Shop your Right Size: A System for Recommending Sizes for Fashion products

G. Mohammed Abdulla[*]
gmohamedabdulla@gmail.com

Shreya Singh
Myntra Designs, India
shreya.singh1@myntra.com

Sumit Borar
Myntra Designs, India
sumit.borar@myntra.com

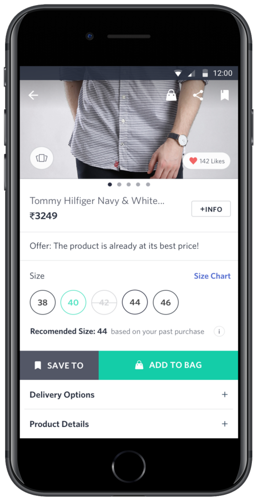
ABSTRACT

Size selection is a critical step while purchasing fashion products. Unlike offline, in online fashion shopping, customers don't have the luxury of trying a product and have to rely on the product images and size charts to select a product that fits well. As a result of this gap, online shopping yields a large percentage of returns due to size and fit. Hence providing size recommendation for customers enhances their buying experience and also reduces operational costs incurred during exchanges and returns. In this paper, we present a robust personalized size recommendation system which predicts the most appropriate size for users based on their order history and product data. We embed both users and products in a size and fit space using skip-gram based Word2Vec model and employ GBM classifier to predict the fit likelihood. We describe the architecture of the system and challenges we encountered while developing it. Further we also analyze the performance of our system through extensive offline and online testing, compare our technique with another state-of-art technique and share our findings.

KEYWORDS

Personalization, Size Prediction, Word2Vec, Doc2Vec

ACM Reference Format:
G. Mohammed Abdulla, Shreya Singh, and Sumit Borar. 2019. Shop your Right Size: A System for Recommending Sizes for Fashion products. In *Companion Proceedings of the 2019 World Wide Web Conference (WWW '19 Companion), May 13–17, 2019, San Francisco, CA, USA.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3308560.3316599



Tommy Hilfiger Navy & White...
₹3249

Offer: The product is already at its best price!

Recomended Size: 44 based on your past purchase

Figure 1: Size Recommendation on our platform.

1 INTRODUCTION

The online fashion industry is growing at an astounding pace. Reliable statistics [11] show that the worldwide revenue is expected to rise from $481.2 billion in 2018 to $712.9 billion by 2022. Fashion products like apparels and shoes are available in various sizes and size selection becomes a critical part while purchasing them. While purchasing offline, consumers typically try the product before making a purchase whereas during online shopping, users have to rely on size charts or images of products to make the size decision. Size charts require customers to remember their body measurements and compare them with product dimensions. Moreover the fashion industry lacks standardization in terms of sizing [5] and the attributes associated with fashion products are highly subjective. Moreover, even for the same brand, different product lines and various fits (Slim, Regular etc) make choosing size a tricky process. To make the online purchase convenient, online platforms typically provide free returns. This results in higher return rates due to size and fit issues in online e-commerce and adds significant operational costs. In this paper, we propose a novel approach to recommend sizes using latent features and gradient boosting classifier(GBC). Figure 1 illustrates the size recommendation displayed on Myntra's mobile app. Every product in the catalog has multiple sizes and a combination of the product and its size is referred as SKU. The GBM classifier takes as input a combination of user and SKU vectors and returns the probability of fit. Further, the SKU vector is a combination of observable features like size, occasion, fit , material etc. and latent features learnt from purchase history using Word2Vec. The latent features transform the users and products into a size and fit space. This space has an inherent property of bringing similar-sized products closer to each other. Moreover, users whose size preferences are similar are also closer in this space. Our experiments demonstrate that models built with both observable and latent features perform better than models built solely on either observable or latent features.

Myntra is India's largest fashion e-commerce portal serving millions of customers daily. We operate solely in the Indian retail

[*] Work done while author was at Myntra

WWW '19 Companion, May 13–17, 2019, San Francisco, CA, USA

ACM ISBN 978-1-4503-6675-5/19/05.
https://doi.org/10.1145/3308560.3316599

market and currently have support for English language on our platform. In order to provide a better shopping experience it is imperative that we help our customers choose the right size for any given product. In our journey to develop this system, we came across several challenges specific to our domain:

- Users mostly tend to purchase from similar price ranges or similar brands. Hence, Word2Vec fails to capture similarity across dissimilar price ranges or brands, resulting in local clusters in the size and fit space.
- Some articles like Shirts, T-shirts, Dresses are bought very often while others like Jackets, Sweatshirts are not purchased frequently. This poses a challenge to recommend sizes with high confidence in less often purchased categories.
- Users in our platform do not just buy for themselves, but use the same account to buy for many people. It is not trivial to make size recommendations for such users.

This paper is organized as follows. Section 3 outlines the size recommendation algorithm as a classification problem. We describe various challenges specific to our domain and our solutions in Section 4. We present a comprehensive architecture of the online size recommendation system in Section 5 followed by results and analysis in Section 6.

2 RELATED WORK

There is substantial research on personalized product recommendation for users in fashion e-commerce. Recommendations based on users' past interactions and affinities have been studied extensively [1, 2, 7]. All these recommendation systems try to model user product preferences and not size preferences. A recommendation system that recommends product sizes like {Small, Fit, Large} to customers in Amazon is presented in [12]. In this work, the authors have defined Hinge Loss and Logistic loss variants to compute true sizes for customer-product pairs by minimizing the loss function which is based on the difference between the true sizes of the customers (derived from order history) and the true sizes of the products (derived from the product dimensions). This approach is based only on size measurements and does not take into account attributes like fit, material, brand while making recommendations. Apart from these, in the textile and garments industry, several methods to find the appropriate size and fit preferences are proposed which rely on the 3D modelling of body shapes [4]. These approaches are based heavily on deriving body shapes from a database of body shape metrics which are curated manually or deriving body shapes from images. There have also been attempts to model user size preferences in industry by the likes of True Fit and Fit Analytics. However, their approach requires users to provide body measurements explicitly via surveys or questionnaires.

3 METHODOLOGY

Our method to solve the size recommendation problem is to model it as a binary classification problem with positive class consisting of SKUs retained by the users and negative class consisting of SKUs returned/exchanged by the users due to size and fit issues. Post training, we use Gradient Boosting model [8] to determine the fit probabilities. The user-SKU pair which gives us the highest confidence is the recommended size. The approach could be divided into three main components: generation of observable feature vectors, generation of latent feature vectors, and using the ensemble method for classification. We have used implicit matrix factorization as the baseline approach.

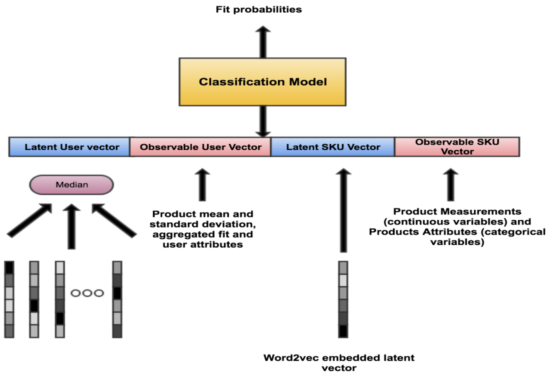

Fit probabilities
Classification Model
Latent User vector
Observable User Vector
Latent SKU Vector
Observable SKU Vector
Median
Product mean and standard deviation, aggregated fit and user attributes
Product Measurements (continuous variables) and Products Attributes (categorical variables)
Word2vec embedded latent vector

Figure 2: Classification model architecture showing the training vector as a combination of User and SKU vectors

3.1 Generation of Observable feature vector

The observable feature vector for a SKU consists of what we "observe" in any product. For our use case, the observable features consist of measurement attributes(like chest, bust, waist dimensions), material and colour. Measurement attributes are continuous values and are not uniform across different categories. For instance, in Men Topwear category, the measurements are chest, shoulder and length; whereas in Women Topwear category it is bust, shoulder and length. Other product attributes are categorical values and are consistent across categories. A combination of these gives the SKU observable features. The user observable feature vector is formed by aggregating all the observable feature vectors of the SKUs bought by a user. To this, we append additional information like mean and standard deviation of the measurement attributes which will capture the user's purchased size distribution.

3.2 Generation of Latent feature vector

Latent feature vectors incorporate the "hidden" representations of user purchase traits. We use skip-gram based Word2Vec [9] model to derive the latent feature vectors. The input to these models are the set of SKUs bought by the user. The SKUs which have been returned or exchanged are removed while generating latent feature vectors. They are however used as negative samples while training the model. These SKUs are not used directly, but are first converted to a combination of Brand, Size, Fit and Usage attribute for that product. For example a SKU with brand "Nike", size "L", fit "Slim" and usage "Sports" can be converted into a word like "Nike-Sports-Slim-L". We have analyzed that a word consisting of all these attributes for a SKU representation gives us better results than only including size and brand or size and occasion. These are necessary features as there are several product lines and usage attributes in a brand which are pertinent to a product. Hence, training a Word2Vec model with these features would help us govern the size profile of a user. The SKUs can be considered as words and all the SKUs purchased by a user can be considered as a document. With these words and documents as inputs we train the Word2Vec network, the final activation layer of every SKU is obtained as an "off-the shelf" feature which will serve as the latent feature vector.

Formally, the user purchase data can be represented as a sparse matrix W where each row represents a user and each column represents the SKUs bought by the user. We replace each SKU by the word as explained above. Each row i of matrix W would be a document of word representations of the SKUs bought by the user. We train the Word2Vec network with matrix W where W_i represents all the purchases of a user i sorted by date which can be represented by the sequence $W_{i1}, W_{i2}, W_{i3}, ..W_{in}$, here each $W_{ij} \in C$ where C is the entire platform catalog for a particular article type and W_{ij} is the word representation of the SKU. The objective of the skip gram model is to maximize the log probability.

$$\frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{-q\leq k\leq q, k\neq 0}\log p(w_{i,j} \mid w_{i,j+k}) \tag{1}$$

where q is the hyperparameter denoting length of the purchase window. Larger q results in SKUs spanning over wide range of purchases to be considered as having same size and fit. The formulation of $p(w_j \mid w_{j+k})$ is given using Softmax function:

$$p(w_j \mid w_{j+k}) = \frac{e^{u_j, v_{j+k}}}{\sum_{k\in W} e^{u_k, v_k}} \tag{2}$$

where u and v are the input and output one hot encoded vector representation of w. After we have trained the neural network, we compute the activation of the hidden layer for each SKU $c_p \in C$ and form a latent feature vector representation f_p. Post training, the activation layer is scraped off and used as the latent vector representation for each SKU. The latent user vector can be computed by taking the mean of the latent vectors of SKUs bought by the user.

3.3 Recommendation as Gradient Boosting classification

We model the size recommendation system as a binary classification problem. For this purpose; we use Gradient Boosting Classifier for training as it will provide non-linearity and reduce over fitting. In our binary classification setting, positive class for recommendation is the set of user-SKU pairs purchased and retained by users and negative class is the set of user-SKU pairs returned or exchanged by users. When any product is returned or exchanged on our platform; the user is required to select a reason from a pre-defined set of reasons like Size/Fit issues, wrong product, late delivery etc. All the return and exchange reasons are logged and we make use of these reasons to generate negative class samples. Post training, we rank the fit probabilities of all SKU-user pairs for a given user and product (having multiple SKUs) and deem the SKU with highest fit probability as the recommended SKU(size). In our GBC model, the training vector is a concatenation of user and SKU vectors. Figure 2 shows our baseline architecture and outlines the structure of the training vector as a combination of user and SKU vectors. Further, the user and SKU vectors are themselves built of latent and observable parts as illustrated in Figure 2. In our current system, we require one SKU bought and retained by the user to make reliable recommendations for him/her.

3.4 Implicit Matrix Factorization

To compare our model we have used Implicit matrix factorization [6]. The data can be represented as a matrix with rows as users, columns as products and each cell in this matrix is the number of times given user has purchased the given product. As directly using every product on our platform resulted in a sparse matrix with sparsity greater than 99.9%; products were replaced with words as described above (example: "Nike-Sports-Slim-L"). This resulted in sparsity less than 99.9% and this sparsity is considered as an empirical standard in collaborative filtering problems. Validation data was used to tune various parameters of the matrix factorization model. The result of this method is compared with our classification approach in Section 5.6.

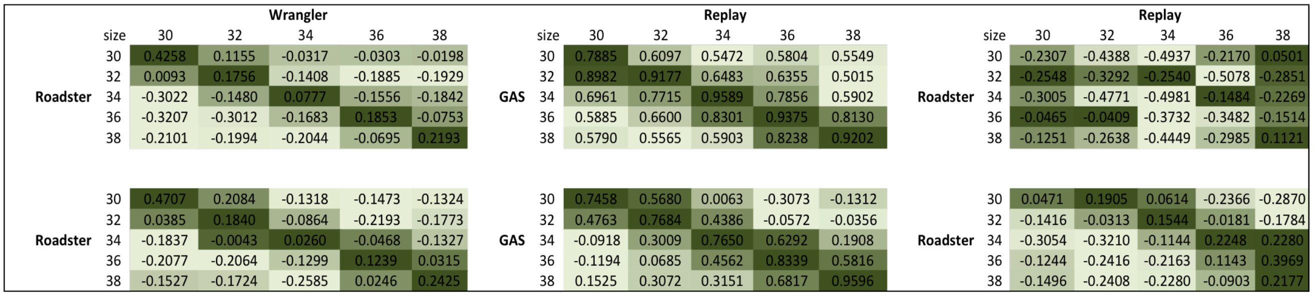
4 CHALLENGES

In this section we address various challenges we faced while building the size recommendation algorithm. Quantitatively we want to improve two metrics - precision and coverage. Precision in our context refers to how accurately we can make size predictions and Coverage is the number of users for whom recommendations can be made with said precision. The following challenges and their corresponding solutions improve either one or both of the performance metrics.

4.1 Local Clustering in Embedding Space

On our platform customers can be largely segmented into mass, mass premium, premium, bridge to luxury and luxury segments. These segments primarily indicate the price bands and brands which certain group of users would be interested in. For instance, in case of Jeans, brands like GAS, Replay, Scotch and Soda, where an average product is priced at 10000 INR, are considered to be luxury brands, while brands like Levis, Roadster, Wrangler and Tommy which are priced around 3500 INR are segmented into mass premium brands. There has been previous work [3] which show that users tend to purchase in their respective segments and rarely move across them while making future purchases. This has an unfortunate effect of less co-purchase history across brands belonging to different segments. Figure 3 illustrates this behaviour by aggregating cosine similarity metric between brands belonging to different segments. Techniques like Word2Vec and Collaborative filtering fail in such cases where user interactions are sparse which is evident in Row 1 of Figure 3. Such behaviour occurs due to the formation of local clusters in product embeddings.

For illustration, in Figure 3, we have considered embedding vectors generated using the method mentioned in above section for four different brands of jeans - Roadster, Wrangler, Replay and GAS. Among these brands Roadster and Wrangler fall into mass premium segment while Replay and GAS are in luxury segment. The first row shows cosine similarity between embeddings for various sizes of these brands. It can be clearly seen that brands from same segment have linear relation between various sizes, but when we consider brands across segments - the similarity values are random and represent misinformation in the embedding space. Further,



				Wrangler							Replay							Replay		
	size	30	32	34	36	38		size	30	32	34	36	38		size	30	32	34	36	38
	30	0.4258	0.1155	-0.0317	-0.0303	-0.0198		30	0.7885	0.6097	0.5472	0.5804	0.5549		30	-0.2307	-0.4388	-0.4937	-0.2170	0.0501
	32	0.0093	0.1756	-0.1408	-0.1885	-0.1929		32	0.8982	0.9177	0.6483	0.6355	0.5015		32	-0.2548	-0.3292	-0.2540	-0.5078	-0.2851
Roadster	34	-0.3022	-0.1480	0.0777	-0.1556	-0.1842	GAS	34	0.6961	0.7715	0.9589	0.7856	0.5902	Roadster	34	-0.3005	-0.4771	-0.4981	-0.1484	-0.2269
	36	-0.3207	-0.3012	-0.1683	0.1853	-0.0753		36	0.5885	0.6600	0.8301	0.9375	0.8130		36	-0.0465	-0.0409	-0.3732	-0.3482	-0.1514
	38	-0.2101	-0.1994	-0.2044	-0.0695	0.2193		38	0.5790	0.5565	0.5903	0.8238	0.9202		38	-0.1251	-0.2638	-0.4449	-0.2985	0.1121
	30	0.4707	0.2084	-0.1318	-0.1473	-0.1324		30	0.7458	0.5680	0.0063	-0.3073	-0.1312		30	0.0471	0.1905	0.0614	-0.2366	-0.2870
	32	0.0385	0.1840	-0.0864	-0.2193	-0.1773		32	0.4763	0.7684	0.4386	-0.0572	-0.0356		32	-0.1416	-0.0313	0.1544	-0.0181	-0.1784
Roadster	34	-0.1837	-0.0043	0.0260	-0.0468	-0.1327	GAS	34	-0.0918	0.3009	0.7650	0.6292	0.1908	Roadster	34	-0.3054	-0.3210	-0.1144	0.2248	0.2280
	36	-0.2077	-0.2064	-0.1299	0.1239	0.0315		36	-0.1194	0.0685	0.4562	0.8339	0.5816		36	-0.1244	-0.2416	-0.2163	0.1143	0.3969
	38	-0.1527	-0.1724	-0.2585	0.0246	0.2425		38	0.1525	0.3072	0.3151	0.6817	0.9596		38	-0.1496	-0.2408	-0.2280	-0.0903	0.2177

Figure 3: Cosine similarity of embedding vectors across different brand-size combinations. Row 1 shows the similarity values in normal Word2Vec method and row 2 shows the similarity values of embeddings which are generated from sampled data.

we aggregate these product embeddings to form error-prone user embeddings. This is an undesired behaviour and results in wrong recommendations. To tackle this problem we have come up with two solutions - improving the aggregation logic of user vectors and improving the underlying product vector generation logic.

4.1.1 Improve User Vector Aggregation. In the current formulation where product vectors form local clusters, there exist three different scenarios while making the final size recommendation. First scenario occurs when a user has purchased from a subset of similar brands and the product for which recommendation has to be made is a brand similar to user's past purchases. Second scenario occurs when a user's past purchases are from a subset of similar brands but the product for which recommendation has to be made is not similar to user's past purchases. Third scenario occurs when a user's past purchases are from various segments (in this case the user embedding will be incorrect) and the product to be recommended for could be any brand in the catalog. The first and second scenarios will not result in wrong recommendation because the final Gradient boosted classifier will learn to recommend correctly in these situations, but, in the third scenario, the mean of segmented vectors falls in a random location in the size and fit space resulting in incorrect recommendations.

To make our model robust for the third scenario, instead of taking mean as an aggregation metric, we experimented with other aggregation techniques. While median is less susceptible to outliers than mean, in our case taking a median of embeddings is equivalent to choosing a dominant subset which results in loss of valuable information. We also experimented by concatenating all the product vectors purchased by the user to generate a user embedding. This does not lose out on information but adds complexity to the model and also makes it sensitive to the order in which products are concatenated. Finally, we generated user embedding along with product embedding using Doc2vec[10]. In our setting, documents are all the purchases made by a user and words are SKUs represented by their attributes. Unlike Word2Vec where we only learn representation for words(SKUs), in Doc2Vec we also learn representation for the document(user in our case). In the Results section, we compare the three aggregation logic and illustrate that Doc2vec vectors are the best.

4.1.2 Improve Product Vector. Our other approach to solve the local clustering problem is to improve the underlying product vector generation logic. As co-purchase history is sparse, we came up with a method to sample from the given purchase data resulting in global clusters. The procedure for sampling is given in Algorithm 1. We create a co-purchase Graph $\mathcal{G}$ with nodes as brands and edge weights as number of times these two brands were purchased together. We also create a map $\mathcal{P}$, which has key as pair of brands and values as a list of all the users who have purchased these two brands. The sampling algorithm looks at every brand(node) in the graph and computes a median edge weight from all the outgoing edges from the current brand. It then looks at each edge, if the weight of outgoing edge is less than the median, it over samples users from $\mathcal{P}$. The output of the algorithm is a list of users $\mathcal{U}$ and for each user in this list we replicate their previous orders. This results in Word2Vec learning relation between brands which were not obvious earlier. Row 2 of Figure 3 illustrates cosine similarity computed using embeddings learned from combination of purchase history and sampled data $\mathcal{U}$.

Algorithm 1 Sampling Algorithm

```
Require: Graph $\mathcal{G}\{V, E\}$, Dictionary $\mathcal{P}$
  Initialization $\mathcal{U} = list$ {initialize empty list}
  for all $a \in \mathcal{G}.V$ do
    $m = median(\mathcal{G}, a)$ {compute median value of weights from node a}
    for all $b \in \mathcal{G}.V$ do
      {G(a,b) - return weight between node a and b}
      if $a \neq b$ & $\mathcal{G}(a, b) \leq m$ then
        $v$ = sample $(m - \mathcal{G}(a, b))$ users from $\mathcal{P}(a, b)$
        $\mathcal{U}v$
      end if
    end for
  end for
```

4.2 Lack of Purchase History

In Section 3 we mentioned that we use purchase history of an article (like Shirts, T-shirts, Dresses) to create Word2Vec embeddings of products. These embeddings are feasible in articles which are sold more often in the platform.

However, many articles like Sweaters and Sweatshirts are purchased less frequently and we do not have enough co-purchase history to create quality embeddings. Hence, to enrich embeddings for these articles, we use information of other similar articles. All the articles in our catalog belong to certain high-level categories: namely Men's Topwear , Women's Topwear, Men's Bottomwear, Women's Bottomwear and Footwear, and we train a Word2Vec model on category purchase history rather than article purchase history resulting in rich embeddings for products in all articles. Although using category for learning product embedddings is superior in theory, implementation of this method has a few challenges. Users can make purchase for multiple people using the same account or may make purchase for different articles for different people. We want to identify a dominant persona from all the different personas that exist for an account. Identifying dominant persona can be formulated as an optimization problem as below.

Let A be the set of all orders of a user then dominant user set is given by

$$\begin{aligned} & |A'|; A' \in \mathcal{P}(A) \\ \text{s.t.}\quad & abs(A_i - A_j) \leq \lambda \\ & \forall A_i, A_j \in A', i \neq j \end{aligned} \tag{3}$$

where λ is the threshold set for optimization and $\mathcal{P}(A)$ is the power set of A.

A is the set of sizes purchased by a user in a given category. For Men's Topwear, set A would contain chest measurements in inches of all the purchased products. The threshold λ is a tuneable parameter and is set with domain knowledge. Identification of dominant profile has to run for millions of users and we present a pseudo code Algorithm 2 which runs in log-linear time complexity. Once the dominant profile and its orders are identified, these orders are used in creating Word2Vec embeddings.

Algorithm 2 Optimization Algorithm

```
Require: List $A$, threshold $\lambda$
Initialization $i = 0$ ,$j = 1$, $maxSet = list$,$curSet = list$ {initialize empty lists}
$A = sort(A)$
while $i < len(A) - 1$ & $j < len(A)$ do

    if $A_j - A_i \leq \lambda$ then
        $curSetA_j$
        $j1$
    else
        $curSetA_i$
        $i1$
    end if
    if $len(maxset) < len(curSet)$ then
        $maxSet = curSet$
    end if
end while
```

4.3 Short Lifetime of Users and Multi-persona

About 65% of users on our platform purchase for more than one person using the same account. The current formulation of size recommendation can only make recommendations for the remaining 35% of the users. To make recommendations for multi-persona users we have developed a sub-system where users can create multiple profiles in the app. The user can tag the bought products to these profiles post placing an order. We aim to target multi-persona users through this approach. Each of these profiles can now be considered as a single-persona user to get profile-level size recommendation. It is observed that the average lifetime of users on our platform is short. We want to make size recommendation available to users during early stage after adoption and we leverage profile-level size recommendation to achieve this task. In the later section we explain in detail how profile-level recommendation system is implemented.

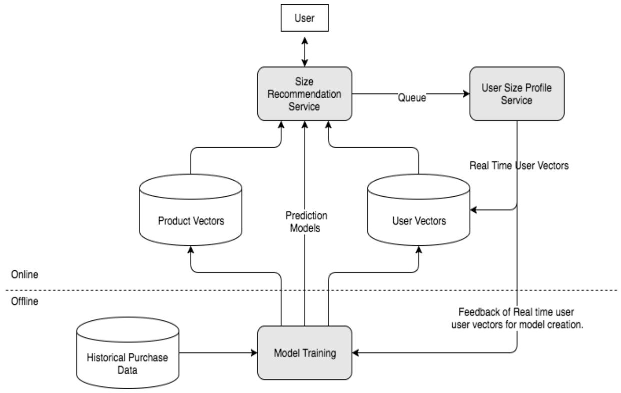


Figure 4: Simplified System Architecture

5 SYSTEM ARCHITECTURE

In this section we outline the components of size recommendation system using the the block diagram in Figure 4. The system consists of an offline training system and an online serving system.

In the offline training system, we use historical purchase data and content data of products to create embeddings. These embedddings are then used by the gradient boosted classifier model to create the prediction models. The prediction models, user vectors and product vectors are fed into databases for the online serving systems.

In the online system, size recommendation service is deployed across multiple servers coupled with a load balancer. Whenever a user opens a product page, the load balancer redirects the request to one of the size recommendation service systems. The size recommendation system fetches the user and product vectors from the database, computes the best fitting size and returns the same to the user. Our current system can handle 700K concurrent requests per minute. The online system is designed such that it can be horizontally scaled whenever traffic on the platform increases.

65% of users on our platform have multiple personas and buy for more than one user using the same account. To tackle this, we have built a profile level recommendation system where users can tag their current purchase to an existing profile or create a new profile. A user is allowed to have multiple profiles but a product can be tagged only to one profile. This gives us explicit feedback for whom the product was purchased. Each of these profiles are considered as different users by our model. The profile tagging system creates user vectors (profile vectors) in real time which are populated in the user vector database. Using the same size recommendation service we can now make size recommendations for new profile

Table 1: Precision scores for various articles

Article	Precision
Men Shirts	84.17%
Men Tshirts	84.33%
Women Dresses	82.07%
Women Kurtas	80.83%
Women Sweaters	93.8%
Men Trousers	95.25%

users. More than a million profiles were created in less than a week of the launch of profile tagging feature and essentially doubled the coverage of users. The profile vectors are fed back into the offline model creation pipeline to make the size recommendation system robust.

6 EXPERIMENTS AND RESULTS

In this section we provide data analysis and results of various experiments that were carried on our size recommendation system. We present both offline test results which were used to validate and test our method on historical data and online AB test results from our production systems.

6.1 Dataset

For the offline experiments we have used historical purchase data. As the goal of our model is to make future predictions, we have split the data by time and used 27 months' data as train data, 2 months' data as test data and 1 month's data for cross validation of model parameters. Our system builds a model for every article (eg : Men Shirts, Women Dresses, Men Trousers etc) on our platform. From here on we will consider the results of experiments carried on 'Men Jeans'. The total number of distinct styles for Men Jeans hosted on our platform is in the order of hundred thousands whereas the total transactions data in in millions. To protect our proprietary information, we do not report the actual product and transaction counts.

6.2 Data Analysis

To build our size recommendation models, we have collected product data and user transaction data to give us valuable insights about user behavior and buying trends. We have plotted the catalog size of product and transaction categories in Figures 5 and 6. Here, we have scaled the main dimension of products to US size convention of 4 to 22. We can see from Figure 5 that most of the product sizes of all four apparel categories lie within 10 to 16 sizes with women apparel categories (Women Sweaters, Women Sweatshirts) having more catalog inventory for smaller sizes in this range (more size 10 and size 12) than men apparel categories (Men Sweaters and Men Kurtas). This trend can also be seen in the transactions data in Figure 6. This can be attributed to the notion that most women are structurally smaller than men as the buying trends signify that popular sizes for women generally run a size or two smaller than men. This is reflected in the product inventory too as appropriate sizes have to be stocked for sale.

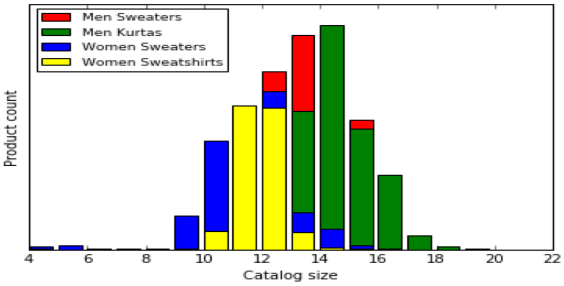


Figure 5: Product count vs catalog size

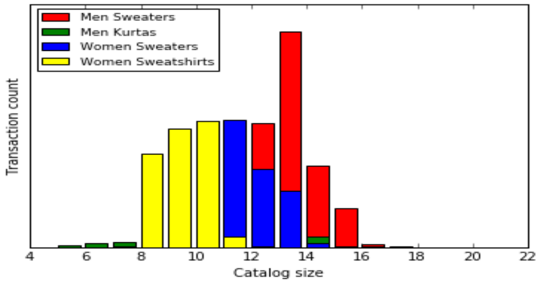


Figure 6: Transaction count vs catalog size

6.3 Model Results

We experimented with three different classifier models with various set of features. The features comprised of concatenated user and product vectors. For the first experiment we only used observable features in the classification task. Second experiment was performed using the latent features. For the third experiment we used both observable features and latent features for creating the corresponding product and user vectors. For comparison of results of the three experiments we have used Area under the curve in ROC. Figure 7 shows results for the three models. It can be seen from the figure that a model with a blend of observable and latent features is the most superior model. Table 1 shows the precision values for some articles and it is noticed that our approach generalizes well to different articles.

6.4 Local Cluster Results

In the previous sections we presented the problem of local clustering due to insufficiency of co-purchase data across various brands. Figure 8 compares different aggregation techniques; it can be seen that the median aggregation method performs better than mean aggregation method as the median metric is less susceptible to outliers. The concatenation aggregation method performs worse than the mean method as concatenated vectors are not robust. In the Doc2vec formulation, user vectors were created during training along with product vectors. This ensures that user vectors are a good representation of users in the size and fit space compared to any other aggregation method. The classification algorithm trained with Doc2Vec aggregation has the maximum AUC of 0.86.

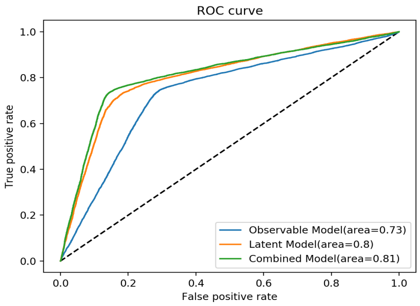


Figure 7: Comparison of classification models using ROC. It can be seen that the combined model with observable features and latent features is better than the individual models.

Table 2: Reduction in return rates for different articles

Article	Return rate reduction
Men Tshirts	6.4%
Men Jeans	8.4%
Men Trousers	4.8%
Men Sweatshirts	9.34%
Women Kurtas	1.08%

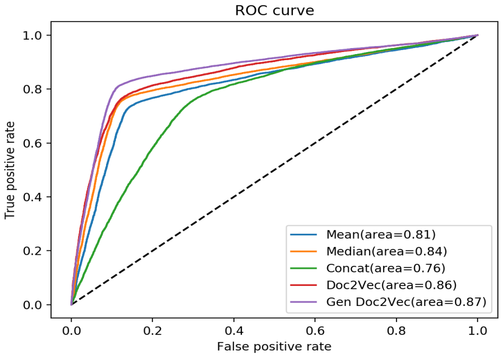


Figure 8: Comparison of various aggregation techniques for computing user vectors from product vectors. Doc2vec user vectors are better than all the other aggregation techniques. The purple curve shows Doc2vec user vector results with generated samples.

6.5 Online Results

Recommending correct sizes to users would result in reduction in returns due to size and fit issues. We set up an online A/B test to track the return reduction due to size and fit. In our platform we have a 30 day return policy which means users can return products purchased anytime in the next 30 days from the date of purchase. We conducted the test during a two-month window, wherein we considered the purchases made in the first month whereas the second month was a cool off period where we track the returns for these purchases. A/B test showed similar return rates for control and test sets before the start of the experiment. Members in the control set did not get any size recommendation while members from test set were shown recommendation from our system. Table 2 shows reduction in return rate for some articles for the month of December 2018.

Table 3: Test precision scores comparison for GBM approach and Matrix Factorization based approach

Article	GBM Test Precision	MF Test Precision
Men Shirts	84.17%	59.18%
Men Tshirts	84.33%	60.46%
Women Dresses	82.07%	60.73%
Women Kurtas	80.83%	58.53%
Women Sweaters	93.8%	61.36%
Men Trousers	95.25%	66.07 %

6.6 Implicit Matrix Factorization Results

We have used Implicit Matrix Factorization method outlined in Section 3.4 as the baseline method to compare our results. We ran both the methods on the test and train dataset for some of our article types and have tabulated the results in Table 3. Our approach outperforms the baseline approach by a huge margin for the test set. It was also observed that the baseline approach suffered from over fitting as the train set results were much higher than the test set results.

7 CONCLUSION

In this paper, we have described a size recommendation system which is used on our platform to guide users choose correct sizes while making an online purchase. We have introduced a method of creating latent features from purchase data and combining them with observable features to form a robust classification model for size recommendation. Further, we discussed the challenges which are unique to this domain like formation of local clusters in size and fit space and presented our approaches to overcome them. Due to inherent behaviour of multiple users purchasing from same account, we introduced a method of tagging orders to user-profiles which enabled us to make recommendation to multi-persona accounts with the same construct. To the best of our knowledge, this is the first attempt in the industry to scale size recommendation to multi-persona accounts. Although, we have demonstrated our work on fashion E-commerce data, it can be extended to other domains where we have sparse user and product signals. For our future work, we want to expand our work to make size recommendation available for users who have not made even a single purchase on our platform. We also intend to leverage the clothing contours of product images which might give valuable insights into product recommendations.

ACKNOWLEDGMENTS

The authors would like to thank Karthik M, Dheeraj Kumar, Sagar Grover, Srijit Sengupta and Gaurav Kumar for their contributions

in building data pipelines and providing valuable product insights in algorithm design, implementation and evaluation.

REFERENCES

[1] Gediminas Adomavicius and Alexander Tuzhilin. 2005. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE transactions on knowledge and data engineering* 17, 6 (2005), 734–749.

[2] Sagar Arora and Deepak Warrier. [n. d.]. Decoding Fashion Contexts Using Word Embeddings. ([n. d.]).

[3] Sandeep Bhanot and Srini R Srinivasan. 2013. A study of the Indian apparel market and the consumer purchase behaviour of apparel among management students in Mumbai and Navi Mumbai. *Navi Mumbai* (2013).

[4] AP Chan, J Fan, and WM Yu. 2005. Prediction of men's shirt pattern based on 3D body measurements. *International Journal of Clothing Science and Technology* 17, 2 (2005), 100–108.

[5] Chih-Hung Hsu. 2009. Data mining to improve industrial standards and enhance production and marketing: An empirical study in apparel industry. *Expert Systems with Applications* 36, 3 (2009), 4185–4191.

[6] Yifan Hu, Yehuda Koren, and Chris Volinsky. 2008. Collaborative filtering for implicit feedback datasets. In *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*. Ieee, 263–272.

[7] Yang Hu, Xi Yi, and Larry S Davis. 2015. Collaborative fashion recommendation: a functional tensor factorization approach. In *Proceedings of the 23rd ACM international conference on Multimedia*. ACM, 129–138.

[8] Llew Mason, Jonathan Baxter, Peter L Bartlett, and Marcus R Frean. 2000. Boosting algorithms as gradient descent. In *Advances in neural information processing systems*. 512–518.

[9] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013).

[10] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. 3111–3119.

[11] Aaron Orendorff. 2019. The State of the Ecommerce Fashion Industry: Statistics, Trends & Strategy. Retrieved Jan 10, 2019 from https://www.shopify.com/enterprise/ecommerce-fashion-industry

[12] Vivek Sembium, Rajeev Rastogi, Atul Saroop, and Srujana Merugu. 2017. Recommending Product Sizes to Customers. In *Proceedings of the Eleventh ACM Conference on Recommender Systems*. ACM, 243–250.