Say Yes to the Dress? An Exploration of Clothing and Fit

Aditi Shrivastava, Andrew Ngo, Bobby Tatum, Lena Krockenberger

I. The Dataset

For this task, we used the RentTheRunway dataset found at cseweb.ucsd.edu /~jmcauley/datasets.html, which consists of 192,544 different raw data points. Each data point, representing a user's review of a rented article of clothing, consists of various quantitative and qualitative information, including user height, weight, body type, age, bust size, user rating, textual review, and item category, among some others. The target variable for this dataset is contained in the fit column, which indicates whether the article of clothing was a good fit.

There were 7 body types included in the data, such as "apple" and "petite," as well as categories of clothing such as "cardigan" or "dress." As a result of the different categories of clothing, we were hesitant to use "size" as a feature across all articles of clothing generally. For example, a size 4 dress may not correspond to a size 4 jumpsuit. The figures below compare the body type and category features' size distributions, and show that body type is much more indicative of fit (Figure 1.)

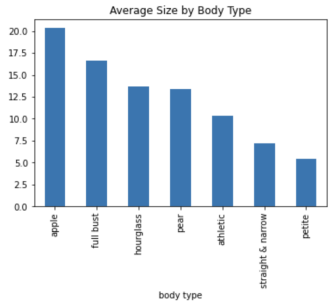
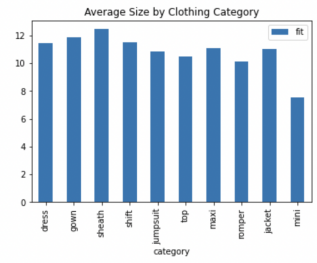


Figure 1 (left)

Figure 2 (below)



The data also includes textual data in the form of review text and review summary, which we hypothesized would be highly important additions to the model, and could be processed via tf-idf, bag-of-words, n-grams, etc. To explore these features further, we identified the most common unigrams and bigrams for both well-fitting and ill-fitting clothes, and found that the words were quite contrasting, as expected (Figure 3 on the next columns.)

Figure 3: Notable Common N-Grams Contained in Review Text (N={1,2})

Well-fitting Items	Ill-fitting Items
'this dress,' 'very,' 'great,' 'perfect,' 'compliments,' 'comfortable,' 'a little,' 'loved'	'but,' 'size,' 'not,' 'a little,' 'and i,' 'if,' 'a bit,' 'tight,' 'length,' 'fabric,' 'short,' 'large'
General positive sentiment	Generally concerning item fit and size

Next, we observed that some data features' distributions are heavily skewed when comparing items that fit and items that did not fit. For instance, the average rating for well-fitting items was about 9.3, while the average rating for ill-fitting items was about 8.47, indicating that while most users gave excellent ratings to their reviewed clothing, there is a notable discrepancy in rating between items that fit and did not (Figures 4 & 5 below.)

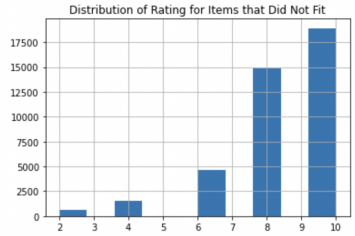


Figure 4

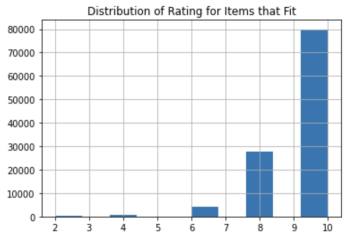
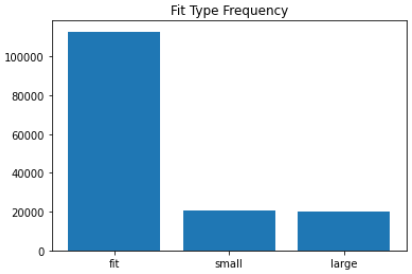


Figure 5

Additionally, about 73.7% of users stated that the article of clothing they reviewed was a fit. This is illustrated in figure 6 below, wherein the label distribution is heavily skewed towards 'fit,' as opposed to the non-fit categories 'small' and 'large.' To account for the skew in fit/non-fit distribution, we randomly dropped articles of clothing that fit so that we had an exact 50/50 split in fit/non-fit data points.

Figure 6



Another problem in the data presented itself in the form of null fields. Some users simply did not include a body type, a height, or a weight. We dropped all data points that did not include all the relevant information. After dropping the appropriate data points and fixing the skew of the data, we ended up with a total of 81,006 data points. Some simple statistics about these data points can be found in the table below (Figure 7.)

Figure 7: Simple Statistics (after dropping data points)

	Mean	Median	Mode	Min	Max
Weight	137.37	135	130	50	300
Height	65.26	65	64	54	78
Rating	9.08	10	10	2	10

II. The Task

For this dataset, we decided to try and predict, for any given user review of an item, whether or not that item ended up fitting that user. This is a binary classification problem, wherein we'll make recommendations based on whether the item is a predicted fit. This entailed changing the fit column values 'fit,' 'large,' and 'small' to simply 'fit' and 'not fit.' We then will evaluate our model using a simple accuracy metric (i.e. the percentage of correct fit/not-fit predictions), as well as its ROC curve and AUC score. We chose these metrics of accuracy because they give us a good idea of how well the model is predicting overall and where it fails, given that we care equally about false positives and false negatives.

We established two baselines to give us a starting point to compare our final models to. First, we created a baseline that collated the most popular items in the dataset (we chose a threshold of 33%, meaning that we selected the most popular third of all items.) If an item was in this set of popular items, we predicted that the item would fit the user. This baseline had an accuracy of about 51.27%. We then created a second baseline that predicted fit based solely on rating, choosing a threshold of ≥ 8 to be a fit. This baseline was slightly more successful than the first one, returning an accuracy of about 51.59%. Both of these baselines were trivial at best, and did not seem to actually predict better than a random choice whether an item would fit a user.

We then examined the dataset and referenced our previous EDA to determine which features might be useful in predicting fit. We determined immediately that size, temporal features, age, and ID features would not be useful in our final model, seeing as there was not much discrepancy among those features between well-fitting and ill-fitting clothes.

Ultimately, we decided to construct models that took in the user's rating of the item, the item review text, and the user's body type, height, and weight. These features are intuitively indicative of a clothing item's fit in the real world, and vary in ways that will best train our model to discern fit vs. not fit. Of course, these features are not usable without first parsing the data into workable numbers. Specifically, this entailed casting the 'rating' column to integer values (from string), and then leaving as is to act as an ordinal categorical variable. Next, we transformed the height and weight columns, by converting height to inches and stripping 'lbs' from weight. These columns were then normalized for the model. Finally, in order to be able to work with the textual data columns, we employed some textual data engineering, beginning first by creating a joint corpus from both the review text and summary columns, and then stripping capitalization and punctuation for appropriate tokenization. This left us with a bag-of-words containing unigrams on both clothes that did and did not fit.

Before proceeding with our model, as mentioned earlier, we handled the imbalance in the dataset using random shuffling, extracted our target binary 'fit' columns, and then split the data into separate train, test, and validation datasets (75%:12.5%:12.5% split).

III. The Model

In order to find the best combination of features to include in our final model, we created three classifications using varying methods and classifiers.

Our first model was a sklearn logistic regression model to categorize each data point into either "fit" or "non-fit." For this model, we first further engineered the body type and created a number of binary categorical variables using one-hot-encoding. The model also includes the user's height and weight. We chose to build the model using these features because we noticed during our EDA that certain body types (such as 'athletic' and 'hourglass' had higher frequencies of well-fitting clothing, and that weights and heights below a certain threshold also appeared to fit slightly better. However, this model notably did not include any information on any item specifications, which precluded to us that it may not account for abnormalities in the data such as unlikely or abnormal user style or fit preference. Unfortunately, this model did not perform very well, resulting in an accuracy of about 52.22%. After tuning the logistic regressor's hyperparameters and appending rating as an additional feature, the accuracy jumped up to about 62.34%. This showed us that while height, weight, and body type are not very predictive, the user's rating of the item was much more predictive. We also recognized the advantage of feeding a classifier the actual rating values rather than using a threshold on this feature. Compared to our baseline model, which predicted fit based on a rating threshold, using rating as a feature in a classifier performed much better.

Next, we explored a model using the Pearson Similarity metric on items and their ratings. For this model, an item's Pearson similarity was calculated using the ratings of all users who interacted with a pair of items. We believed that this might be a good model given that users would rate well-fitting items similarly to other well-fitting items, and the same for ill-fitting. Then, a fit prediction was made based on whether the most similar item exceeded a certain threshold and whether the most similar item was a 'fit' or 'not fit'. To determine the best threshold, a series of values were tried, but, notably, none yielded strong results. Rather, the best Pearson model had an accuracy of 49.7%, indicating that this similarity metric did not provide useful information. It's possible that an imbalance in item reviews contributed to this, as analysis yielded a high variance for number of reviews per item based on the data set distribution ($\sigma^2 \sim= 700$). Also, given that we found that most users gave excellent (high, >8) ratings to all clothing items regardless of fit, it's possible that all of the clothing items ended up being classified as quite similarly rated and therefore indiscernible in terms of fit.

Finally, we decided to employ sentiment analysis on the review text and review summary columns in the dataset. We hypothesized that these features would likely be quite indicative of fit, given that the review text contains details on the user's experience with the item. Also, we observed manually earlier that there was an observable distinction between words commonly found in well-fitting vs. ill-fitting items. After this, our task was to manipulate the textual data in a way that could be interpretable by the model. We experimented with different *n*-grams and chose the most common *k* words in both the review text and the review summary. Each feature vector consisted of the counts of the most popular *k* words as they appeared in the review text and review summary. From the results of our first attempt, we also appended the user's rating of the item to the end of this bag of words feature vector. This feature vector only included information about how the user reviewed the item, taking in no data about the user's body measurements. Using a pipeline to experiment with different values for *n* and *k*, we found that the most successful model used a mixture of unigrams, bigrams, trigrams, 4-grams, and 5-grams, with a dictionary size of

5000 words and an alpha of 1000. After training the model, we chose to use a Ridge Classifier, which uses regularization to reduce collinearity in the dataset, which may affect the accuracy. This model ended up being much more successful than the other two, yielding a final accuracy score of 77.77%.

Although the accuracy of our final model was relatively high which is a strength, we identified a few weaknesses. First, the data that the model uses for fit prediction is required from an existing review between the given user and item. Although for this task, this information is accessible, this model would not be able to predict on cold-start data or data where the review text and rating are not provided. In terms of applicability, our model would not be very useful in a real world setting to predict potential fit between any user and item pair. In addition, our model predicts a binary fit versus non-fit. In reality, there are three categories of fit, and our model does not capture the more specific "small" and "large" labels provided in ill-fitting reviews.

IV. The Literature

The dataset we used came from user ratings of articles of clothing on RentTheRunway. Specifically, we sourced this dataset from Julian McAuley's website. RentTheRunway is a platform that allows women to rent clothes for different occasions. Similar datasets to this one exist; one dataset is called ModCloth and is a website that sells vintage accessories and clothing for women.[1] Another, called the DeepFashion dataset was collected by the Multimedia Lab at the Chinese University of Hong Kong. One individual that we found on towardsdatascience.com used Convolutional Neural Networks (CNNs) on this dataset to predict labels for clothing.[2]

In our attempt to find literature that undertakes the same task that we are doing, we found a paper written by Rishabh Misra, Mengting Wan, and Julian McAuley.[3] This paper uses a much more sophisticated model to perform a similar prediction task on the very same dataset. To roughly summarize, the authors of this paper used a latent factor model to model fit semantics. They utilize two separate datasets in their model, RentTheRunway, and ModCloth. After formulating fit, they used projected gradient descent to optimize their objective function. Instead of just dropping "fit" labelled data to account for skewed distributions as we did, the authors used the Large Margin Nearest Neighbor metric learning technique. They used an average AUC (Area Under Curve) metric to evaluate this classifier.

Compared to our model, this paper had only a few similarities. We both accounted for skew in the "fit" labels, but their paper used LMNN to do this, while we simply chose random "fit" data points to drop. Additionally, while we opted for a simple logistic regression model, they used the Metric Learning model. Our model also does not use any latent features, opting instead to use a bag-of-words model to determine fit. Their model also is able to predict on cold-start data values given just a user and an item pair while ours cannot.

The results of this paper showed that their best model produced a highest average AUC value of 0.719 on the RentTheRunway dataset. This is lower than the 0.85 AUC value we had for our model which is most likely because their model predicts on cold-start data points. In general, the conclusions of this paper mirror some of the conclusions we made about the dataset and the task as we both had to find a way to account for the imbalance in labels. However, because of our differing approaches, and slightly different predictive tasks, most of our findings in terms of successes in our model differ from theirs.

1 https://modcloth.com

2 https://towardsdatascience.com/would-this-clothing-fit-me-5c3792b7a83f

3 Misra, Rishabh & Wan, Mengting & McAuley, Julian. (2018). Decomposing Fit Semantics for Product Size Recommendation in Metric Spaces. 10.1145/3240323.3240398.

V. The Results

Overall, our best model had a test set accuracy of 77.77%. This model was a Ridge Classifier with an L2 loss included (sklearn RidgeClassifier). As inputs, we used the user's review text to create a feature vector of the 5000 most popular unigrams + bigrams + trigrams + 4-grams + 5-grams, along with the user's rating for the item, and a bias term. Notably, this model far outperformed the baselines and other experimental models, including the ratings model, the Pearson similarity model, and the one-hot encoded body-type + height + weight LogisticRegression model.

Based on the success of the n-gram + rating model, we're confident that the review text + ratings fields of a review are strong signals for the fit of an item. Starting with the rating, its significance comes to light when analyzing *why* a user would poorly rate an item -- or rather, when an item poorly fits. Thus, while the dataset was heavily imbalanced when it comes to the rating field (mean = 9.08), this imbalance actually yielded useful information -- low ratings are highly correlated with poor fit. Supporting this, an ablation test on our first model (excluding the rating field, including height, weight, one-hot encoded body-type) indicated that the rating feature was highly important for classification accuracy (62.34% with rating vs 52.22% without).

Next, regarding review text, sentiment analysis revealed useful information to our model, as review sentiment largely correlated to fit. Logically, the reviewer's sentiment would correlate to fit as positive words would strongly associate with a positive experience with the clothing item; hence, a likely fit. Supporting this, our exploratory data analysis indicated 'fit' reviews commonly contained the following n-grams: 'great', 'perfect', and 'comfortable'. On the other hand, negative sentiment in the review text would indicate a poor experience with the clothing item, and thus, a likely non-fit. To illustrate, our exploratory data analysis revealed the following notable common n-grams for negative reviews: 'short', 'large', and 'tight'.

Overall, we're confident that this model found a significant impactful relationship between review text, rating, and fit. To support this, we plotted True Positive Rate against False Positive Rate in an ROC curve (Figure 8.) We found that our top performing model had an AUC of 0.85.

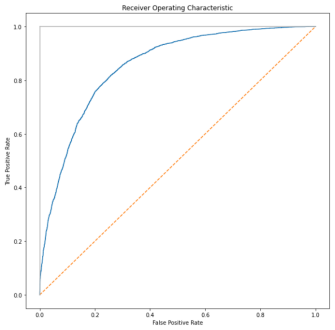


Figure 8

Also, we created a confusion matrix to gain better insight on where the classifier tends to fail, and found that the model performs slightly better for clothing items that did fit, indicating a higher false negative rate than false positive rate (Figure 9.) This entails that we may need more extensive data on ill-fitting items for the classifier to be able to better classify.

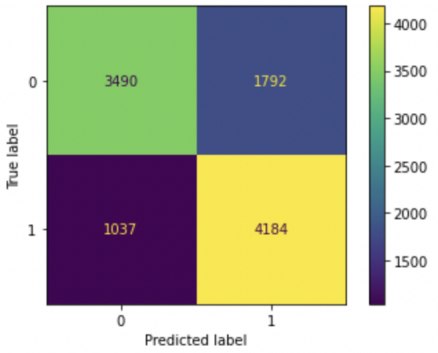


Figure 9

On the other hand, it was surprising that the weight and height and body-types contributed little to no information about the fit. Our model which combined these 3 features only achieved 52.22% accuracy, roughly equating to a guess on each test example. This might indicate that the relationship between weight, height, and

body-type is too complex for the model to capture. And, given the subjectivity inherent in a reviewer assessing their own body-type, it's possible that the true relationship between body-type, height, and weight is not captured in this dataset.

Additionally, another potential flaw in the weight, height, fit relationship may be found in the fact that weight and height alone might not strongly indicate fit. This could be explained by the fact that a fit on two users with identical weights and heights may vary based on body-type and personal-preferences, among other factors.