CSE 158 - Assignment 2

Kenneth Nguyen Egor Pustovalov Mandy Lee Yongqing Li

***Abstract*—This paper focuses on the reasoning behind choosing different learning models for predicting ratings on items in a data set. The findings from the best model are analyzed and compared to existing studies on the same dataset or similar predictive undertakings**

I. Task 1

RentTheRunway is an online designer clothes retailer that focuses on renting out designer clothes to consumers rather than selling them to allow the customer flexibility and variety on their clothing options without forcing them to commit to an item of clothing. They do offer purchase options should a consumer really enjoy a particular item. Each item on the website contains a review section where a consumer can describe their experience with that particular item [1]. The "RentTheRunway" dataset used in this assignment contains 192,462 reviews on articles of clothing purchased through the website from 2010 - 2018. Each review contains the user's review text, body measurements and age, rating of the item, unique user id, the item id, summary of the review, and the date on which the review was made. The ratings among all of the reviews take an even value between 2 and 10 (i.e. 2, 4, 6, 8, 10) where 2 is when the user was the most dissatisfied with the clothing and 10 is when the user was the most satisfied with the clothing. This is a point of interest because standard rating scales typically take every value between 1-10 or 1-5 (upon further research it was stated that the reviews were originally given on a scale of 1-5 and scaled up to the value range of 1-10, which explains the reviews being only even numbers source here). As we can see from Figure 2, the average rating of clothing in the dataset is around 9.09 with a standard deviation of 1.43 which shows that the users in the dataset were typically quite satisfied with their purchases with an average of a 1 point deviation between the different ratings. Looking at the distribution of the user ratings from Figure 1, the rating data seems very heavily left skewed (which the mean value does support this observation) which means that there are not many dissatisfied reviews that the model will be trained on. This could possibly cause an issue with how the model could be biased towards higher review predictions as most of the training data will consist of the higher rated items and reviews.

When looking at the amount of null values in the dataset as shown below in Figure 3, the data shows that the categories with the most missing values are age, body type, bust size, and weight. Based on these categories being the most missing, it seems likely that these data values are missing completely at random due to some of the women most likely not being very comfortable with sharing their personal information in their reviews. Keeping this information in mind, the model will most likely not include these variables as features due to the limited amount of usable data available.

Fig. 1. Distribution of User Ratings

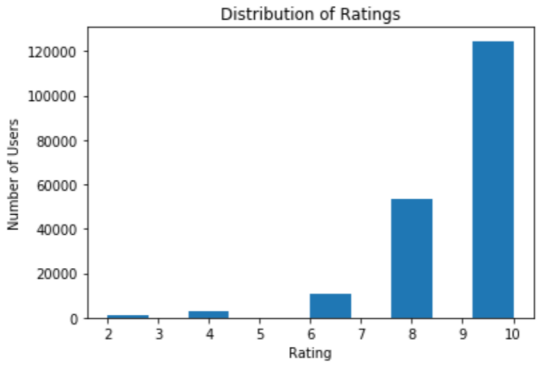

Fig. 2. Summary Statistics of Rating and Size Variables

	rating	size
count	192462.000000	192462.000000
mean	9.092371	12.245254
std	1.430044	8.495463
min	2.000000	0.000000
25%	8.000000	8.000000
50%	10.000000	12.000000
75%	10.000000	16.000000
max	10.000000	58.000000

Fig. 3. Number of Missing Values by Variable

weight	29957
bust size	18397
body type	14637
age	959
height	677
rented for	10
fit	0
user_id	0
item_id	0
rating	0
review_text	0
review_summary	0
category	0
size	0
review_date	0

The variable "rented for" (which indicates the occasion for which the article of clothing was purchased for) has 10 missing values (which is less than a quarter of a percent of the data set),

but can serve as an interesting feature because the context in which the clothing was purchased could have some influence on how high or low the users rate the item. For example, after looking at the count of different examples of reasons a dress was bought for in Figure 4, a more joyous occasion such as a wedding might yield higher ratings due to the user's elevated mood while a less joyous occasion such as work might yield lower ratings due to the user's less enthusiastic mood about going to work). Using this reasoning, the heavily left skewed distribution of user ratings in Figure 1 make sense as weddings (typically a joyous occasion) are the most prevalent reason for people renting or buying the clothes. This large number of weddings could have possibly skewed the results more positively as the festive mood could have made the users more generous in their ratings. To further quantify this observation, the average ratings could be stratified by their "rented for" occasion. Looking at the average user rating for each occasion group in Figure 5, the data seems to follow this observation as the more joyous events (wedding or vacation) have a higher average rating than an occasion that is not as joyous (work or everyday). This might prove to be a useful feature to integrate into the final model.

Fig. 4. Occasions for why the Clothing was Bought/Rented and their counts

wedding	57768
formal affair	40365
party	35613
everyday	16822
other	15381
work	15042
date	7387
vacation	4073
party: cocktail	1

Fig. 5. Average User Rating by Occasion Rented/Bought For

	Occasion	Average Rating
1	formal affair	9.210901
0	wedding	9.190278
4	other	9.114622
2	party	9.056468
7	vacation	8.981095
6	date	8.955733
5	work	8.851084
3	everyday	8.830460
8	party: cocktail	8.000000

II. Task 2

For this dataset, the goal will be to use a bag of words representation on the review text of each user as well as a one hot encoded "rented for" feature in a linear regression model to predict the rating (which is a numeric variable) a user gave the article of clothing. The ratings will be rounded to the nearest even number to match the scaled up ratings in the dataset. From the EDA above, it seems like the "rented for" variable could have some indication on how a user is likely to rank their clothing so it seems like a fairly useful feature to include in the model. To evaluate how well this model performs on this predictive task, the Mean Squared Error (MSE) will be used to measure its performance. A baseline model that can be used to compare this model's performance against will be a model that predicts the rating based on the bag of words representation of the review summary only. This would give a brief overview of what the user thought of the item and will give a ballpark estimate of what the rating could be. To obtain the bag of words feature for the model, all of the words from the review text in the training data was made lowercase, stripped of the punctuation, and filtered out for any stopwords (which are very common words in a language that do not contain any significant meaning such as "the", "and", "a", etc.). The frequency in which each unique word appears in a review is kept track of and then sorted by greatest frequency. The 1000 most frequently used words in a review are being used for this model. From this list of 1000 words, each review in the training set is broken up into its individual words and a feature vector keeps track of how many times a word from the list of 1000 most popular words appears in the review. These feature vectors are then stored in a sparse matrix for memory efficiency. To create the feature which will encode the "rented for" variable, a one hot encoded vector will be used to represent which of the 8 reasons the clothing was used for. This model's predictions are valid in the scope of the RentTheRunway website as all of the data for the model was taken from the website, so it would not make sense to extrapolate the data beyond the scope of the website. Using the general understanding that positive words and negative words should correlate to a higher and lower rating prediction, respectively. Additionally as mentioned above, higher ratings should also correlate with the happier occasions that the clothing is rented for. The predictions should reflect this common knowledge and will be validated according to this schema. However, because the dataset is composed of mostly very positive reviews as seen in Figure 1, it will be difficult to claim that the model will be an unbiased model as it will not have as many negative reviews to train on as it did positive reviews.

III. Task 3

- The model that is being used to predict clothing rating is a linear regression model using a lambda value of 120 as a regularizer. The best lambda value was found by testing out many different lambda values on the validation set and finding the MSE of each iteration. After testing the different lambda values, the results concluded that a lambda of 120 yields the smallest MSE on the validation set. The features that this model will use are a bag

of word models and a one hot encoded representation of the "rented for" variable. To optimize the model for efficiency, sparse matrices were used where possible to save space in memory and the 1000 most frequently appearing words were used to build the bag of words model rather than using all of the words to cut down on memory and computation time. One unsuccessful attempt was to incorporate TF IDF vectors into our bag of word vectors as it gives us worse accuracy compared to using a bag of words.

- Along the way we have also used an extended latent factor model, fastFM, to explore the interactions between combinations of features. We first populated the feature vectors with a one hot encoding of the user and review ID. We used sparse matrices to account for the large feature dimensions which lowers memory and computational time. We defined our model as fastFM.als.FMregression with n_iter = 1000, $init\_stdev$ = 0.1, $rank$ = 5, $l2\_reg\_w$ = 0.1, and $l2\_reg\_V$ = 0.5, (parameters are taken from lecture notes and homework assignments). The resulting test MSE was 2.436 as shown in Fig. 6, this is not as accurate as our final bag of words and "rented for" features model. As we mentioned earlier, the "rented for" variable could be an interesting feature to include into our model. Thus we built another fastFM model that incorporated the one hot encoding of the "rented for" variable alongside the user ID and review ID. The parameters did not change from the previous model and the resulting test MSE was 2.74. For both of the fastFM models we also attempted to manually round the predictions to the nearest even number, as we observed only even numbered ratings are present in the dataset. We can see in Fig. 6 that the MSE for both models did not improve as compared to the exact predictions. Currently the best model only includes the user ID and review ID, so we will keep the model simple and experiment with different parameters and try to improve the resulting MSE's. Among the parameters, we saw that changes in $init\_stdev$ and $rank$ positively contributed to lowering the MSE. We first increased $rank$ from 5 to 10 and $init\_stdev$ from 0.1 to 2, this improved the model by 0.2. Next, we decrease the $init\_stdev$ from 2 to 0.5, and that has further decreased the MSE by 0.04 in the validation set. We have provided the results in Fig. 7. Ultimately, in terms of a fastFM model we were able to achieve the best results with parameters as follows: n_iter = 1000, $l2\_reg\_w$ = 0.1, $l2\_reg\_V$ = 0.5, $init\_stdev$ = 0.5, $rank$ = 10. Still, using fastFM that explores the interaction between the user and review ID does not do as good of a job as using linear regression. This is caused by the inclusion of multiple other features in the linear regression model, primarily the usage of review texts. We have learned in class that text data contains crucial information that can dramatically impact the model predictions. So one reason that contributed to the unsuccessful attempt of using fastFM could be the simplicity of the model.

Fig. 6. FastFM With Different Features Combinations

	User Item	User Item Rented	User Item Round	User Item Rented Round
Val MSE	2.481330	2.794968	2.918632	3.239115
Test MSE	2.435629	2.736420	2.883151	3.180548

Fig. 7. FastfM With Different Parameter Combinations

	stdev = 0.1, rank = 5	stdev = 2, rank = 10	stde = 0.5, rank = 10
Val MSE	2.481330	2.261816	2.229162
Test MSE	2.435629	2.277118	2.265582

IV. Task 4

This data came from the website of RentTheRunway, which is a company that allows users to rent or buy designer clothing. This public dataset is fairly popular among data science projects regarding recommendations and predictions.

In one study by data scientist Brittany Fowle, this dataset was used to predict item ratings for various dresses[2]. The dataset includes 68 unique categories for clothing, and they decided that 10 of these categories, including sheaths, shirt-dresses, ballgown, etc, should be categorized as dresses. Their baseline model was a Pearson correlation from the KNNBasic module in the surprise library and it was used to find the most similar user and take their rating for an item. Using only dresses with item counts less than 20, this gave them an RMSE of 1.48. From here, they improved to an RMSE of 1.30 by using singular value decomposition and hyperparameters optimized by GridSearchCV, also from surprise. This translates to an MSE of 1.69, which is 0.159 higher than our MSE from our best and final model. However, it is important to note that they used a subset of the data by modeling dresses with user and item counts greater than 15; in total this gave them 3,800 ratings to work with. For a model utilizing only 0.0197% of the ratings, this is an impressive MSE value.

In another study by Bingkun Wang et al. the goal was to compute rating prediction on items (music, movies, and books from another dataset in this case) just like we did, but using state-of-the-art methods[3]. Their specific goal was to show that methods involving review text content (RTC) and those based on reviewer-item rating matrices (RIRM) were much more effective when combined together. In their study they explored three models that relied on this combined information where the function $f(RIRM, RTC) = (1-)f_1(RIRM) + f_2(RTC)$ should predict a rating.

The first model had linear regression for RTC and k-nearest neighbors for RIRM which yielded an RMSE of 1.0491. The second model had linear regression for RTC and matrix factorization for RIRM which yielded an RMSE of 1.0042. The final model involved linear regression for RTC and k-nearest neighbors with matrix factorization for RIRM which yielded an RMSE of 0.9959. These all involved tuning lambda

and other variables to prevent overfitting and other problems, but ultimately all methods had an RMSE lower than the lowest individual RIRM or RTC method which was 1.065 for matrix factorization. Interestingly, our results differ from theirs in the fact that there was no strong correlation between users and items; this idea is supported by Brittany Fowler from the study above who used the same dataset as we did. Perhaps the items they examined had a higher cultural importance and thus greater sense of self compared to those from our dataset.

V. Task 5

The fastFM models (both Alternative 3 and Alternative 4) did significantly worse than the other ridge models in terms of MSE as they did not take the review text into consideration, which is an important part of deciphering how a user would review an article of clothing. For our baseline model, we implemented a bag of words on "review summary" along with one hot encoding on "rented for" (Baseline in Figure 8). We then apply that feature vector onto Ridge regression with a lambda value of 120 since a lambda of 120 gives us the best result with our validation data set. As for our final model, we used a bag of words on "review_text" along with one hot encoding for "rented for". The final model resulted in a MSE of 1.531 while the baseline model resulted in a MSE of 1.542. This means that "review_text" provides more information about a user's rating compared to "review summary". This is reasonable since review text has much more context about a user's opinion on an item compared to brief description in the "review summary". Therefore, having "review_text" as a feature is beneficial to the model. Another alternative model is including TF IDF on the review text as well as a bag of words representation of the review text (Alternative 1 in Figure 8). It results in a MSE of 2.531, which is a significant difference compared to an MSE of 1.542. This shows that adding on TF IDF Vectorization is not helpful to the model. This could be because the TF-IDF shows the importance of a word in a document relative to the rest of the other documents. However, since the reviews probably have very similar words to describe how much they like the clothing (words such as "beautiful" or "amazing"), the TF-IDF would not be as effective in helping predict their rating. The last alternative model we tried is to use the bag of words on "review_text" without one hot encoding on "rented for" in order to determine if the "rented for" feature is beneficial to the model (Alternative 2 in Figure 8). The MSE we get from that model is 1.543, meaning that having "rented for" variable as a feature is still a slight improvement from this alternative model. Knowing what event the user rented for provides in depth information of their possible mood. For example, events such as wedding or party might result in a higher rating due to a joyful mood. Overall, our final model that uses ridge regression, bag of words on "review_text", and one hot encoding for "rented for" yields the best result of 1.531 MSE. However, this result should be taken with a grain of salt as the dataset itself was not a very balanced mix of positive and negative reviews. Perhaps it might have been better to predict if a rating meet a certain threshold rather than the specific rating as that task might be better suited for this positively skewed data.

Fig. 8. Breakdown of Model Features Used and their Performances

	BoW on "review_text"	BoW on "review summary"	TFIDF	OHE on "rented for"	Review ID	User ID	MSE
Baseline		✓		✓			1.542
Alternative 1	✓		✓				2.531
Alternative 2	✓						1.543
Alternative 3 (Using fastFM)					✓	✓	2.4356
Alternative 4 (Using fastFM)				✓	✓	✓	2.2654
Final	✓			✓			1.531

References

[1] Project, UCSD CSE Research. "Recommender Systems and Personalization Datasets." Recommender Systems Datasets, Julian McAuley, https://cseweb.ucsd.edu/ jmcauley/datasets.html.

[2] Fowle, Brittany. "Creating a Recommendation Engine Using Rent the Runway Data." Medium, Medium, 20 May 2020, https://medium.com/@befowle/creating-a-recommendation-engine-using-rent-the-runway-data-c4c7867ad9c.

[3] Wang, Bingkun et al. "Combining Review Text Content and Reviewer-Item Rating Matrix to Predict Review Rating." Computational intelligence and neuroscience vol. 2016 (2016): 5968705. doi:10.1155/2016/5968705