Deep Learning based approach for personalized size-fit recommendation

November 2022

1 Dataset

For this assignment we chose to use modcloth fitness of clothing dataset [3], the dataset is collected from Modcloth, an online retail company primarily for women's clothing. The dataset captures the product user interaction as well as characteristics of both user and clothes. Dataset also includes the user review and experience of buying these clothes.

1.1 Dataset Description

The dataset contains 47958 unique user and 1378 unique product and 82790 records. Each record in the dataset is an interaction between an clothing product and a user. Features including clothing characteristics (type, size, etc) and user characteristics (height, weight, etc), and interactions like user's rating and review, a complete list of features are listed below.

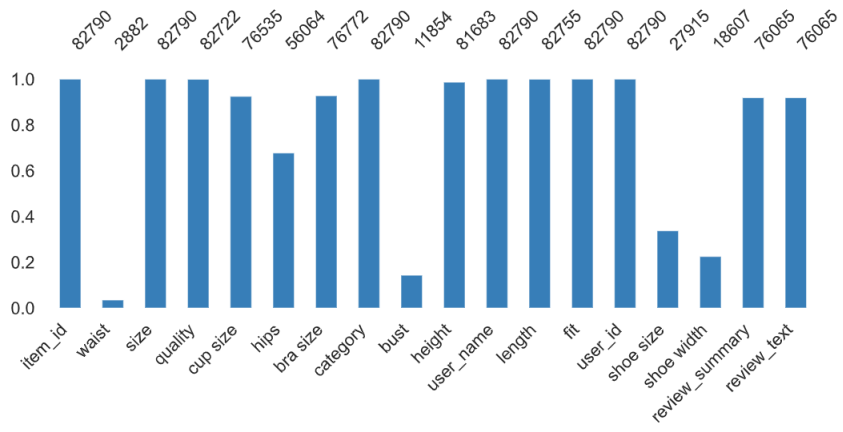
- item id: Id of the product. (Categorical identifier)
- waist: Waist size of user. (Numerical)
- size: Clothing size. (Numerical)
- quality: User rating to the cloth. (Numerical, 1-5)
- cup size: Categorical Woman cup size. (Categorical, 12 unique value)
- hips: User heap size. (Numerical)
- bra size: User bra size. (Numerical)
- category: Clothing category. (Categorical, 7 unique values)
- bust: User bust size. (Numerical)
- height: User height. (Numerical)
- user name: User name. (Categorical identifier)

- length: Is length of the clothes compatible with the user. (Categorical, too long, too short, just right)
- fit: How the clothing fits the user overall. (Categorical, fit, too large, too small)
- user id: User id on the website. (Categorical identifier)
- shoe size: User shoe size. (Numerical)
- shoe width: User shoe width. (Categorical, samll avg large)
- review summary: One sentence review summary. (Text)
- review text: Detailed review of the clothes by the user. (Text)

In those features, item id, category, are features associated with product. Fit, length, review text, review summary and quality are associated with user product interaction. The rest of features are associated with the user.

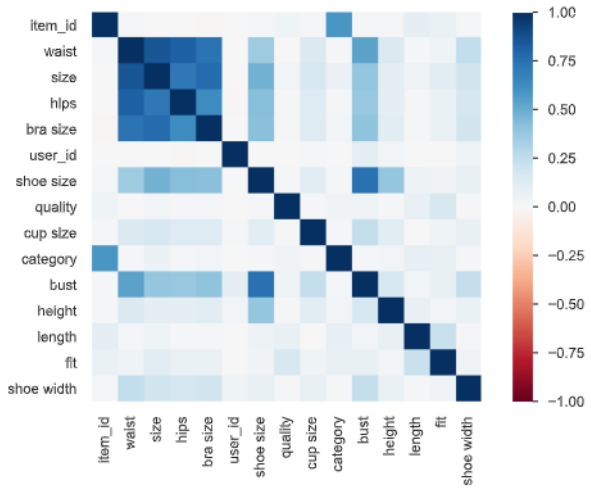
1.2 Exploratory Data Analysis

In the section we conduct EDA on the Modcloth dataset and identify some challenges with respect to any down stream predictive tasks.

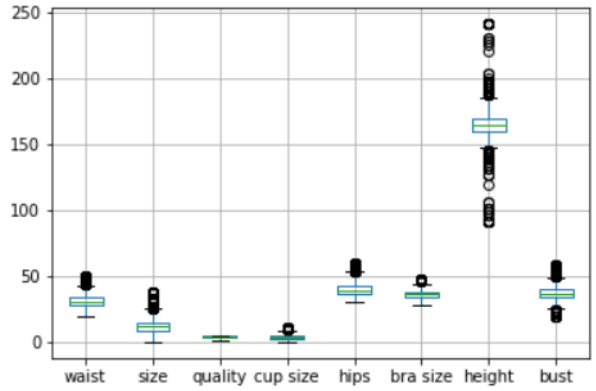


A simple visualization of nullity by column.

First we analyze the missing values of in the dataset, we find that, for some features, the majority of the values are missing or unavailable, for example for waist size, 96.5% of the data is missing, bust size has 85.7% of the data is missing and shoe size/width have large portions of missing data as well.



Next we analyze the correlations between features, we find that size related features to the user are highly correlated, that means some features offer somewhat redundant information thus some features with many missing values can probable be ignored.



Lastly, we analyze the range and scales of numeric values, we found that features have different range and scale are vastly different (for example, height and quality.) For any predictive task, feature scaling, normalization or standardization is likely required.

Based on the above EDA, we recognize there are following several challenges of doing predictive task on this dataset:

- The dataset has many missing values which can cause features being ineffective or even damage the performance.

- There are no features that strongly correlate with user-produce interactions, so feature engineering and is important to extract such information from raw data.
- The range / scale of features is different which can cause strong bias towards features with larger numeric value.
- Lastly, text based features like review and review summary are difficult to use in traditional predictive models so they need to be pre-processed to any downstream model.

2 Task

In this section, we describe the identified predictive task based, our pre-processing step and feature engineering in detail. In addition, we also describe the baseline that we will be using.

2.1 Predictive Task

Given the physical features of user and the the clothing product, the task of interest is to determine the product user interaction. We attempt to determine the fitness between an product and user using "fit" feature as our label and other features as model input. This is a meaningful predictive task as the ability to determine the fitness of product and user is important in any clothing recommender system and should be studied.

2.2 Data Pre-processing and Feature Engineering

Due to the complexity of the dataset different features requirs different level of pre-processing, which is described in detail here:

- Cup size: Categorical feature of 12 unique values, one option is to do one-hot encoding on this feature, however since each cup size is associated to some underlying physical quantity, instead of one-hot encoding we transformed it from categorical value to numerical value is associated with each cup size.
- Height: Original data for this feature is in empirical system (XX ft XX in), we transform it to metric system and represent in numeric value in centimeter.
- fit: The label of our task, for each label, we encode it into a unique integer (fit-0, small-1, large-2).
- shoe width: Also transformed into numerical value form categorical value based on physical size.
- Category: The type of clothes, we apply one-hot encoding to the feature.

- review text: Since the sentiment of the review is correlated with the user experience, using pre-trained BERT [1] language model, we conducted text sentiment analysis to the review text ranging form -1 to 1 where -1 represents absolutely negative sentiment and 1 represents prefect positive sentiment.

We leave user IDs and product IDs as it is for down the line latent factor model to deal with. Since some feature offer redundant information (user name, review summary) we removed those from our feature vector. We also applied different feature selection and scaling operations on the model we use, we will discuss them in the Baseline and Method sections. We recognize the the inclusion of review text as a part of our feature is sometimes not realistic as it is not usually available during the recommending phase, however we want to study the uses of text-based features and language model in recommender system, so we experimented with both the inclusion and exclusion of this feature, we discussed this issue more in the result section.

2.3 Baseline

For the baseline we employ some basic statistical learning models. We have prepared three baselines: Logistic regression, KNN, and random forest classifiers. To deal with the missing values we use KNNImputer to make inferences on the missing values, KNNImputer works by finding missing values base on the K-nearest neighbors. After filling the missing values, we normalize each numerical features and then three baseline classifier are applied, the configs and results of each baseline are discussed in result section.

2.4 Metric

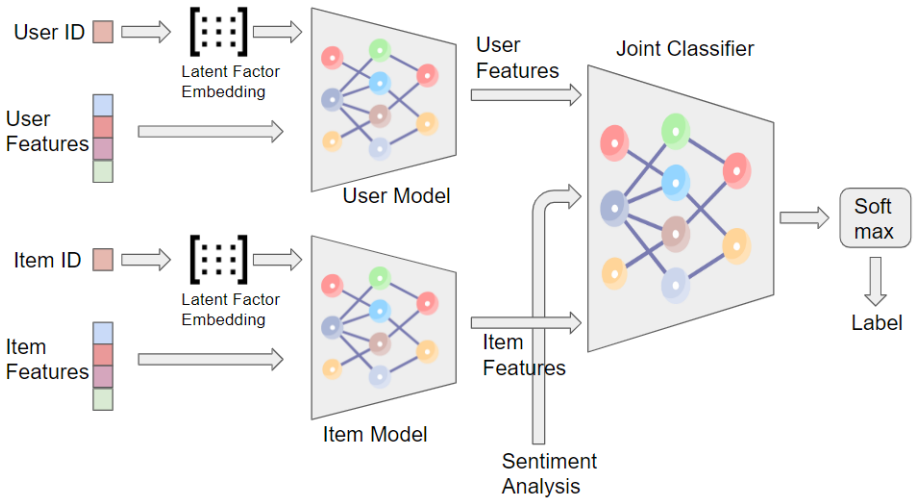
We employed multiple standard metrics to evaluate our models. Accuracy F1 and AUC score are used for evaluation. However, due to the class imbalance issue, we also applied weighted where Accuracy, F1 and AUC score are calculated with respect to each class and averaging among classes.

3 Method

The dataset contains 47958 unique user and 1378 unique product and 82790 records. So it is relatively dense dataset meaning a user likely has interacted with multiple products and a product has been reviewed by multiple users. To fully capture the user item interaction, we employ an latent factor model to model the relations between user and item, for that we created a deep learning model where the user and item as well as interactions are learned through stochastic gradient descent.

3.1 Model and Training

There are three major components to our model, the user / item latent factor embedding, user / item feature model and joint classifier.



We first divide features into user features and item feature, for example, clothing category would be considered as item feature where as height would be considered user feature. Both the user id and item id are first encoded into latent factor vector and joined with other features, we call these vectors as user feature and item feature. User feature and item feature are then fed into user / item feature model and then, user latent feature and item latent feature are concatenated together to form the joint feature vector. Since sentiment analysis of user's review is dependent on both the user and item, the sentiment analysis feature mentioned earlier is also concatenated with joined features. Then, the entire feature vector is fed into final classifier to predict the fitness given user and item. We experimented with two basic building blocks for our networks, feed forward layers and jump connection layers (ResNet [2]) and analyze both results in the later section. For features, we consider item id, category as item features. User id, cup size, waist, hips, bra size, height and show size as user features, missing values of each feature are inputted with a simple statistical imputer. Finally, the review text is used for joint features. We train the network with cross entropy loss and adaptive momentum optimizer for more stable gradient, the results are discussed in later sections.

The intuition about this neural network architecture comes from SFNet [4]. For this assignment, we tried four variants of SFNet and will show the results in later sections. The differences among these four models are: whether we take sentimental analysis as an input into the neural network, and whether we choose MLP or skipped connection layers (ResNet [2]) after the embeddings. The original SFNet[4] is most similar to the model using skipped connection layers

without sentimental analysis. We will then have these four models: SFNet-MLP/Res with/without sentimental analysis.

3.2 Hyper-parameter Tuning

We trained the Multi-layer-Perceptron(MLP) model with different hyper-parameters.We tried random search to find the hyper-parameters for the model that give the best accuracy. Random search is a better algorithm as compared to the traditional grid search algorithm as it takes less time to finish and therefore can handle larger search space. The best accuracy obtained using the MLP model is 0.695. We also found that model learning rate of of great importance in training our model, a low learning rate and more number of epochs works better than having a large learning rate. For user, item and joint models we used three layers with [256, 128, 64] hidden units. Batch size of 128 and a random drop out of 0.3 to improve the generalization of our model. User and item latent factors have 10 dimensions.

4 Literature Review

The clothing fit dataset is an existing dataset available on Professor McAuley's website. There are quite some papers that try to do personalized size recommendation clothing fit dataset. In this section, we describe a few approaches that have been tried out -

- McAuley et al. use a latent feature model. They use large margin nearest neighbour with prototyping to handle label imbalance issue. [3]
- Some other papers incorporate Bayesian approach and its variations like hierarchical Bayesian models.
- Our work for the neural network models is inspired from Abdul-Saboor Sheikh et al. which frames the optimization problem as a likelihood maximization problem. [4]
- It is rather generic in the context of collaborative filtering and is closely related to Neural Collaborative Filtering(NCF) and can be seen as a generalization of logistic matrix factorization.

The current SOTA implementation employs deep learning method to learn the fitness semantic between user and item latent features. We take inspirations from this line of work, attempted to learn separate user item features in the context of other available feature and review sentiment analysis for better performance.

5 Result

As mentioned in the early section we used three metrics to measure the performance of baseline and our model, to deal with the class imbalance issue, we use weighted metrics, that means each metrics is calculated for each class and then the performance of each class is averaged for final scores.

Method	Acc	F1	AUC
KNN	66.9%	58%	-
Logistic Regression	68.01%	58.03%	-
SFNet-Res(w/o sentimental analysis)	69.20%	58.99%	69.68%
SFNet-Res(w/ sentimental analysis)	69.31%	60.31%	67.81%
SFNet-MLP(w/o sentimental analysis)	69.16%	62.85%	70.76%
SFNet-MLP(w/ sentimental analysis)	68.72%	55.98%	65.87%

For KNN classifiers, we use K=10 and for logistic regression we applied regularization with C=0.1. Our model is denoted as SFNet, "-MLP" suffix meaning linear layers are used in model, "-ResNet" means ResNet layers with jump connections are used instead.

Upon inspecting the above results, we observed that ResNet classifier architecture did not help with the model performance, Residual connection architecture is most suitable for training deep neural networks as a way to alleviate gradient vanishing problems, since the model we uses only have a few layers, it is likely the reason this particular architecture did not help much. We also observed that by applying deep learning approaches, we have only managed to achieve marginal improvement from our baseline models, we highly suspect it is a result of feature imputing steps as it can introduce a lot of noises and incorrect information to the dataset thus led to worse result, so other techniques or develop a model that work well with missing values is a potential future direction of this work. Lastly, we recognize that the inclusion of review text sentiment analysis is sometimes an unfair assumption as review text usually only become available when the fitness of a user item pair is available, however, we still included it in our study to explore the potential uses of language model in recommender system, or in some special cases that review text are provided but not the fitness or the review text are revealed before the fitness information. In case review texts are not available, additional, we also study and included results of the same model without using review text features as shown in above table.

References

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[3] Rishabh Misra, Mengting Wan, and Julian McAuley. Decomposing fit semantics for product size recommendation in metric spaces. In *Proceedings of the 12th ACM Conference on Recommender Systems*, pages 422–426, 2018.

[4] Abdul-Saboor Sheikh, Romain Guigourès, Evgenii Koriagin, Yuen King Ho, Reza Shirvany, Roland Vollgraf, and Urs Bergmann. A deep learning system for predicting size and fit in fashion e-commerce. In *Proceedings of the 13th ACM conference on recommender systems*, pages 110–118, 2019.