Product Fit Prediction From Customer Review Feedbacks

ABSTRACT

This research introduces a method for predicting clothing fit based on customer review data, utilizing the RentTheRunway dataset. Alongside our model development, this paper also dives into key insights drawn from the dataset, outlines the predictive tasks, and discusses relevant literature in the field.

1.DATASETS

We introduce the dataset from the website *RentTheRunway*, a platform facilitating the rental of women's clothing for diverse occasions. This dataset contains 192,543 reviews of clothing rental from the website with a rich array of information, including user and product ids, detailed user profiles highlighting body types, and comprehensive product details such as categories. Additionally, it provides user-generated content, including ratings, reviews, and specific feedback on the fit of the clothes.

Statistics of the dataset are provided in Table 1. Note that the datasets are notably sparse, as large amounts of items and users are involved in only one transaction each.

Statistic	Count
# transaction	192,543
# user	105,571
# item	5,850
# user with 1 transaction	71,824
# item with 1 transaction	314
# fit	3
# rating	5
# bust size	106
# body type	7
# rented for	9
# category	68

Table 1: General dataset statistic

Upon examining the dataset's missing values, as detailed in Table 2, it becomes apparent that the gaps are mainly found within columns pertaining to user characteristics. This trend may stem from a reluctance among users to disclose personal details. To assess the impact of these missing values on the data's integrity, we conducted tests wherein missing entries in categorical attributes, such as 'bust size' and 'body type', were filled with the mode value, while gaps in numerical attributes like 'age' and 'weight' were supplanted with the mean value of the column. The analysis revealed that these imputations did not significantly alter the data's underlying patterns. Consequently, we proceeded with the replacement of all missing values as tested.

weight	29,982
bust size	18,411
body type	14,637
age	960
height	677
rating	82
rented for	10
fit	0
user_id	0
item_id	0
category	0
size	0
review_text	0
review_summary	0
review_date	0

Table 2: Number of Missing values

During the exploratory data analysis phase, we identified a trend that warrants further investigation. As depicted in Figure 1, there is a paradoxical pattern where users are more likely to report items as too small when the average item size increases, and conversely, items are often reported as too large when the average item size is smaller. This observation suggests that user perceptions of fit may not align with standard sizing metrics, indicating a potential discrepancy between

size labels and actual fit. This insight will be integral to our continued analysis and could lead to more nuanced understandings of customer satisfaction drivers.

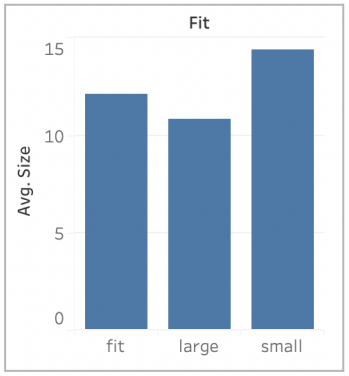

Figure 1: Fit feedback v.s. Average size

2.Predictive Task

For this dataset, our objective is to predict user fit feedback using available features. To measure the performance of our predictive model, we employ accuracy as the metric of evaluation.

In assessing feature relevance, we observed that several factors influence fit perception beyond the previously discussed impact of size. Notably, Figure 2 illustrates that the ratings by the users are highly imbalanced. The users are generally satisfied with the transaction and tend to give a high rating, causing the bar chart to be highly left skewed. According to the proportion of the fit feedback in each rating, items with higher ratings predominantly received feedback that they fit well. In contrast, the lower ratings feature a more balanced distribution of fit feedback, with similar proportions for items that fit well, were too small, or too large. This suggests that customer satisfaction with the fit of the clothing correlates positively with higher ratings. To rigorously evaluate the influence of these features, we aim to predict fit feedback using all provided dataset features, excluding 'review_text', 'review_summary', and 'review_data' due to their less quantitative nature.

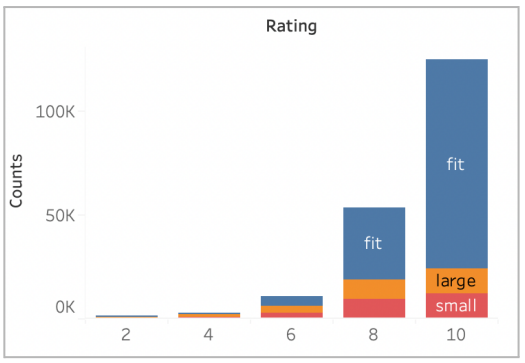

Figure 2: Distribution of user Ratings & fit

Prior to model training, we engaged in comprehensive data preprocessing to refine our features. Initially, we isolated numerical values from the 'weight' column, removing any textual units (e.g., extracting '132' from '132lbs') and converted the entries to a floating-point data type. Subsequently, to facilitate a more intuitive analysis of the 'height' variable, we introduced a new column titled 'height_cm', converting heights from inches to centimeters (for instance, translating 5' 6" to 167.64 cm). Following these adjustments, we transformed the data types of our columns as detailed in Table 3, ensuring the dataset is optimally structured for the subsequent phases of modeling.

Column	Dtype
fit	category
user_id	object
bust size	category
item_id	object
weight	float64
rating	float64
rented for	category
body type	category
category	category
size	int64
age	int64
height_cm	float64

Table 3: datatype of each column

3.Model Selection

3.1 Training Approach

In our analysis of the runway prediction dataset, each order provides

information of User ID, Item ID, and independent features. We aimed to explore the impact of these features to our interaction model. Our objective is to determine their potential to enhance the overall accuracy of the prediction model.

3.2 Factorization Machine

Factorization Machines are a type of predictive model well-suited for handling datasets where interactions between features play a critical role in modeling, capturing interactions between variables in high-dimensional sparse matrices. This modeling is facilitated by the use of factorized parameters, enabling the model to efficiently process a large number of potentially interconnected features. This approach empowers Factorization Machines to uncover complex patterns and relationships hidden within the data, thus offering a significant advantage over traditional linear models. This advantage is particularly notable in scenarios characterized by sparse data, where conventional models might struggle to identify underlying trends and interactions effectively.

We initiated our modeling process by applying one-hot encoding to the categorical features (user_id, item_id, bust size, rented for, body type, category). Given that most of these feature elements are zeroes, we utilized a sparse matrix to enhance the efficiency of computation and data storage. Our initial model was developed using FastFM's sgd.FMClassification, with parameters set to their default values. To evaluate our model, we constructed three distinct models, each based on a different formulation of the output variable y. In the original dataset, the label column y comprises three outcomes: fit, small, and large. We redefined this in Model A as {fit: 1, small: -1, large: -1}, in Model B as {fit: -1, small: 1, large: -1}, and in Model C as {fit: -1, small: -1, large: 1}. To consolidate these models, we adopted a final prediction strategy. If the prediction is {fit: 1, small: -1, large: -1}, we classify it as 'fit'. If the prediction falls into either {fit: -1, small: 1, large: -1} or {fit: -1, small: -1, large: 1}, we classify it as 'unfit'. In any other case, the default classification is 'fit'. This modeling approach yielded an accuracy of 0.7382 (Table 4).

Prediction	Classification
{fit: 1, small: -1, large: -1}	Fit
{fit: -1, small: 1, large: -1}	Unfit
{fit: -1, small: -1, large: 1}	Unfit
Else	Fit

Table 4: Final Prediction Strategy

3.3 Optimization

With the primitive model, we encountered several difficulties to be solved. Our initial focus was on refining the training parameters to improve outcomes. To achieve this, we implemented Optuna, an optimization library, setting its parameters as follows: *n_iter* = (500, 2000), *init_stdev* = (0.0001, 0.25), *l2_reg_w* = (0, 0.2), *l2_reg_V* = (0, 20), *rank* = (1, 5), *n_trials* = 1000. Another critical issue during this process was the imbalance in the training dataset between the 'fit' and 'unfit' data, with counts of 'fit': 141985, 'small': 25776, and 'large': 24690. This imbalance hindered effective training of the features, leading to skewed results in our models. For Model A, while the test data (y_test) comprised 35718 instances of 'fit' (1s) and 12395 of 'unfit' (-1s), the predicted data (y_pred) showed a disproportionate prediction of 46946 'fit' (1s) and only 1167 'unfit' (-1s), resulting in an accuracy of 0.742. Similarly, Model B's predictions were heavily biased towards 'unfit' with a prediction count of 48113 'unfit' (-1s), leading to an accuracy of 0.8671. Model C exhibited a similar trend, with a prediction of 47790 'fit' (1s) against 323 'unfit' (-1s), achieving an accuracy of 0.842. This pattern indicated a significant challenge: the model's inability to predict 'unfit' data

effectively due to a lack of representative ‘unfit’ samples.

To address the imbalance in our dataset, we implemented Synthetic Minority Over-sampling Technique(SMOTE). SMOTE is a widely used algorithm in machine learning designed to balance datasets by artificially generating samples for the minority class. It operates by selecting from the top ***K*** most similar minor instances based on the Euclidean Distance metric. With the implementation of SMOTE, we were able to significantly increase our sample size to a total of 306,483 data points. This included 106,483 instances of 'fit' data, and 100,000 instances each for 'small' and 'large' data. This enriched dataset contributed to an enhanced model performance.we observed an optimization in our model's accuracy, which rose to 0.7481. The individual model performances were as follows: Model A: Predictions were 45,536 for 'fit' (1) and 2,536 for 'unfit' (-1), with an accuracy of 0.748. Model B: Predictions were 47,965 for 'fit' (1) and 148 for 'unfit' (-1), with an accuracy of 0.867. Model C: Predictions were 47,562 for 'fit' (1) and 551 for 'unfit' (-1), with an accuracy of 0.874. The cumulative accuracy of all models post-SMOTE implementation stood at 0.7386.

3.4 Review and Improvement

Dissatisfied with the result of the model, we dug deeper to explore the unresolved issues. (1) **Bias Toward 'Fit' Predictions**: The outcome remains skewed towards ‘fit’. Shown from the result, the inability for the model to predict ‘unfit’ output impacted the overall accuracy and could be the primary reason to impede the improvement of the accuracy. (2) **Overlooking Ordinal Relationships**: Another oversight in our initial approach was the failure to recognize the ordinal relationship between the categories 'small', 'fit', and 'large'. Acknowledging this hierarchy potentially increases the ‘unfit’ dataset. For example, items that are smaller than those categorized as 'small' could be labeled as 'unfit', thereby enriching the dataset with more nuanced information. (3) **The Cold Start Problem**: Cold Start is a major issue in our mode. The dataset predominantly features popular items, which are well-represented and thus effectively trained. In contrast, less popular items suffer from insufficiently trained features and interaction parameters, leading to a gap in the model's comprehensive learning and predictive ability."

4.Related Literature

4.1 Dataset

We are utilizing the Rent The Runway [2] dataset, a comprehensive resource focused on clothing fit measurements. This dataset was originally introduced by researchers in our field who study the same research question. Within this dataset, we find a rich collection of customer reviews, along with detailed information about each item and user. This information includes features like fit, user_id, item_id, rating, rented for, body type, category, size, and additional data about customers' body measurements. The most important attribute is the "fit" column. It records users' feedback regarding the fit of an item, categorized into “Small,” “Fit,” or “Large.”

4.2 Literature Outline

The paper[1] focuses on product size recommendation and fit prediction, which are crucial for enhancing customer shopping experiences and reducing product return rates in online fashion retail. Its plan is to build a new predictive model to generate whether an item is fit, small, or large to the user based on customers’ fit feedback. From the dataset we’ve just introduced, there are three

challenges to overcome: subjectivity in fit feedback, ordinal nature of feedback, and label imbalance issues.

Customers' fit feedback depends on both objective product size & customer body measurements and subjective elements such as the style and design of the product. The authors develop a model that factorizes the semantics of customers' feedback. This means that instead of just looking at the size labels, the model tries to understand the underlying meanings in the text, for example, terms customers use to describe fit, such as "baggy" or "tight."

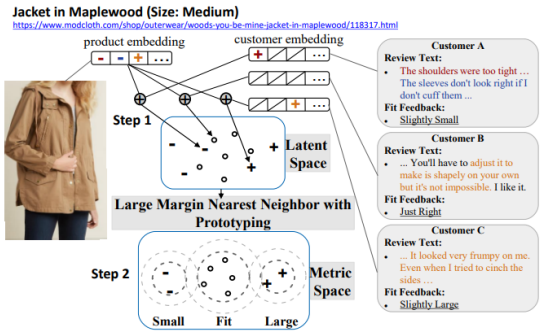

Figure 3: Workflow of the proposed framework from the literature.

The feedback's ordinal nature, which means small, fit, large have a logical sequence, is addressed through an ordinal regression procedure. This method learns representations that preserve the order of labels, ensuring that the model keeps the sequence of fit from small to fit to large. By doing so, the model can be more accurate to predict whether an item is fit or not to a certain customer for new products.

To handle label imbalance issues, where most feedback is "Fit", is tackled using metric learning and prototyping. Prototyping selects key examples from the data, which helps to balance the classes and reduce noise from outliers. Metric learning improves the local data neighborhood by ensuring transactions with the same fit feedback are closer together, while those with different feedback are farther apart. This approach helps the model to better distinguish between the classes despite the imbalance in the original data.

4.3 Methodology

Fit semantics learning and metric learning approach are two key concepts to discover true relationships between data points. Fit Semantics Learning quantifies customers' subjective feedback on product fit, converting qualitative feedback into quantitative data. On the other hand, the metric learning approach defines a distance metric to scale similarities or differences between customer transactions, enhancing the accuracy of fit predictions.

To model fit semantics, the researchers adopted a latent factor formulation. They assign a score to each transaction that indicates the fitness of an item to a specific customer. For example, if a customer (denoted by c) purchases a product (denoted by p), which is a certain size of a parent product (denoted by pp), the model computes a fitness score for this transaction (denoted by t). This fitness score is based on latent factors and bias terms that aim to capture the complex relationship between customer preferences and product characteristics. This approach allows the model to quantify how well a product fits a customer, taking into account the subjective aspects of fit feedback.

$$f_w(t) = \left\langle \underbrace{\mathbf{w}}_{\text{weight vector}} , \overbrace{\alpha \oplus b_{t_c} \oplus b_{t_{pp}}}^{\text{fit bias terms}} \oplus \underbrace{(\mathbf{u}_{\mathbf{t_c}} \odot \mathbf{v}_{\mathbf{t_p}})}_{\text{fit compatibility}} \right\rangle \quad (1)$$

Formula 1: Fitness score model

The metric learning approach is introduced to address the challenge of label imbalance in fit feedback. The method begins by altering the distribution of training data through a process called prototyping, which involves re-sampling from different classes. This helps to balance out the number of examples in each fit category. The Large Margin Nearest Neighbor (LMNN) algorithm is

then used to refine this process. LMNN works by optimizing a distance metric to ensure that transactions with the same fit feedback are closer together, while those with differing feedback are further apart. This creates a metric space where the distances between points accurately reflect the differences in fit feedback, thus improving predictions about product fit based on customer feedback.

4.4 Model and Conclusion

The final model presented in the paper, denoted as K-LF-ML, integrates a metric learning approach with K-dimensional Latent Factors (K-LF) to produce the final classification. This model is different from other models in the paper because it employs metric learning instead of Logistic Regression (LR). The primary goals of this approach are to effectively capture the fit semantics over 'true' sizes from both products and customers, learn robust latent representations, and address label imbalance issues. The model's performance is evaluated by Area Under the Curve (AUC) metric, which is a common evaluation method in ML to assess the accuracy of a model's predictions, especially in classification tasks.

Dataset/Method	(a) 1-LV-LR	(b) K-LV-LR	(c) K-LF-LR	(d) K-LV-ML	(e) K-LF-ML	improvement (e) vs. (a)	improvement (e) vs. (b)	improvement (e) vs (c)
ModCloth	0.615	0.617	0.626	0.621	0.657	**6.8%**	6.5%	4.9%
RentTheRunWay	0.61	0.676	0.672	0.681	0.719	**17.9%**	6.4%	7%

Table 2: Performance of various methods in terms of average AUC.

Table 5: Performance of various methods in terms of average AUC.

To execute the model, the whole dataset is divided into training, validation, and test sets with an 80:10:10 split. The model employs ℓ2 regularization during ordinal regression optimization, and hyperparameters are tuned by grid search and Bayesian optimization. In conclusion, the proposed K-LF-ML model outperforms other models with 0.719 in terms of average AUC.

In summary, the predictive framework generated significant effectiveness in predicting whether a certain product size item is fit with a specific user. Thus, the paper offers positive evidence that our research direction is viable, suggesting that the information from Rent The Runway dataset can be useful in accurately forecasting the fit of clothing items in transactions.

5. Conclusion

Compared to the approach outlined in existing literature, where a new model was developed from the ground up specifically to address the unique challenges of clothing fit datasets, our methodology took a different route. We began by selecting the most suitable recommender model for predicting product fit. Only after this initial choice did we address the issue of label imbalance in the data. Consequently, the inherent problems in the dataset continue to significantly impact the predictive accuracy of our model. Therefore, this report has explained the reasons why generic recommender models are not fully effective for the specific problem we are addressing. Developing a tailored predictive framework is essential for effectively managing datasets such as Rent The Runway, and it is key to producing accurate predictions.

REFERENCES

[1] Rishabh Misra, Mengting Wan, and Julian McAuley. 2018. Decomposing Fit Semantics for Product Size Recommendation in Metric Spaces. In RecSys '18.

[2] Julian McAuley. Clothing Fit Data. In Recommender Systems and Personalization Datasets website. https://cseweb.ucsd.edu/~jmcauley/datasets.html#clothing_fit

[3] fastFM 0.2.10 documentation. https://ibayer.github.io/fastFM/tutorial.html#logit-classification-with-sgd-solver

[4] Julian McAuley.Visual Clean slides. In CSE 158/258 (MGTA461/DSC256): Web Mining and Recommender Systems https://cseweb.ucsd.edu/classes/fa23/cse258-a/