Predictive Analysis of User Ratings in RentTheRunWay Data

CSE 158R

Thu Mai
Trung Luu
Anh Vuong
Thanh Phan

Professor Julian McAuley
University of California, San Diego
December, 2023

Source Code: https://github.com/ahvuong/Predictive_Analysis_of_User_Ratings

CONTENTS

1 Dataset

1.1 Introduction

Fashion is essential to express distinctive personalities of each individual, hence, clothing fit is even significant to bring confidence and satisfaction when customers decide to purchase any clothing products. Understanding these aspects, our group wants to utilize one of the professor's datasets that apparently attractive to us, which is RentTheRunway from Clothing Fit Data [4], to explore the relationships among different factors provided and develop a model to predict customer ratings in order to enhance the quality of products and exploit the potential of technology towards fashion industry.

Diving into the dataset itself, the information includes various properties such as fit feedback, user ratings and reviews, user and item measurements, and other categories associated with the customer purchase. Through observations and investigations, we can potentially determine the correlation between the variety of items with specified customer information given, for instance, events in 'rented for' impact the category type, such as dresses are usually for formal affairs or wedding and distribute to the rating results. Additionally, body measurements, age also affect the sizing and fit. Those attributes altogether influence the overall final report of users which require our deeper analysis.

Given the expansion of the online fashion sector and the diverse size ranges found in various clothing items, there is value in exploring the automated delivery of precise and personalized fitting advice.

Table 1: Basic Statistics

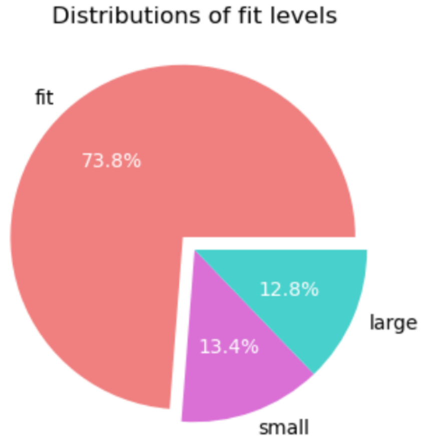
Statistics	RentTheRunWay
Number of Users	105,508
Number of Items	5,850
Number of Transactions	192,544

1.2 Data Cleaning/Preprocessing

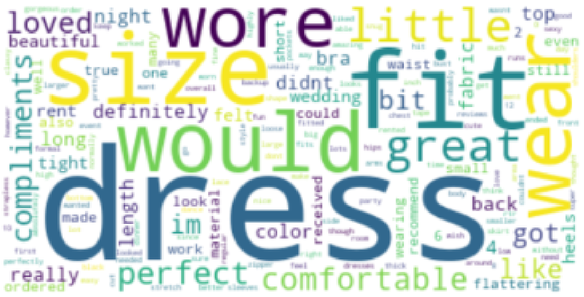
Initially, we aimed to enhance data cleanliness by dropping missing entries. Null values in key features, including 'rating', 'age', 'weight', 'height', 'body type', and 'rented for' were dropped as a crucial part of our data preprocessing and model development later. For 'weight' and 'height', we extracted their numerical values by trimming associated units. The 'rating' feature was transformed from a scale of 2, 4, 6, 8, 10 to a more user-friendly range of 1 to 5. Additionally, we preprocessed the 'review_text' by disregarding capitalization and removing punctuation. Our goal is to minimize uncertainties in incomplete reviews, improving the overall predictive quality of our models.

1.3 Exploratory Data Analysis (EDA)

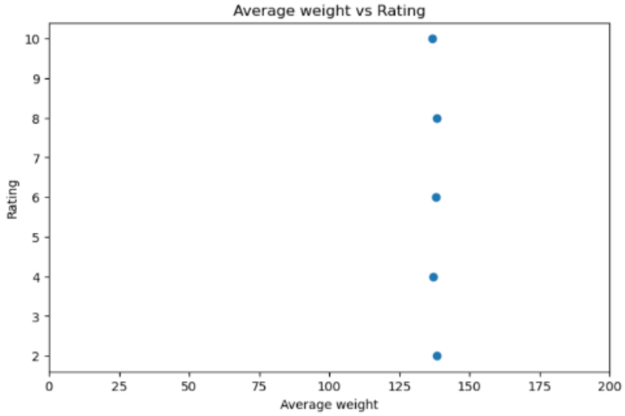
We first performed an analysis on the fit level of each clothing item. We saw that about 73.8% of people who left at least one review have items that fit their bodies. Note that this does not mean those people necessarily gave positive ratings.



We also did some analysis on the wording used in the user reviews. We tokenized the reviews into words and eliminated stop words and punctuations. We then sorted them in descending order, and extracted 1000 top most frequent ones. In the following word cloud, larger words are associated with higher frequencies.

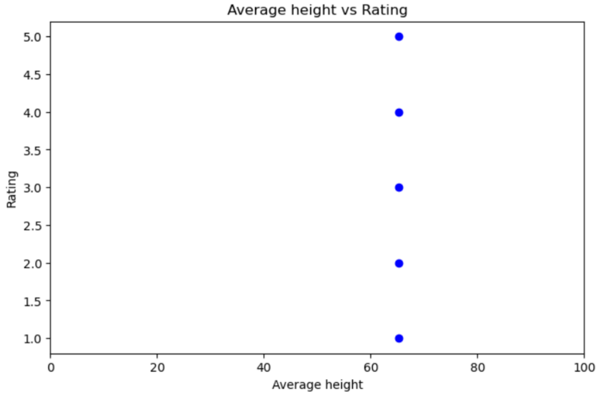


The third analysis we did on the data was figuring out the correlation between the weights and the ratings. The following graph uses average weights since they minimize the MSE which can be useful to predict the trend during training.

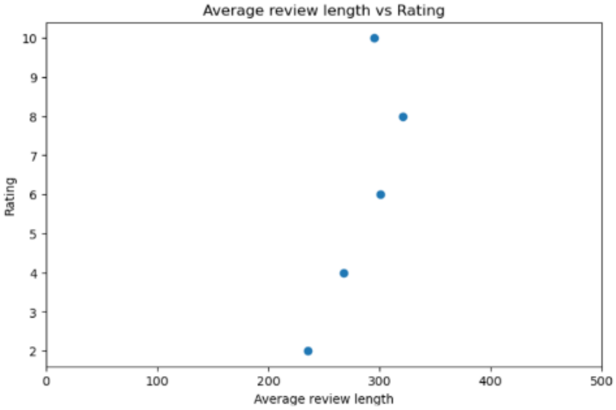


From the above graph, we can see that users tend to give ratings that are even and that weights do not seem to determine any rating patterns.

We did the same thing for the height attribute. The following scatter plot shows that the height, similar to weight, does not contribute much to rating prediction.



However, examining the review lengths gives us a trend in almost linear fashion.



2 Predictive Task

After careful consideration, we aimed to predict the ratings of clothing items using the dataset RentTheRunway, to understand how well the clothing fits the specified users given their measurements and other categories, as the features to build our predictive model. We found that it has the potential for predicting ratings because the dataset is a mix of categorical and numerical feedback, which essentially determines accurate ratings in clothing fit. After all, ratings do not depend solely on the fit level such as being tight or loose but also on the user satisfaction, which could be expressed through review text or review summaries.

To assess the validity of our predictions and confirm that they are significant, we took a strategic approach, from simple to more complex models for observing how the dataset responds differently to find the best ultimate-performing solution. Starting with the straightforward global average rating, then Linear Regression, directly applying the numeric values after pre-processing the potential data information. Another approach is the basic Latent Factor Model, which enhanced our performance by emphasizing user-item interactions. Then we utilized our experience on text mining techniques through the class to perform sentiment analysis combined with potential regression.

To improve our predictions, we implemented the Bag of Words model by incorporating item2vec and then, TF-IDF to capture important and unique review data, giving extra weights to them to indicate a group of significant words that potentially correlates to the final rating. In case of further optimization, we would first try the sentiment analysis integrated with Logistic Regression to figure out emotional tones in reviews. Later on, we went on to explore another combo between sentiment analysis and Linear Regression with the pursuit of more optimal results.

In our results of EDA, we have visualized several components inside the dataset to figure out which features should be applicable in our models. Since there is a significantly imbalanced distribution among different types of ‘fit’, we encounter uncertainty about whether this information influences the rating customer submitted. Then we process the ‘review_text’ and end up getting 1000-word indicators, after excluding irrelevant words, unnecessary characters, and many positive sentiment words that could potentially express users’ perspectives on their comments, hence, our approach of text-mining consists of this use of review texts. Then, our initial inclination was to gradually remove ‘weight’, ‘age’, and ‘size’ components when building the models because our scatter plots seemingly determined no significant correlation between the item rating and customers’ feedback. Nevertheless, our intuition is that the relationship between ratings and those attributes may not conclude anything, the components themselves may not make any effects, but what if the combination of them with others may obtain changes, so we still end up retaining those for further testing. Thus, to ensure the effective functionality of our model, we would try to incorporate all processed features in our prediction models to achieve the lowest errors. After conducting multiple combinations, we would first identify features for each model that gave the lowest errors after cleaning and pre-processing:

- Linear Regression: review lengths, size, weight, fit (small, large)
- Basic LTM: category, fit
- Text-mining: age, size, weight, height, rented for, body type, category

We decide to use MSE as our estimator to identify the loss of our models because we want to put more attention on significant errors, in other words, we don’t necessarily concentrate on perfect fit, but rather emphasize on substantial disappointments since the distribution of ratings is skewed towards high ratings.

3 Models

3.1 Global Average Rating Model (Baseline)

Similar to assignment 1, our first approach was straightforward, and that is always predicting the ratings using the global average rating of all clothing items. Before running the code, we split the data into two subsets where 80% of the data is for training and 20% of the data is for testing. This gave us an MSE of 0.5203.

3.2 Linear Regression Model

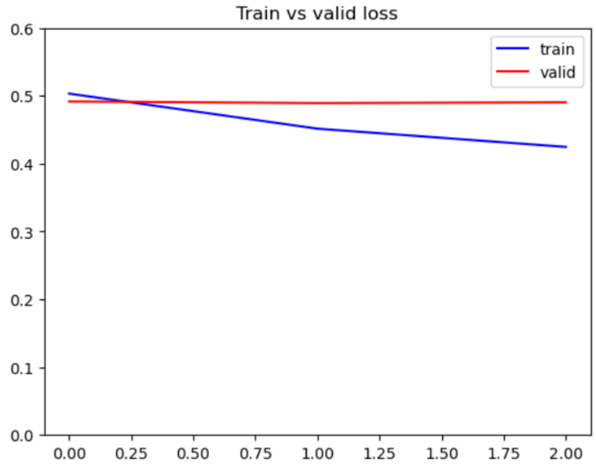
In our Linear Regression approach, we trained a simple predictor that estimates the rating from multiple features including length of review texts, size, weight and fit feedback. Inspired from our previous homework, we normalized the length of review texts by scaling the feature to be between 0 and 1 by dividing the maximum review length in the dataset to ensure the variety of review lengths contribute proportionally to the model. Additionally, we processed the fit feedback by using one-hot encoding to differentiate 'small', 'fit' and 'large' to understand the contribution of clothing fit towards rating feedback. The formula of our model features based on the basic simple formula to estimate a rating through linear relationship[3]:

$$\text{fit rating} \approx \theta_0 + \theta_1 \times [\text{review lengths in text}] +$$
$$\theta_2 \times [\text{size}] + \theta_3 \times [\text{weight}] +$$
$$\theta_4 \times [\text{fit == 'small'}] + \theta_5 \times [\text{fit == 'large'}]$$

By looking for the appropriate dataset splitting, we came to the result of dividing our processed dataset into 80% for the training set and 20% for the test set. The final MSE was 0.4726.

3.3 Basic Latent Factor Model

The next approach we considered was building a Latent Factor Model(LTM). There are numerous ways to implement LTM, but we chose to do it using TensorFlow which enables us to tune the hyperparameters. Before running the code, we split the data into three subsets where the first 80% is for training and validation and the next 20% is for testing. Within the training and validation set, we reserved the first 80% for training and 20% for validation. The model we built uses mini-batch gradient descent where MSE is the chosen loss function and Adam is the chosen optimizer. Using the learning rate of 0.005, a batch size of 256, and 3 as the number of epochs, we obtained an MSE of 0.4918 for the test set and an MSE of 0.4904 for the validation set. The following graph shows the training and validation loss for LTM where the y-axis represents the loss and the x-axis represents the epochs.



3.4 Text-mining Models

In our text mining approach, we integrated a standard bags-of-words model with item2vec item similarity scores to predict user/item pair ratings. After partitioning the data into 80% for training and 20% for testing, we assessed the model using MSE, resulting in approximately 0.5202. This is somewhat higher than the MSE from previous approaches and only marginally smaller than the baseline MSE. Exploring further, we considered another text mining model using TF-IDF scores for the 1000 most common unigrams and used log base 10 to transform data.

Table 2: MSE based on various C

C	MSE
0.32035752204149515	0.4809274983670803
0.17514703001325804	0.4776290006531679
0.07944591079715957	0.47802090137165254
0.9741681063775872	0.48020901371652513
0.5611151462276235	0.4788047028086218
0.9919763566446149	0.47854343566296537
0.047944061686267414	0.4792292619203135
0.22663521701730716	0.47926192031352055
0.3595267081827229	0.4800783801436969
0.785339655348197	0.4807968647942521

Leveraging logistic regression with a randomly tuned regularization constant (C), we achieved an improved MSE of approximately 0.4776 at C=0.785 on the test set. This performance slightly surpasses both the baseline and other models previously tested.

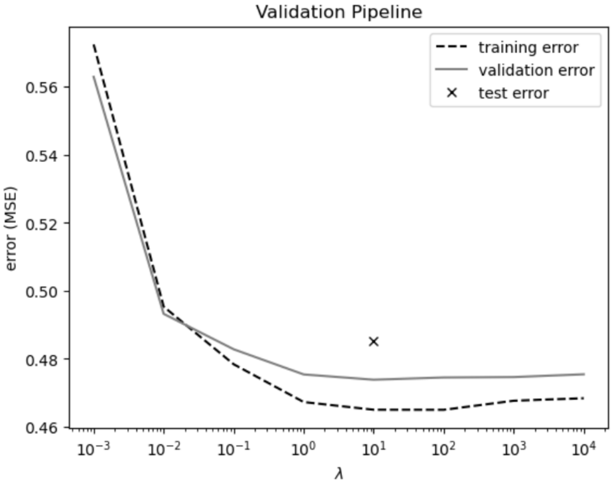
3.5 Optimization

The observation of our text mining models indicates that our hybrid approach, while innovative, may benefit from additional fine-tuning or feature engineering to achieve a more competitive performance. In pursuit of optimization, we broadened our exploration by incorporating more sophisticated models. In our first attempt, we introduced a sentiment analysis pipeline combined with a logistic regression model[2]. The data was split into three subsets, with the first 80% designated for training and validation, and the remaining 20% for testing. Features encompass a dictionary size of the 1000 most popular words, lower-case conversion, and punctuation removal.

Table 3: MSE Validation based on various λ

λ	MSE Validation
0.001	0.5628674069235794
0.01	0.49322338340953625
0.1	0.4828135205747877
1	0.4754245591116917
10	0.47387328543435664
100	0.47452645329849774
1000	0.4746489222730242
10000	0.4754653821032005

Running with different regularization constants (λ) passed to the logistic regression model, we achieved an MSE of 0.4739 on at C=10. A validation pipeline graph below, depicting training, validation, and test errors, was plotted to visualize the model's performance. While surpassing the baseline and the Latent Factor Model and being just a little higher than the text-mining model using TF-IDF scores, it marginally exceeds the others.



In our second attempt, we used train_test_split to allocate 80% for training and 20% for testing. Focusing on optimizing a linear regression model with feature engineering and sentiment analysis, we initially observed the 1000 most common unigrams in 'review_text.' This resulted in a significantly improved MSE of approximately 0.3767, outperforming other cases. Expanding the study, we added other features. Comparing MSEs, the most optimized performance was achieved by combining the bag-of-words feature vector with 'age,' 'size,' 'body type,' 'rented for,' and truncated 'category' (with more than 1000), reaching the lowest MSEs of about 0.3757. This case surpassed other models and outperformed the baseline.

All models efficiently handle datasets with over 150,000 entries, eliminating scalability concerns in our modeling process. Additionally, meticulous data-cleaning procedures have successfully mitigated the risk of overfitting. As highlighted in the data preprocessing section, entries without ratings or missing essential properties like size, body type, reason for renting, age, etc., were removed. Given the small number of such entries, their exclusion has minimal impact on the dataset and does not compromise the effectiveness of the models.

4 Related Literature

The dataset RentTheRunway is in the collection for research by the professor Julian McAuley. Through searching, we have discovered that Rent The Runway is an online service that enables customers to rent designer clothing or accessories for cost-effective purposes. In the paper Decomposing fit semantics for product size recommendation in metric spaces[4], the dataset RentTheRunway was utilized to understand and improve the clothing fit recommendation, specifically handling cold-start scenarios and uneven distributions across different kinds of fitting feedback. Methods include logistic regression with latent variables, logistic regression with latent factors, and metric learning approaches.

Another paper we found is A Deep Learning System for Predicting Size and Fit in Fashion E-Commerce[5] which demonstrates the size and fit prediction using a neural network architecture named SFnet using both ModCloth and RentTheRunway datasets. They aimed to build a probabilistic model that could maximize the results using customer and article interactions on training data. Understanding that customers may have different preferences, the neural network model leverages stochastic gradient descent (SGD) to minimize outcome errors. In SFnet, the state-of-art methods the paper considers is inspired by Siamese networks but do not share input pathways, specifically, it takes two sets of input of customers and articles separately, then using embeddings for these set identifiers (such as user, item ID) so model learns from the information. Next it concatenates the embeddings for both entities to make inferences about fit and size. Features were used including customer and user data and their interactions for later embedding purposes. In their experiments, their model outperforms previous methods such as MLP baseline, LV-LR, LF-LF, ect. Furthermore, in the case that attributes of articles provided, SFnets still wins benchmark methods such as Naive Bayes and boosted trees. Even in other scenarios such as cold-start or multi-user accounts, SFnet stands out in overall predictive efficiency. We found a few similarities such as the same dataset, and leverage the customer feedback and interactions for prediction tasks , however, we apply only basic concepts compared to them, SFnet is remarkable by using deep learning techniques for optimal personalized recommendation results.

Another similar paper that also predicts clothing fit based on two same datasets is Analyzing Customer Feedback for Product Fit Prediction[1] by Stephan Baier. The paper analyzed various predictive performances and ultimately decided to perform fit recommendation, whether "fit", "small" or "large" by NLP techniques. They have considered various approaches including linear classifiers on top of TF-IDF, word embeddings using GloVe (Global Vectors) and transfer learning models, in particular, ULMFit Fine-tuned and BERT Fine-Tuned. Their input is only the review text, and they split 80% data for training and validation, 20% for testing which is mostly similar to us, except that they used 5% of training as validation, which is numerically different from our models. We also demonstrate Bag-of-Words using TF-IDF and sentiment analysis to transfer raw texts into features for machine learning analysis. They have concluded that their ULMFit provides the best results among all techniques, in numerical ways, 0.8269 as accuracy for ModCloth dataset and 0.8420 for RentTheRunway. Compared to our findings, the best model is also based on text mining strategies but different models since ours incorporates Linear Regression with word frequencies.

5 Results and Conclusion

Our initial baseline model used the global average rating for review rating prediction in the RentTheRunWay dataset, achieving a satisfactory MSE of 0.5203. However, subsequent models demonstrated superior performance. We enhanced the baseline by employing simple Linear Regression, incorporating features derived from rating estimation mainly based on review length and feature vectors for fit. This resulted in a notable improvement, yielding an MSE of 0.4726, surpassing the baseline. In the process, we also explored the basic Latent Factor Model (LTM), which learns user-clothing item compatibility through matrix factorization. However, LTM exhibited inferior performance compared to the simpler Linear Regression Model, with an MSE of 0.4902. Notably, our experiments and early graphs indicated that review length correlates with the rating quality. In our final exploration, after experimenting with various text-mining models, combining features from the Linear Regression Model with word frequencies yielded the best result, achieving the lowest MSE at 0.3757.

From the analysis phase graphs and our model training results, we deduce that 'height' and 'weight' features have negligible impact on regression outcomes. In contrast, the combination of 'age,' 'size,' 'body type,' 'rented for,' and one-hot encoding of truncated 'category' (with more than 1000) feature vectors, along with word frequency in each review used in the Bag-of-Words Model, plays a crucial role in achieving the lowest MSE for our predictive task. This aligns with the intuition that these features directly influence users' explanations of their ratings and purchasing decisions. In conclusion, the success of combining sentiment analysis with the linear regression model stems from seamlessly integrating numerical and sentiment analysis features. This approach outperforms alternatives, resulting in a more comprehensive understanding of user preferences and superior rating estimations.

References

[1] Stephan Baier. 2019. *Analyzing Customer Feedback for Product Fit Prediction.* Retrieved December 4, 2022 from https://arxiv.org/abs/1908.10896

[2] Julian McAuley. 2022. *Personalized Machine Learning - Chapter 8.* Retrieved December 3, 2022 from https://cseweb.ucsd.edu/~jmcauley/pml/code/chap8.html

[3] Julian McAuley. in press. *Personalized Machine Learning.* Cambridge University Press.

[4] Rishabh Misra; Mengting Wan; Julian McAuley. RecSys, 2018. *Decomposing fit semantics for product size recommendation in metric spaces.* Retrieved December 3, 2022 from https://cseweb.ucsd.edu/~jmcauley/pdfs/recsys18e.pdf

[5] Abdul-Saboor Sheikh; Romain Guigoures; Evgenii Koriagin; Yuen King Ho; Reza Shirvany; Roland Vollgraf; and Urs Bergmann. 2019. A Deep Learning System for Predicting Size and Fit in Fashion E-Commerce. In *Thirteenth ACM Conference on Recommender Systems (RecSys '19).* Copenhagen, Denmark, New York, NY, USA, 9 pages. https://doi.org/10.1145/3298689.3347006 https://arxiv.org/abs/1907.09844