurements but chooses to fit with a large-size clothes. We finally aborted this plan because we think fit is a very subjective thought and rating is closely related to this subjective thought. So "objective" prediction of fit won't increase the accuracy of prediction but will backfire by introducing an extra layer of complexity.

4 LITERATURE

One other research that used the RentTheRunway dataset is *Decomposing Fit Semantics for Product Size Recommendation in Metric Spaces* done by Misra et al [1]. They are focusing on solving the imbalance labeling problem of the dataset and providing a new fit prediction model which helps to reduce the customer return rate.

Similar to our approach, Misra et al. also used latent factors on the RentTheRunWay dataset. However, they used a latent factor formulation to analyze the customers' reviews in the ModCloth and RentTheRunWay dataset. Sembium et al. used a latent factor model to recommend products of a certain size by minimizing the difference between the size of a customer and a product [2]. They also try to make the prediction based on latent factor model. The difference between their method and our method is that our model is based on TensorFlow, which may have some limitations compared to their self-made model. Moreover, a study excluded the customer's bias of rating by using the entropy of the customer's rating, and used a review-item matrix to predict rating [3]. In addition, Ochi et al extracted words from customer's reviews as features by analyzing the relationship between review comments and ratings to predict customer's ratings [4]. These researches have a similar direction as our research. We all found the importance of users' review when predicting the final rating.

5 CONCLUSION

Based on all the information stated above, we can conclude that linear regression with NLP on text_review is the most suitable model for predicting tasks on our dataset. Our NLP optimized linear regression model has a MSE of about 0.380 and an MAE and an R^2 of 0.468 and 0.259 respectively. Compared to other models we have tried which normally have a MSE of above 0.5, our NLP optimized linear regression model is much better than other alternatives, which shows the fact that an accurate prediction always requires an analysis of both the numerical and text features. From our experiment, we found that feature representations such as "height", "weight", "age", "size", "fitness" and "review_text" works well whereas features such as "review_date" and "rented_for" don't work well. This makes sense because intuitively whether a cloth is fit can directly affect a user's rating and the review_text is usually a user's explanation of his/her rating. In our model, the coefficient for each numerical feature is as following:

Table 1. Coefficients

feature	theta
intercept	4.4827
height	-0.0022
weight	-0.0003
age	-0.0020
size	-0.0009
fit	0.2929

We can see that the feature "fitness" contributes most to the prediction, which makes sense since fit always usually means higher rating and vice versa. Thetas for height, weight, age and size are negative, which means a higher height, weight, age or size usually means a lower rating. However, given the small magnitude of these thetas, we can conclude that these features have relatively small impact on rating prediction. Given we have incorporated NLP into our linear regression model, the top ten words that contributed the most to our prediction is as following:

Table 2. Coefficients of Most Influential Words

word	coefficient
incredible	0.2122
princess	0.1869
afraid	0.1766
deal	0.1690
stop	0.1686
weeks	0.1665
dream	0.1655
glove	0.1617
fantastic	0.1589
bother	0.1533

As we can see in the above table, the word "incredible" has the largest theta which means its occurrence in the review_text usually means higher rating. Given that our NLP optimized linear regression model has a much better MSE, we can say that this model is successful compared to other alternatives and its success can be attributed to the combination of both numerical and sentiment analysis.

REFERENCES

[1] Rishabh Misra, Mengting Wan, and Julian McAuley. 2018. Decomposing fit semantics for product size recommendation in metric spaces. In *Proceedings of the 12th ACM Conference on Recommender Systems*.
[2] Vivek Sembium, Rajeev Rastogi, Atul Saroop, and Srujana Merugu. 2017. Recommending Product Sizes to Customers. In *Proceedings of the eleventh ACM conference on recommender systems*.
[3] Masanao Ochi, Yutaka Matsuo, Makoto Okabe, and Rikio Onai. 2012. Rating Prediction by Correcting User Rating Bias. In *2012 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology*.
[4] Masanao Ochi, Makoto Okabe, and Rikio Onai. 2011. Rating prediction using feature words extracted from customer reviews. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*.