Vectorize Review Semantics in Rental Clothing Fit Prediction

Sutianyi Wen
A13992949
s5wen@ucsd.edu

Bo Hu
A13805750
boh016@ucsd.edu

Siqi Huang
A14143758
sih041@ucsd.edu

Yifei Ning
A14508232
y3ning@ucsd.edu

1.EXPLORATIVE ANALYSIS

The datasets used are measurements of clothing fit from *RentTheRunway*, an online dress rent company that provide costumes rental to people in social occasions. The dataset has 192544 rows and 15 columns, which have attributes like ratings, reviews, fit, user measurement, item measurement, category information, etc.

Examining the data closely, we first convert the attribute Height in centimeters and the attributes Weight, Rating, Size, Age in numbers and check how many entries are nulls. It turns out there are 29982 entries are null in the attribute Weight, 18411 nulls in Bust Size, and 14637 nulls in Body Type. People tend to skip those private questions more likely than other non-private ones. Since blank entries are not useful for the prediction tasks, we drop rows that contain any of the null values. After dropping all the nulls, there are 146381 rows in the dataset.

Among all the rental data, around 74% of users think their clothes are fit to them and around 13% of users think their clothes are small and around 13% of users think their clothes are large. People who think their clothes are fit give ratings with an average of 9.30 which is higher than people who rated “small” or “large”; and those who rated “small” give an average rating of 8.42, with those who rated “large” giving an average rating of 8.52 (Figure 1).

```
fit
1    9.301774
2    8.422893
3    8.517187
Name: rating, dtype: float64
```

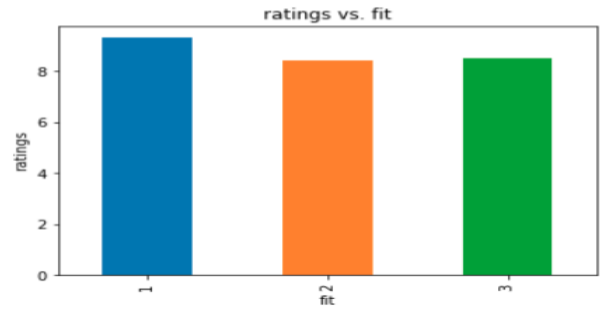

Figure 1 average ratings among different fit group

Grouping the data by the fit categories enables us to further study the inner distribution of ratings within each group. Given Figure 2, we are able to tell how the distribution differs group by group. Within each group, people still follow the general trends that most of them give good (8 or 10) ratings while only a few rates low (2 or 4), although People who think their clothes are fit tend to give high ratings (8 or 10) scores than the other two groups. Interestingly, the differences among each group’s rating distributions motivate us to use one hot encoder. (Figure 2).

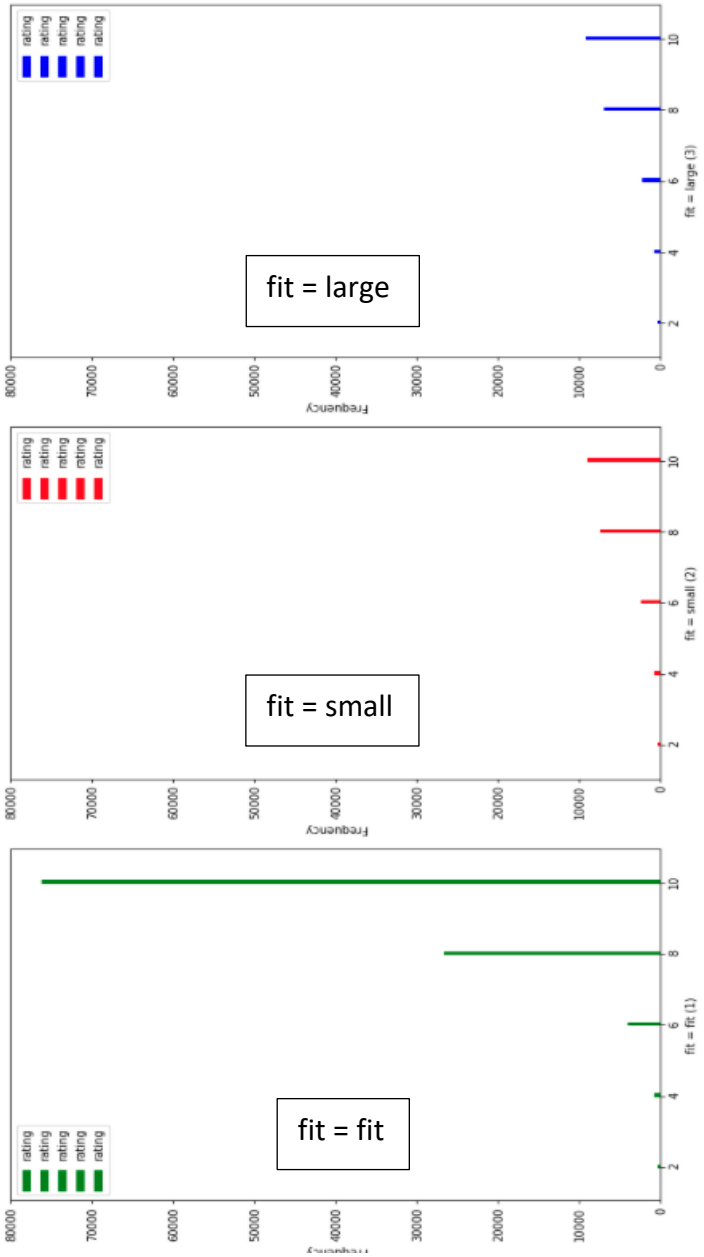

Figure 2 inner rating distribution within each fit group

Additionaly, the data are highly biased in terms of good ratings portions: about 98% customers give at least mediocre ratings (at least 6). Only around 2% users are not satisfied with the services (and thus give 4 or 2). This finding gives us motivations to randomly shuffle and split the data to obtain a fair and representative in our latter prediction tasks. Since people who are only satisfied with the rental services only occupy 2% of the total population, we need to think of a more discriminative metrics meature such as Balanced Accuacy Rate and we wil assign balanced weights to the model parameters etc (Figure 3).

```
10.0    0.644148
8.0     0.278458
6.0     0.056981
4.0     0.015063
2.0     0.005349
Name: rating, dtype: float64
```

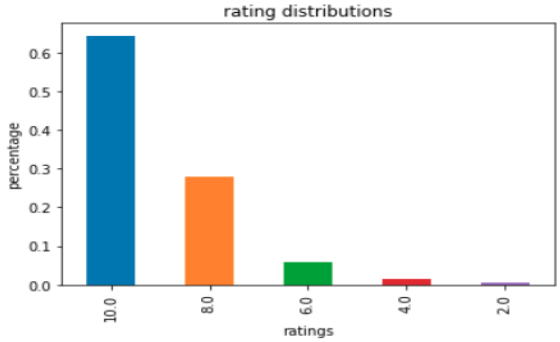

Figure 3 the distribution of ratings in the data set. Most the people rate scores above the "average" score 6.

Another interesting finding is that cloth rental is popular among younger people. Our data population has an average age of 34 with a standard deviation of 8 years old. The density distribution of ages is right skewed implies that the data contain more people of younger ages (Figure 4).

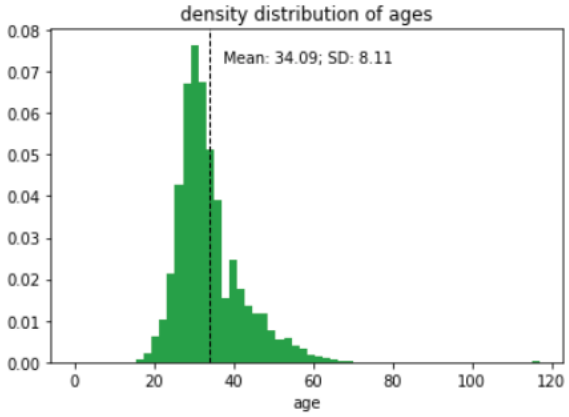

Figure 5 ages' density distribution: the cloth rental service is popular among younger population.

Detailed data general descriptions are summarized in the table below (Table 1).

To find the hidden correlations among between different pairs of attributes, we create such a heat matrix, wishing to capture if strong associations exist in our data set. Lighter color means the correspondent attributes have higher association. Apparently, size and weight are highly correlated. About 84.5% changes in weight can be explain by changes in size. Size and height of a person is also slightly correlated. Expectedly, height and weight are correlated as well. Also, age and size have a slight positive correlation.

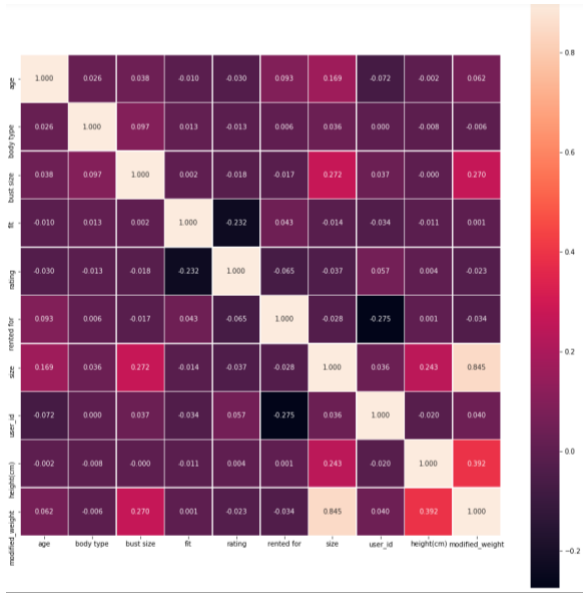

Figure 5 heat matrix shows the correlation coefficients for every pair of attributes in the data set.

Interestingly, the rating and fit are negatively correlated. About 23% decrease in rating could be explained by decrease in "fit". This finding demonstrates our previous assumption that people who think their clothes are fit to them would give higher ratings and vice versus.

	age	bust size	rating	size	height(cm)	modified_weight
count	146381.000000	146381.000000	146381.000000	146381.000000	146381.000000	146381.000000
mean	34.089800	5.725108	9.081985	11.437919	165.768155	137.209870
std	8.113217	7.182163	1.437853	7.826784	6.754444	21.540182
min	0.000000	0.000000	2.000000	0.000000	137.160000	50.000000
25%	29.000000	1.000000	8.000000	4.000000	160.020000	123.000000
50%	32.000000	4.000000	10.000000	9.000000	165.100000	135.000000

Table 1 general data descriptions. Mean, rows count, standard deviation, min, max, quantile values for the data set.

2. PREDICTIVE TASK

In this study, we will be using ***customers' review text*** and ***explicit users and items' feature*** (i.e., age, rented for) in dataset to classify ***their fit feedbacks*** (i.e., small, fit or large). Because we noticed there are multiple features in the dataset demonstrated high correlations with 'fit' (*figure5*), one-hot-encoding those features and applying it to Logistic Regression would be an efficient and appropriate for the baseline model. And review text is essential for performing semantics analysis for our more advanced models, like TF-IDF, CNN. Besides,

Data are cleaned: some values like weights/heights are converted into valid numbers. Categorical columns are transformed using label encoder and one-hot encoded in order to fit the linear regression model.

We performed a train-test-split to avoid overfitting on the data, which has the ratio: train_size/test_size = 2/1. After that, we trained our models on training set and evaluate models' performances on test set. To be more specific of the evaluation, we adopted ***accuracy*** and ***balanced error rate*** as our metrics since the label distribution is imbalanced. We observed that the data we used have imbalanced labels, so to have a fair evaluation, we will be using the balanced error rate to average the side-effect of imbalanced labels.
Our baseline model utilized Logistic Regression with general user & item's features from the dataset.

Because this is a typical classification problem, we have learned Naïve Bayes, Logistic Regression and SVM in class. In the explorative analysis section, the heat map (Figure 5) implies the necessity of using Logistic Regression to predict whether the cloth is fit to a particular customer. Compared with Naïve Bayes, Logistic Regression didn't depend on the assumption that features are independent from each other. Compared with Support Vector Machine, it will not be efficient enough to treat data set of such a large size. Regarding logistic regression's exceptional performance on both multi- and binary classification, we decide to use logistic regression as our base model. And our more advanced models: TF-IDF Model, Ensemble Model and CNN utilize only the review text column to conduct the classification process because customers will write out their opinions of cloth fit in the reviews, which makes this feature contains much information.

3. MODEL

First, we decided to use features that are easily accessible in the data set, such as weight, age, rating, etc. We trained a Logistic Regression based on these features(one-hot-encoded). Then, we tried to explore more on the review text of the data. We developed a TF-IDF matrix out of the review text and fed it to a Logistic Regression to get prediction on whether the cloth rented was fit or not. In order to keep increasing the prediction performance, we trial out multiple combinations of the feature-based model and TF-IDF model using methods like Ensemble Learning and fed into a predictor. We also trial out Ensemble Learning on combination of feature-based learner and TF-IDF based learner. Finally, we decided to adopt a neural network approach in order to see how a non-linear predictor would perform on this task.

Logistic Regression: the optimal problem is minimizing the negative log likelihood of the original logistic function plus the regularizer. For sklearn package, it utilizes the "liblinear solver" which implemented the trust region Newton's method. Newton's method is similar to gradient descent. Not only using the gradient as the steepest descent route, it also takes advantage of the hessian inverse of the matrix in order to get faster approach to the minimizer. To optimize our model, we tested out different C values to get a higher accuracy.

TF-IDF: We adopted the sklearn's TF-IDF vectorizer to perform the classification task. The package gave us efficient to transform review text in training set to vectors, which can be fed into our classification model later on. Besides the efficiency on implementation level, TF-IDF is a great algorithm that can return documents that are highly relevant to a specific query. However, TF-IDF can't distinguish singular and plural words, which decreases words' expression value and requires a cleaner review text but overall, it's an ideal text vectorizer for the classification task. But during the training process, we encountered the out of RAM problems, so we had to shrink the vectors' sizes.

Ensemble Learning: We switched to Ensemble Learning approach because it could combine the result for two previous models in order to generate a better performance on classification. But the weakness is it's hard to explain the features since they are coming from previous models' prediction. However, it actually yielded good result

Compared with TF-IDF and Ensemble learning, feature based logistic regression is more efficient in training and utilizes a variety of information in the dataset while TF-IDF only concerns with text review in the dataset. However, due to the simplicity of the logistic regression model, it's regarded as our baseline model and the predictions are used for later Ensemble learning. For the TF-IDF model, the customers' reviews are good measurements of clothing fitness and the TF-IDF would extract representative words in the corpus. Nevertheless, the degree of the cleanness of the corpus would be a weakness of the model since removing stops words, distinguishing singular and plural words and stemming have both pros and cons. Moreover, the result matrix occupies great space compared to feature base logistic regression and ensemble learning. As regard to the ensemble learning model, it combines multiple machine learning models we used to improve the overall performance.

Convolutional Neural Networks: For neural networks, we only investigated in review texts. We have successfully performed two neural networks based on

our review texts. For both methods, we used the same feature engineering. We cleaned the text reviews by converting them to lower case and removing the punctuations. Then, with the cleaned review text, we created a vocabulary, which are the words in the review text, to integer mapping dictionary. With the dictionary, we encoded every word of the review to an integer and turned each of the review text into a list. We also encoded our labels into integers. Thus, with the feature engineering on review texts, we have successfully transformed our review texts into lists of integers and our labels into integers. After the encoding, we removed the empty reviews. In order to make all the reviews the same length, we padded features for the reviews that are not long enough and cuts off the reviews that are too long.

One of the models we used is without convolutional layer, and the other one we used contains convolutional layers. For both neural networks, we used word embedding to start and used a fully connected layer to output each with three features in order to have three classes. For the neural network with convolutional layer, we use one-dimensional convolutional layer to train our model and we concatenate the filters which are the three convolutional layer. The specific structures of our models are shown as the followings:

Model Without Convolutional Layer	
Word Embedding	Input Dimension: 43441 Output Dimension: 256
Fully Connected Layer	Input Features: 256 Output Features: 3

Table 2 The Specific Structure of the best performed neural network without convolutional layer

Model With Convolutional Layer	
Word Embedding	Input Dimension: 43441 Output Dimension: 256
Convolutional Layer 1	Input Channels: 100 Output Channels = 100 Kernel Size = 3 Padding = 1 Stride = 1 Activation Function = ReLU
Convolutional Layer 2	Input Channels: 100 Output Channels = 100 Kernel Size = 4 Padding = 1 Stride = 1 Activation Function = ReLU
Convolutional Layer 3	Input Channels: 100 Output Channels = 100 Kernel Size = 5 Padding = 1 Stride = 1 Activation Function = ReLU
Concatenate the three convolutional layers	
Drop Out Layer	Probability: 0.5
Fully Connected Layer	Input Features: 300 Output Features: 3

Table 3 The Specific Structure of the best performed neural network with convolutional layer

Unsuccessful attempt along the way: Since TF-IDF matrix has a high dimension, we ran into out of memory issue when concatenating features and TF-IDF matrix together. We solved it by switching to a bigger RAM computer and decreasing the dictionary size of the TF-IDF model. Since Neural Network models are widely used in text classification tasks and their performances usually outperform simple machine learning models, we planned to use Recurrent Neural Network and Long Short-Term Memory, but we failed to do it because we could not figure out how to make the output size to three, since we have three classes.

4. Literature

The cloth size problem has been studied fairly so far. One study focuses on the semantics of customer review text and “handle imbalance labels of data using metric learning approaches with prototyping” [1]. In this paper we will use the same dataset as Rishabh et al.
Our method differs from this study in that we concentrate on building a better vector representation of review data and evaluating our model on metrics spaces, using balanced error rate.

The data set we used come from this website: https://cseweb.ucsd.edu/~jmcauley/datasets.html#clothing_fit. It is contributed by Rishabh Misra, Mengting Wan, and Julian McAuley and used for modeling the semantics of customers’ fit feedback. Since customers have their own preference of fit regarding different sizes of clothes (small, fit, or large), the semantics of their feedback after purchasing/renting the cloth may better capture their true preferences.

Other studies like G Mohammed Abdulla and Sumit Borar try to solve the fir problem using a “skip-gram” latent factor model. They concluded that the most popular sizing is more intricate and costumers with “petite, boxy and oval shapes” encounter some levels of difficulties in fitting the right fit. Conclusively, our findings share some similarities with Professor

McAuley, which emphasize on analyzing the relationship between fit and review text. Customers' feedbacks usually demonstrate their opinions of the clothing fit, so part of our work focused on utilizing the text as only feature and applied it to TF-IDF model and Recurrent Neural Network. Our model has the same emphasize on semantic analysis as Professor McAuley. Although we adopted different models to approach the same classification tasks, we ended up finding review text is essential factors for predicting fit feedbacks.

5. CONCLUSION

Firstly, let's look at the most straightforward data of our models, the accuracy and we also include the balanced accuracy rate. The baseline model which implemented using logistic regression runs on features: fit, age, body type, rating, rented for, size, height, weight. It could only achieve an accuracy of 0.707. The Balanced Accuracy Score is 0.470.

The TF-IDF model uses only the text as the single feature and surprisingly gives explicit improvement. The accuracy score could achieve 0.775 while the balanced accuracy rate could be 0.644.

The Ensemble model combines the baseline model and the TF-IDF model. The accuracy as well as the balanced accuracy rate are higher and it achieves accuracy 0.778 and the balanced accuracy rate could achieve 0.680. Both of the scores outperform other models.

For neural networks, for the model without any convolutional layer, we obtained an accuracy of 0.735 and balanced accuracy score 0.682 but for the model with concatenated convolutional layers, we obtained an accuracy about 0.825 and balanced accuracy score 0.791.

During the implementation, we also trialed Bag-of-words model in respect with TF-IDF. However, the former feature representation didn't work out well because Bag-of-words model didn't capture the important words that specifically appeared in this corpus.

With cross validation, we tuned our parameter to achieve the best performances on test set. For logistic regression, we set "class_weight = balanced" to incorporate the imbalanced label. When we transform the text to vector using Tfidf vectorizer, we removed stop_words because those words don't represent actual meanings.

For neural networks, we focus on the changes for two parameters, the length of the reviews and the embedding dimensions. Based on the distribution of length of the reviews, which is shown in Figure 6, we choose lengths 50, 100, and 150 for the review lengths and review length list of 100 performs the best. For the selection of the number of embedding dimensions, we choose 100, 256 and 512 to experiment. With word embedding dimension of 100, our model performs the best.

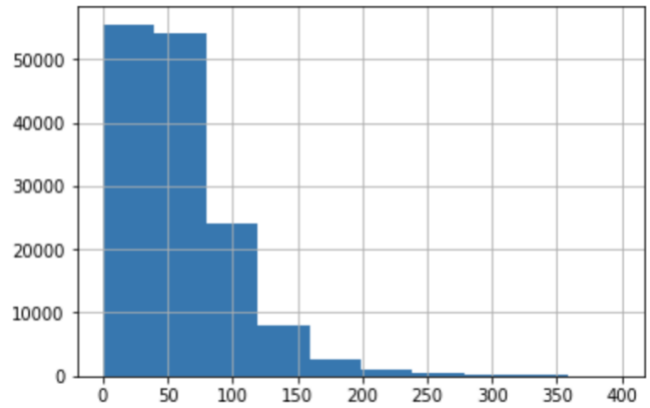

Figure 6 The above figure shows the distribution of the length of reviews.

To prevent overfitting for the neural networks, we choose to keep track of validation losses for each neural network and stop training after two consecutive increases in validation losses. Also, since the dataset is unbalanced, in our neural network, we shuffle the dataset and use minibatch to train the model in order to decrease the effects of unbalanced dataset. The below shows the training and validation accuracies and losses.

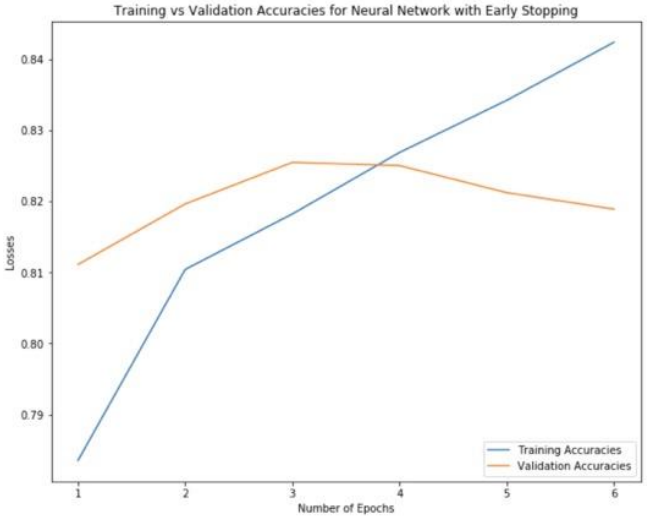

Figure 7 The above figure is the training and validation losses for the neural network with better performances.

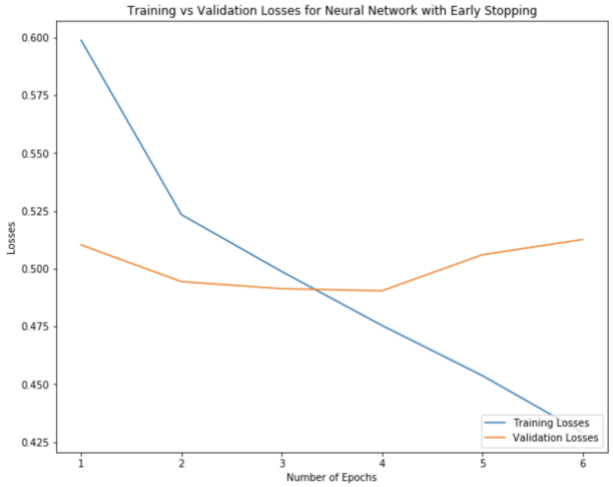

Figure 8 The above figure is the training and validation accuracies for the neural network with better performances.

Our models can be applied to the clothing rental industry in order to improve customers' shopping experience. Clothing company could label their clothes' fit(run large or small) to maximize customers' satisfaction.

To further improve our over performance, we could expand our data and eliminate the label imbalance in idealized situation. Moreover, we could utilize more computing resources to train the neural network in order to achieve better result.

6. REFERENCE

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

[2] G Mohammed Abdulla and Sumit Borar. 2017. Size Recommendation System for Fashion E-commerce. KDD Workshop on Machine Learning Meets Fashion (2017).