Assignment 2 – Cloth Rating Prediction

Zhixuan Cui Shi Qiu Yueh-Huan Ho

1.1 Data identification

The two datasets we use are from ModCloth and RentTheRunway, which contain measurements of clothing fit. Modcloth is a company Dreamt up in a dorm room in 2002 by a girl who loved vintage fashion, they still stay true to this aesthetic through our classic-meets-contemporary designs. While They've outgrown the dorm room, they're still a small team committed to providing customers with beautiful vintage-inspired fashion in inclusive sizes, made ethically by factories around the world.

Rent the Runway was co-founded in 2009 by Jenn Hyman and Jenny Fleiss. The company is a dream closet filled with an infinite selection of designer styles to rent, wear and return (or keep!). Every trend, every color, every print, everything you've ever wanted to wear — for a fraction of the cost.

	Modcloth	Renttherunway
Number of users:	47,958	105,508
Number of items:	1,378	5,850
Number of transactions:	82,790	192,544

Table 1: basic stats

1.2 Exploratory Analysis

First, we investigated the dataset of RentTheRunway. We found the top 10 popular items that have been rent for thousands of times.

item_id	fit	user_id	bust size	weight	rating	rented for	review_text	body type	review_summary	category	height	size	age	review_date
126335	2241	2241	2006	1662	2231	2241	2241	2166	2241	2241	2237	2241	2234	2241
174086	1724	1724	1586	1358	1724	1724	1724	1673	1724	1724	1719	1724	1717	1724
123793	1714	1714	1553	1485	1714	1714	1714	1612	1714	1714	1709	1714	1705	1714
132738	1582	1582	1412	1192	1577	1582	1582	1534	1582	1582	1577	1582	1574	1582
145906	1478	1478	1348	1189	1472	1478	1478	1455	1478	1478	1476	1478	1468	1478
127865	1393	1393	1279	1220	1393	1393	1393	1311	1393	1393	1386	1393	1382	1393
136110	1197	1197	1053	936	1197	1197	1197	1144	1197	1197	1193	1197	1195	1197
137585	1100	1100	1018	946	1100	1100	1100	1061	1100	1100	1093	1100	1097	1100
131533	1091	1091	970	864	1091	1091	1091	1034	1091	1091	1087	1091	1088	1091
172027	984	984	930	873	984	984	984	963	984	984	981	984	982	984

Table 2: top 10 popular items

Then we tried to find the mean of ratings. But we found out that there are 83 null values in the rating columns. So, we dropped those null values to find the mean. And the average mean is 9.09. It's relatively high since most people rate it 10.

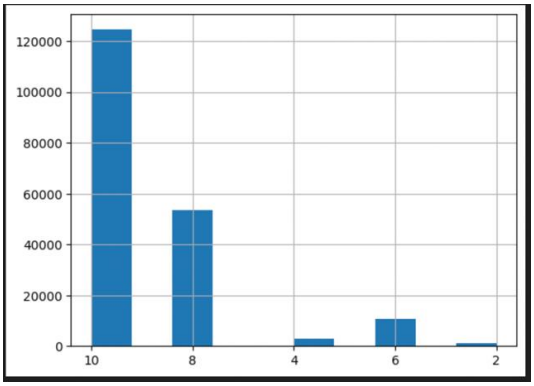

Table 3: Rating distribution

Conducting the same process, we found that there are 960 null values in the age column and the average age is 33.87 years old.

Then we investigated the dataset from ModCloth, we found that the average size of clothes people bought is 12.66.

	size	quality
count	82790.000000	82722.000000
mean	12.661602	3.949058
std	8.271952	0.992783
min	0.000000	1.000000
25%	8.000000	3.000000
50%	12.000000	4.000000
75%	15.000000	5.000000
max	38.000000	5.000000

Table 4: stats of clothes size

2.1 Predictive Task

The task in this project is to predict the user's rating for a given item_id and the user's features. Using this model, we recommend to each customer a set of items by ranking the predicted ratings.

In the dataset, there are several interesting features, such as age, body size, bust size, height, and review text that is worth discovering in the model training process. However, some features are not easy to obtain before the user posts the rating. In our project, we want to recommend to every customer accurate items set even though some are first-time customers.

Therefore, a more reasonable way for building our predictive model is to discard those features such as review text and review summary that are not easily observed in the first place. Nevertheless, the way that we don't use text data doesn't mean the text data is useless. Text mining is still a worth-discovering method since it provides valuable information for each item in the training set.

To validate our model's performance, we select MSE as our loss function and try to fit the model by minimizing the MSE of the model. Before creating the baseline model, we first implement a model that always predicts the average and the MSE for this model is around 2.08. The MSE is surprisingly low. By further investigating the rating distribution in the dataset, we found that

the average rating is relatively high and the variation of ratings is not huge; most users gave 10 for the rating. This subtle variation in the actual ratings might be the reason why the primary model is performing so well.

In our baseline model, we first implemented the latent factor model and assigned a unique beta for each user_id and item_id that appear in the training set. We also apply a regularizer containing two lambdas.

Baseline equation:

$$Min\ \Sigma_{u,i}\left(\alpha + \beta_u + \beta_i + \gamma_u \cdot \gamma_i - R_{u,i}\right)^2 + \lambda_1[\Sigma_u\beta_u^2 + \Sigma_i\beta_i^2] + \lambda_2[\Sigma_i||\gamma_i||^2 + \Sigma_u||\gamma_u||^2]$$

In this project, the training and validation sets are split in a 7:3 ratio. The training data contains 150000 observations and the validation data contains 45000 observations. By observing the training and validation datasets, we also found that there is a lot of user_id not seen in the training set. The percentage of unseen user_id in the

validation set is about 42% and the one of item_id is less than 1%.

However, if there is an unseen user_id or item_id in the validation set, the beta and the latent factor matrix will be assigned zero. Using this method in this highly unbalanced data might be inappropriate. Therefore, we assume that those unseen proportions user_id might sever as the main reason for the high MSE in our baseline model.

To get more information on the user and item interaction, we believe that the difference between a particular user's age and the item's average might be a strong factor that affects the user's preference and therefore the rating. We first calculate the average buying age for each item and the difference in age for each observation.

In this dataset, some users did not provide their ages in their reviews. To solve this problem, the global average age in training data was used to replace those missing age values.

3.1 Select/design model

To optimize the baseline model, we first apply Adam as the model optimizer and set the learning rate at 0.05. In the training step, we implement the batch-based model and then run 300 iterations of gradient descent to fit our model. After several experiments, we found that setting too many iterations might cause the model to overfit. For the regularizer, we utilize two lambdas.

We found that it is not easy to make the regularizer work by using only one lambda. Using two lambdas, we have more control over the betas and the latent factor matrix. We found that setting $\lambda 1$ too small very easily causes model overfitting. The lambdas used for models are $\lambda 1$: 0.0001 and $\lambda 2$:0.1. The final MSE of our baseline model is around 1.98.

We also try the heuristic mentioned previously to invent two models. In the following models, we mainly focus on using the age_diff feature that we just created to refine our baseline models.
In the first model, we implement a

Item_Id	Average age
233953	38.26
836119	31.00
1016759	34.10
1326545	28.73
1566348	32.68

User Id	Item Id	Rating	Age Dif
145486	296781	6	-10.64
600599	1285250	10	5.57
736674	1501483	8	-6.82
855009	2204233	10	-6.05
836250	1076484	4	0.97

Table 5: features used in the model training process

regression model after we calculated the prediction results using the baseline model. The main reason for using a regression model on the age_dif feature is because it's easy to investigate the influence of age_dif using a basic statistical model.
After running a regression model in the training set, we got a theta and a coefficient of the feature, and then further apply the theta and coefficient for the validation set.

Model (c) Latent Factor + Regression

$$R = \theta + \alpha_{age\_diff} * R_{baseline}$$

The training MSE before applying the regression method is 1.65. Compared with the MSE in the baseline, the model in the training set improved by 4%. However, when we calculate the MSE on validation data, the MSE increases to 2.06. This model performs extremely badly on the validation set might be because the age_diff feature is not as significant as we thought previously. The other reason might be the fact that the regression model is prone to overfitting the model. The method in which we implement the regression afterward might also be problematic. However, by implementing this model, we know that the age_diff is influencing the expected rating negatively. This finding indicates that users are more likely to buy items that have a similar average age to their ages.

We create our second model by refining model (c). To solve the problem of training coefficient after we finished the latent factor model. We add one trainable variable δi in the baseline model. The δi is the coefficient of age_dif for each item and we train the δi with the variables in the baseline model altogether. To solve the overfitting problem on δi, we also apply λ3 for the regularizer. In the training process, we try to minimize the following equation:

Model (c) Latent Factor + Age_diff

$$\begin{aligned} Min\ \Sigma_{u,i}\big(\alpha\ &+ \beta_u + \beta_i + \gamma_u \cdot \gamma_i + \delta_i \\ &* age_{dif} - R_{u,i}\big)^2 \\ &+ \lambda_1\left[\Sigma_u \beta_u^2 + \Sigma_i \beta_i^2\right] \\ &+ \lambda_2\left[\Sigma_i \left|\left|\gamma_i\right|\right|^2 + \Sigma_u \left|\left|\gamma_u\right|\right|^2\right] \\ &+ \lambda_3[\Sigma_u \delta_i^2] \end{aligned}$$

This model has an MSE of around 1.875, which is 5% better than the baseline model. In this model, we use more users' information besides user_id and item_id. This implementation solves the problem of a high percentage of unseen user_id in the validation set. In the future model design and experiment, it's worth trying to apply more custom features to the model training process.

4.1 Related literature

We did not introduce text mining into our model, but since text mining is usually very useful for rating prediction, I will share some literature about how text mining works in the prediction model.

In the article *Yelp Dataset Challenge: Review Rating Prediction, the author Nabiha Asghar* focuses on Review Rating Prediction problem for restaurant reviews on Yelp. He treats it as a 5-class classification problem and examines various feature extraction and supervised learning methods to construct sixteen prediction systems. Experimentation and performance evaluation through k-fold cross validation yields one system, Logistic Regression on the set of top 10,000 features obtained from Unigrams & Bigrams, that exhibits better predictive powers than the others. (Nabiha, 2016)

In the article *Contextual Recommendation based on Text Mining,* the author introduced a probabilistic latent relation model for integrating the current context and the user's long-term preferences. This model takes advantage of traditional collaborative filtering approaches. It also captures the interaction between contextual information and item characteristics. The experimental results demonstrate that context is an important factor that affects user choices. If properly used, contextual information helps ranking based recommendation systems, probably because context influences users' purchasing decisions. Besides, more accurate contextual information leads to better recommendation models. (Yize *et al*. 2010)

In the article *Latent aspect rating analysis on review text data: a rating regression approach,* the author define and study a new opinionated text data analysis problem called Latent Aspect Rating Analysis (LARA), which aims at analyzing opinions expressed about an entity in an online review at the level of topical aspects to discover each individual reviewer's latent opinion on each aspect as well as the relative emphasis on different aspects when forming the overall judgment of the entity.

They also propose a novel probabilistic rating regression model to solve this new text mining problem in a general way. Empirical experiments on a hotel review data set show that the proposed latent rating regression model can effectively solve the problem of LARA, and that the detailed analysis of opinions at the level of topical aspects enabled by the proposed model can support a wide range of application tasks, such as aspect opinion summarization, entity ranking based on aspect ratings, and analysis of reviewers rating behavior. (Hongning *et al*. 2010)

5.1 Results

Method	(a) Always_mu	(b) Baseline	(c) LF + RG	(d) LF + Age_diff	Imporvement (a) vs. (b)	Imporvement (b) vs. (c)	Imporvement (a) vs. (d)
Training MSE	2.077	1.719	1.646	1.618	17%	4%	6%
Validation MSE	2.086	1.986	2.062	1.875	5%	-4%	5%

Table 6: model training results

Our results are summarized in table 6. Model (d) has the lowest MSE performance compared with others. We conclude that the main reason is that model (d) takes the user feature and the interaction with the corresponding item into consideration. This kind of feature allows the model to get more information to predict the unseen user/item pair. The reason why mode (c) performance is not ideal is probably that we overfitted the model in the training set.

Moreover, the regression model is probably not a suitable model for this kind of prediction because the model is not flexible enough to predict the uncertainty in this kind of dataset. Nevertheless, model (c) still provides useful insight that allows us to observe the influence of features easily. In the future model design, we suggest that more features can be used and combined with our baseline model. Text mining is also a feasible method to discover more information in user/item interaction.

Moreover, our model can be further improved if we implement an extra validation set or maybe hyperparameter tuning. We do not suggest using too complicated a model since it's easy to cause the model to overfit. In the future, we suggest

focusing more on feature engineering to create more informative features or changing the way of splitting train/valid data.

REFERENCES

[1] Asghar, Nabiha. "Yelp dataset challenge: Review rating prediction." arXiv preprint arXiv:1605.05362 (2016).

[2] Yize Li, Jiazhong Nie, Yi Zhang, Contextual Recommendation based on Text Mining

[3]Wang, Hongning, Yue Lu, and Chengxiang Zhai. "Latent aspect rating analysis on review text data: a rating regression approach." Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining. 2010.

[4] Translation-based factorization machines for sequential recommendation Rajiv Pasricha, Julian McAuley RecSys, 2018

[5] Translation-based recommendation Ruining He, Wang-Cheng Kang, Julian McAuley RecSys, 2017