This One's Just Right

Maxwell Fang[1]

[1] *University of California San Diego, California, United States of America*

December 5, 2023

Online shopping is the new norm of attaining the things that we want. With a few clicks, we are able to purchase practically anything we want around the world. Nowadays, Black Friday is now just another Cyber Monday because no one is entering stores anymore but rather just ordering everything on the store websites. However, what if we want to customize our orders, specifically clothes? What if we want to try on a beautiful dress that exists on the other side of the world? We surely would not book a plane ticket just to try on a dress. In this paper, I propose a supervised framework to tackle the product fit problem, which gathers the customers' attributes and employs regression techniques to find the right size dress.

1 Introduction

Size customization of clothing online can be difficult. Considering each product can be tailored for their own uses and adjusted to their own country's demographics, we definitely cannot assume a large in another country will be a large in our own. For example, a small size in the United States can be a large size in Japan. Using categorical terms of "small", "medium", and "large" isn't very representative of size in many circumstances, and it can be deceiving. Instead, we should use quantifiable number sizes, like shoe sizes, to determine the exact size someone can wear.

To determine someone's size, I primarily will be using basic characteristics of a person, such as their height, weight, and other attributes. However, this will be quite difficult because people have has varying body types. Since conducting measurements will be ultimately impossible, we should categorize what type of dress is requested by consumer. Narrowing down a certain type of clothing can determine whether someone needs extra room for comfort or whether it should be a more slim or tight.

In this paper, I will be using different regression models to predict the size for each customer and comparing their performances against each other. The predictions for each model will be evaluated using mean squared error, and each model's generalization will be assessed on how well it performs on unseen data. Unlike previous work of predicting size for any type of apparel, I will not be predicting size for any type of clothing but only for dresses, gowns, and sheaths because there is not enough data int other categories to make strong predictions.

2 Related Works

The product size recommendation problem is quite recent with only a few studies so far. One approach captures the semantics behind customers' fit feedback alongside a metric learning technique that resolves label imbalance issues (Misra, Wan, and McAuley, 2018). From this paper, I utilized the one of same datasets (Renttherunway) for my model, but I did not want to use semantics. A more relevant approach that I explored was a Bayesian logit and probit regression model with ordinal categories. Instead of using quantifiable sizes, they use size fits categories of small, fit, and large. In addition, it relies heavily on the latent factors for customers and products that correspond to their physical true size (Sembium et al., 2017). The same authors also made an approach that recovers products' and customers' "true" sizes and used these as features in a classifier for fit prediction (Sembium et al., 2018). It was closer to what I wanted, but I wanted to work with quantifiable sizes.

There was a lot of research already done on latent factor models. In one latent model case, there was a study that captured subtle semantics of customer reviews with the Bert model to learn latent representations (Lin et al., 2020). However, in my case, I never intended to work with customer reviews in particular because people online may give inaccurate responses, which can be irrelevant to their thoughts about a project. As the search continued, I eventually ended up finding a paper predicting user measurements in order to make suitable sport garments. Their method consisted of size predictions with descriptive measures as inputs, predictions from regression analysis, and predictions from a shape model based on 3d measurements. Their input variables were straightforward demographics of age, gender, stature, and weight (Vleugels et al., 2022). Their main method of using regression analysis was the approach that I took for this paper. Regression analysis is one of the models that takes the least amount of processing power to analyze in comparison to 3D models that require additional data and a lot of processing power. It obviously was not the best predictor, but it had a fair performance based on its prediction capability. With only a couple base parameters, we can quickly predict measurements decently in a short amount of time.

3 Dataset

The dataset called "Renttherunway" comes from Professor McAuley's datasets. RentTheRunWay is a unique platform that allows women to rent clothes for various occasions. This data contains self-reported fit feedback from customers as well as other side information like reviews, ratings, product categories, catalog sizes, customers' measurements (etc.)

Statistics	age	height (cm)	weight (kg)
mean	34.019	165.790	62.319
std	8.108	6.738	9.932
min	1	137.16	22.68
med	32	165.1	61.235
max	117	198.12	136.078

Table 1: *General Renttherunway statistics.*

General statistics of the dataset is provided in Table 1. Note that there are some outliers. Considering all the customer data, it is unlikely someone who is one year old can rent clothing. Most buyers seem to fall around their mid-thirties. This won't have such a great impact because we have about 161,750 entries to work with. In addition to age, height, and weight, we are also taking into account of categories or type of clothing (e.g. ballgown, blazer, blouse). The downside to this data is that it is heavily dominated by three categories: dresses (78002), gowns (36831), and sheaths (16306). The total of these three is roughly greater than 80 percent of all the data. This demonstrates that people mainly go onto the platform to rent dresses, gowns, or sheaths. Since these three categories are most popular, we shall look at the distribution of dresses, gowns, and sheaths in relation to age.

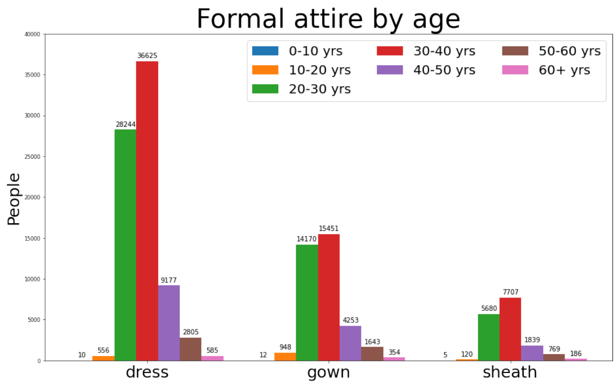

Figure 1: *Distribution of dreses, gowns, and sheaths.*

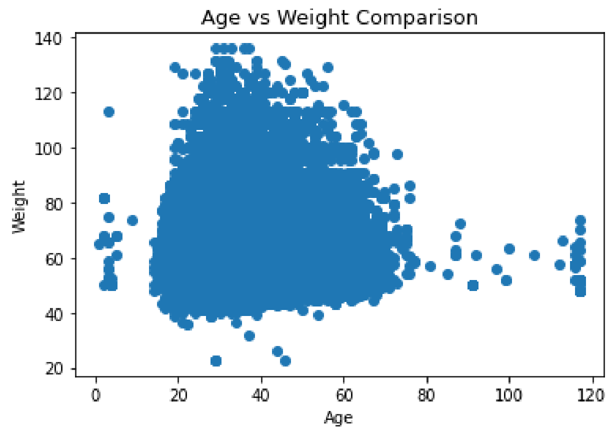

Figure 2: *Relationship between weight and age.*

From Figure 1, we can see that the number of consumers is generally the greatest for people in their 20 - 40s. Those who are over 40 seem to not rent as much; they could be less interested in trying or renting new clothes. People within their 20 - 40s could be the age where they try different styles or are more involved in party/events. I understand that many young-adults like to keep up with fashion trends and maintain their looks, so they could maybe want more of a slimmer fit. Another theory of why many people rent dresses between 20 - 40s is because the average age of weddings is around 30. People can be attending or having their weddings during this period.

As we get older, one of the questions that may be addressed is that we may gain weight as we get older. I wondered what the relationship between weight and age was going to be like, but I was left in awe instead. As age increased as shown in Figure 2, weight actually goes down a bit. There is seems to be weak correlation in Figure 2, but it could be kind of seen that weight slightly goes down as age increases. My theory is that there is actually a larger distribution between age 20 and 40 of young adults who are willing to try out new clothes. One interesting pattern is that there seems to be a couple data points close to 120 years. These might honestly be fake ages inputted by the customer; it could be a typo or just intentional. In addition, I

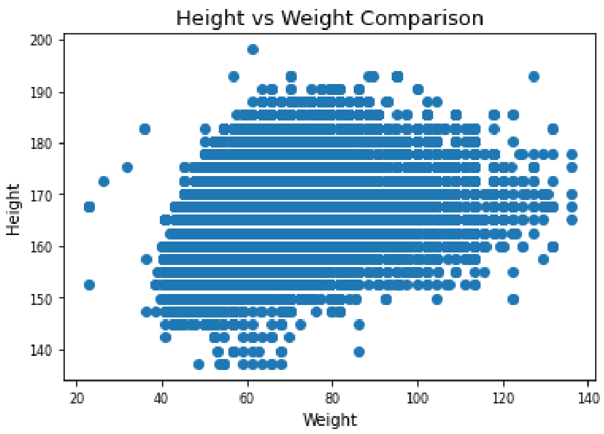

Figure 3: *Relationship between height and weight.*

think people at extremely young ages, especially those under 10, should not be renting clothes online unless they had adult supervision. If the age at 120 was true, then I would not expect that customer to ever return the dress.

One of comparisons I became interested in was the relationship between height and weight. I thought that the taller someone might be, they might be heavier in most cases. However, from Figure 3, I was proven wrong. Even at heights of 180 cm, there were numerous individuals that had weights under 60 kg. The shape of the points was just a huge blob, signifying that there is very weak correlation or no relationship between height and weight.

4 Methodology

Considering that my parameters are straightforward demographics, e.g. age, weight, height, and category, there may be relationships between these variables and size. For example, the heavier you get, the larger the size. A good approach would be to employ regression models, both linear and non-linear, for finding the best matching relationship given the features and the size. In the following sub-sections, I will first apply a linear regression model as a baseline. Afterwards, the non-linear models - random forest and gradient boosting - will be applied.

4.1 Experimental Setup

The data was cleaned by eliminating any values that could not be used. While exploring the entries, there were strange values that should not even be possible. For example, there were many entries that had the age at 0, which indeed would negatively affect the age parameter for my model. The attributes height and weight were not in integer or float format, so I had to convert them into float data structures. Height was was in the typical feet and inches format, and it was quite difficult to use even after extracting them with regex. After some time wrangling with the height attribute, I decided to use the metric system for both height (cm) and weight (kg). Lastly, I incorporated a LabelEncoder for the categories, so that they can be represented numerically. A brief summary of my pre-processing can be given below:

1. Remove extraneous and impossible values
2. Transform data structures
3. Apply conversions if necessary

Following pre-processing, the Training, Validation, and test sets are created using a 55:25:20 split of the data. Each model is trained on the training data, and assessed on its generalization through its training, cross validation, and test scores. The hyper-parameters (estimators, features, sample splits, and boostrap) are tuned on the training data for the Random Forest model.

4.2 Linear Regression Approach

Linear regression as you know is a linear model with coefficients to minimize the residual sum of between observed targets in the data and targets predicted by linear approximation. Everyone knows that the standard equation for a linear equation is y=mx+b, where m is the coefficient and b is the intercept. However, in this case, a more complex model is applied, where multiple parameters are taken in as coefficients. The equation for our model can be displayed as the following:

$$Size \simeq \theta_0 + \theta_1 \times (Age) + \theta_2 \times (Height) + \theta_3 \times (Weight) + \theta_4 \times (Category)$$

4.3 Random Forest Approach

I knew that linear regression itself was not the most optimal solution. Linear regression is far too simple and the relationships between the parameters and predictions will not be linear. Instead, I decided to look at non-linear regression approaches like Random Forest. Random Forest is a meta estimator that fits a number of classifying decision trees on various sub-samples of the data and uses averaging to improve predictive accuracy and control over-fitting. The idea of using decision trees can be great for this problem because it can be used in both classification and regression problems. We start at the root of the tree and based on certain categories, we can split based on variable outcomes until a leaf node is reached. In addition, ensemble learning can be applied to the data to find a more powerful predictive result. Ensemble learning is the process of using multiple models trained over the same data, and it takes the average results of each model. We can also have bootstrapping to randomly sample subsets of a dataset over many iterations for a given number of variables. By taking these into account, I can optimize my Random Forest model. The parameters that I chose to optimize are shown below:

- number of estimators : [10,20,30]
- max feature : ["auto", "sqrt", "log2"]
- min samples split : [2,4,8]
- bootstrap: [True, False]

With GridSearchCV, I fitted the model with my training data. As a result, the best parameters was having a total of 30 estimators, 8 minimum sample splits, 'sqrt' for max features, and having boostrap set to true. It is important to note that GridSearchCV may not boost the performance of the model greatly. There can still be unnecessary variables or outliers that throw the model off.

4.4 GradientBoosting Approach

The final regression model is Gradient Boosting. This model is great for non-linear relationships and great for dealing with outliers. It essentially is an ensemble of weak predictive models. The estimator that I used builds an additive model in a forward stage-wise fashion, where it optimizes arbitrary differentiable loss functions. In each stage a regression tree is fit on the negative gradient of the given loss function. Gradient boosting does take a while to run because it can be visualized as running multiple decisions trees and combining their results together. At one point, I tried optimizing the base gradient boosting model with GridSearchCV, but the code took too long to run. I tried experimenting with the learning rate, max features, and estimators, but the code never stopped running.

The time complexity of training gradient boosting takes O(tdx log n).

- t is the number of trees
- d is the height of the trees
- x is the number of non-missing entries

Using GridSearchCV to optimize gradient boosting might not be good idea because I use 5 k-folds of cross validation. If we have cross validation, GridSearchCV will default to shuffling the data. There is an option to make shuffle to be false in GridSearchCV, but I decided not to bother anymore with it. Whenever gradient boosting makes predictions for new data, the time complexity will be O(td).

5 Results And Analysis

Our results are summarized in Table 2. We find that more complex models, such as gradient boosters, performs slightly better than simpler models like linear regression. Model such as random forest may not perform much better, where there may be tendencies for it to overfit. One possibility for overfitting in random forest was it may have taken account of very detailed cases of people with people being very young or very old in the training data. From Figure 2, we can clearly see that there is great distribution of weight for people in almost every age group, indicating that there are too many variations. All in all, the model that came on top was gradient booster. Its training, validation, and test scores did not have great differences. In addition, gradient booster maintained the lowest MSE of 16.047.

Statistic	Linear	Random	Gradient
Train	0.757	0.833	0.770
CV	0.761	0.755	0.768
Test	0.753	0.752	0.762
MSE	16.670	16.627	16.047

Table 2: *Train, validation, test, and MSE scores for linear regression, random forest, and gradient booster models.*

Considering that the MSE was not far off for the linear model against more complicated models, it demonstrates it may be more worth to use a simpler model for predicting size based on basic demographic inputs. It takes less computational power while still achieving a pretty good performance. Even though I had to tune the hyperparameters for random forest, it could barely beat the performance of a basic linear regression model. Since linear regression predicts almost with the same performance as a random forest model, this demonstrates that there could be weak correlation between the input and output. In my linear regression model, the coefficients for age, height, weight, and category was 0.11425837, -0.12288522, 0.71516225, and 0.17732374. The largest coefficient was weight. It can be seen that the heavier someone is the larger size, but that statement cannot be strongly supported.

I would say that my regression models did fairly well, however, they can be improved upon with more features. Many different dresses, gowns, and sheaths have different styles for a variety of body types. I should into account other features like bust-size, hip-size, and body-type to refine my predictions even more. In some cases, more features can be great, but it will increase the chances of overfitting the training data and poorly performing on unseen data. On the bright side, with more features, we can create more complicated models to precisely calculate the perfect size.

6 Conclusion

In the study, I propose a supervised framework for the clothing size recommendation problem. Specifically, I applied three regression models and compared their performance by taking the mean squared error of their predictions. The findings display that a simpler model may be preferred with basic demographic features. Quantitative and qualitative results show the effectiveness of the proposed models.

References

Lin, Cheng et al. (Mar. 2020). "Size Prediction for Online Clothing Shopping with Heterogeneous Information". In: 1. URL: https://ieeexplore.ieee.org/abstract/document/9023656/references#references.

Misra, Rishabh, Mengting Wan, and Julian McAuley (Oct. 2018). "Decomposing Fit Semantics for Product Size Recommendation in Metric Spaces". In: URL: https://cseweb.ucsd.edu/~jmcauley/pdfs/recsys18e.pdf.

Sembium, Vivek et al. (Aug. 2017). "Recommending Product Sizes to Customers". In: 1. URL: https://cseweb.ucsd.edu/classes/fa17/cse291-b/reading/p243-sembium.pdf.

Sembium, Vivek et al. (Apr. 2018). "Bayesian Models for Product Size Recommendations". In: 1. URL: https://doi.org/10.1145/3178876.3186149.

Vleugels, J. et al. (Aug. 2022). "Predicting User's Measurements without Manual Measuring: A Case on Sports Garment Applications". In: URL: https://www.mdpi.com/2076-3417/12/19/10158.