H&M Personalized Fashion Recommendations Project

Christos Panourgias

Contents

1 Introduction

The H&M Group is a family of brands and markets with 53 online markets and approximately 4,850 stores. The H&M online store offers shoppers a wide variety of products to browse. But with too many options, customers may not quickly find what they're interested in or what they're looking for, and ultimately may not make a purchase. To improve the shopping experience, product recommendations are essential.

In this work, a product recommendation system will be developed using data from past transactions as well as customer and product metadata. The available metadata ranges from simple data such as clothing type and age of the customer to textual data such as product descriptions.

2 Problem Definition and Motivation

While the H&M Group online store offers a wide range of products, the sheer volume of choices can be overwhelming for customers. As a result, customers can struggle to find the products that align with their interests. This challenge not only affects the customer's ability, but can also affect the performance of the business, it can lead to reduced conversion rates and lost sales. In addition, excessive returns in the fashion industry contribute to emissions from transport, negatively impacting environmental sustainability. Therefore, the development of a robust recommendation system becomes crucial as it aims to address these challenges by providing personalized product recommendations, optimizing the shopping experience and promoting sustainable practices. The final file that will be produced will recommend up to 12 products for each customer and will be evaluated based on the following metric:

$$MAP@12 = \frac{1}{U}\sum_{u=1}^{U} \frac{1}{min(m,12)} \sum_{k=1}^{m} min(n,12) P(k) \times rel(k)$$

where U is the number of customers, $P(k)$ is the accuracy up to k, n is the number of predictions per customer, m is the number of ground truth values per customer, and rel(k) is a characteristic function that is equal to 1 when the product at position k is a correct prediction and 0 otherwise.

3 Dataset Description

3.1 Transactions (Number of samples 1371980)

The core data set is transactions_train which contains information on all transactions occurring during the two years starting from 20/09/2018 and 22/09/2020. Specifically, the following information is provided:

- **t_dat :** The date of the respective transaction.
- **customer_id:** The key that characterizes each customer.

- **article_id:** The key that characterizes the product purchased.
- **price:** The price of the product.
- **sales_channel_id:** The product's purchase environment, where 1 = live and 2 = online.

3.2 articles (Number of samples 105542)

This set contains metadata about the store's products. Below are only the features that were used:

- **article_id:** The key that characterizes the respective product.
- **detail_desc:** Detailed description of the product.

3.3 customers (Number of samples 31788324)

This data set contains metadata about the clients. Below are only the features that were used:

- **customer_id:** The key that characterizes each customer.
- **club_member_status:** The status of the customer in relation to the loyalty club of the store through which points and various discounts are offered. Where ACTIVE = active member of club, PRE-CREATE = not yet joined, and LEFT CLUB = no longer a member of club
- **fashion_news_frequency:** The frequency of informing the customer about fashion. Where REGULARLY = more often than once a month, Monthly = exactly once a month, and NONE = not at all.
- **age:** The age of the customer.
- **postal_code:** The postal code of the customer.

4 Description of Data Analysis Method

4.1 Data Processing

The data in the customers dataset have many nan values, which, depending on their magnitude, will be filled or the column that contains them will be removed. FN and Active columns have 65% empty values and are therefore not used. To fill in the remaining columns, the MICE(Multiple Imputation by Chained Equations) method was used, this method is popular because it takes into account the relationships between variables when imputing missing values, making it more accurate than simpler data-filling techniques. The articles data set has 416 products which have no description. For these products the description is replaced with an empty string. The set transactions_train has no missing values and is therefore taken as is.

4.2 Text similarity and clustering using LSH

4.2.1 Text Preprocessing

Initially, the letters of all product descriptions are converted to lowercase, then special characters (@, #, $ etc.), numbers and non-contributing words are removed from these descriptions in the text information (stopwords : ”a”, ”an”, ”the”, ”in” ”on” etc.). Finally, lemmatization is applied to the texts so that only the foundations of their words are extracted.

4.2.2 Application of LSH and Clustering

The basic idea in using the Locality Sensitive Hashing (LSH) method is to find similar product texts and then cluster them to form clusters with similar products, so customers who have previously purchased a product may be interested in similar products, or similar customers may be interested in products of the same cluster. Using LSH in this particular problem might be a good idea for several reasons. By using LSH, we can quickly identify items that are likely to belong to the same class, simplifying the subsequent analysis. In addition, LSH has the advantage of scalability, allowing us to process large data sets with reduced computational cost. This makes it particularly beneficial in this problem because of the large volume of customer, product and transactions data sets. The process begins by applying LSH with words as shinglings to the processed product descriptions and using 128 permutations of the count vectorizer table. Therefore a measure is derived by which we can identify similar product descriptions, and thus similar products from the set of all products. The LSH application ran in 67.44 seconds and then the clustering using the LSH results ran in 158 seconds. It is worth noting that these time measurements are quite low given the volume of the sets.

Next, nearest neighbor clustering is applied using the result of LSH with query each description. The 1000 nearest neighbors are entered into the same cluster, thus creating 304 product clusters.

4.3 Creating Datasets for Training

For each sample in the sets articles and transactions_train the number of the class corresponding to it is added based on the clusters created using the results of the LSH method as a measure of similarity.

Then the data set of transactions (transactions_train) is divided into training set (number of samples 1333465), validation set (number of samples 249511), and test set (number of samples 75481) where the test set contains the transactions that occurred in the last 7 days, the validation set contains the transactions that occurred in the last 30 transactions before the 7 days of the test set, and the training set contains all the remaining transactions.

For each of the above sets, grouping is applied based on customers (group by customer_id) and then the customers data set is joined with the set of transactions with respect to the customer_id attribute.

The above process is repeated for the entire data set of transactions transactions_train so that the model that will be created at the end, exploits the entire data set in its training. Therefore four data sets, train, val, test, and train_all have been created.

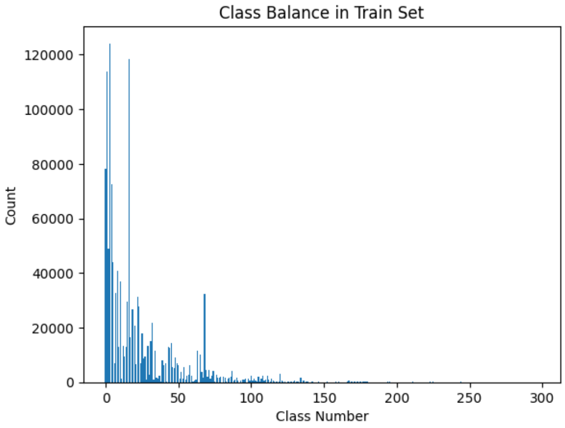
For the above data sets, for each customer, the average price is calculated and added as an attribute so that the model takes into account the price range of the products they buy. In addition, the product purchase environment, and product key columns are removed, and the categorical attributes are converted to integers so that the model can process them.

Then, for each customer, probability vectors of the same dimensions as the clustering classes are created in order to find the class from which a product is most likely to be purchased by each customer. The maximum of that vector is taken as the corresponding class for the given customer.

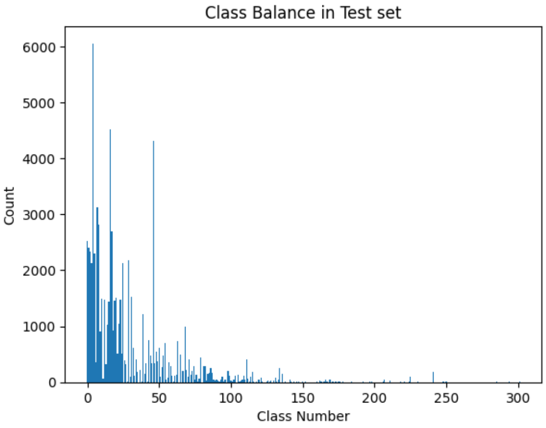
Therefore the attributes club_member_status,
fashion_news_frequency, age, postal_code and mean_price will be used to predict the class. Before training the model, the features of the training, validation, and test sets are normalized by the MinMax method to the range $[0, 1]$ based on the training set, and the features of the set containing all the data are normalized based on all of the data.

4.4 Class Balance

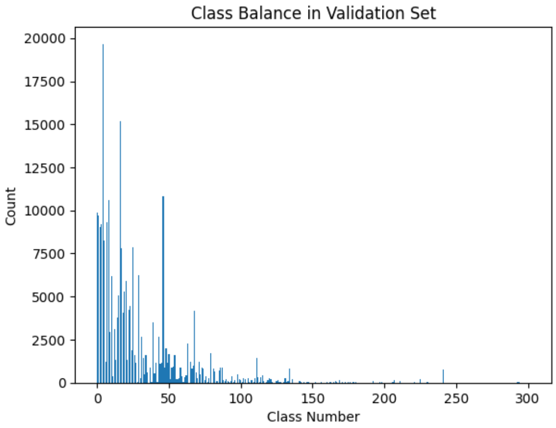
Below is the balance of classes in each data set.

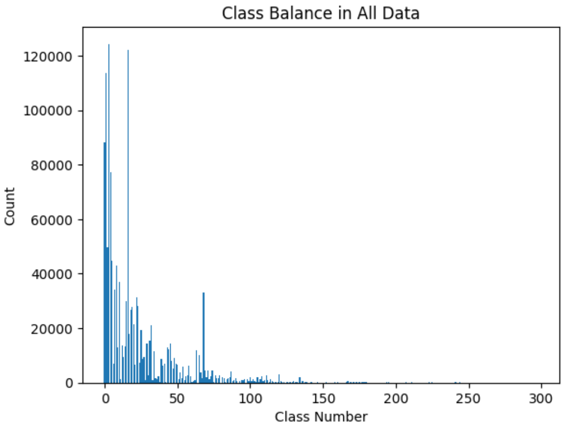


Class Balance in Train Set

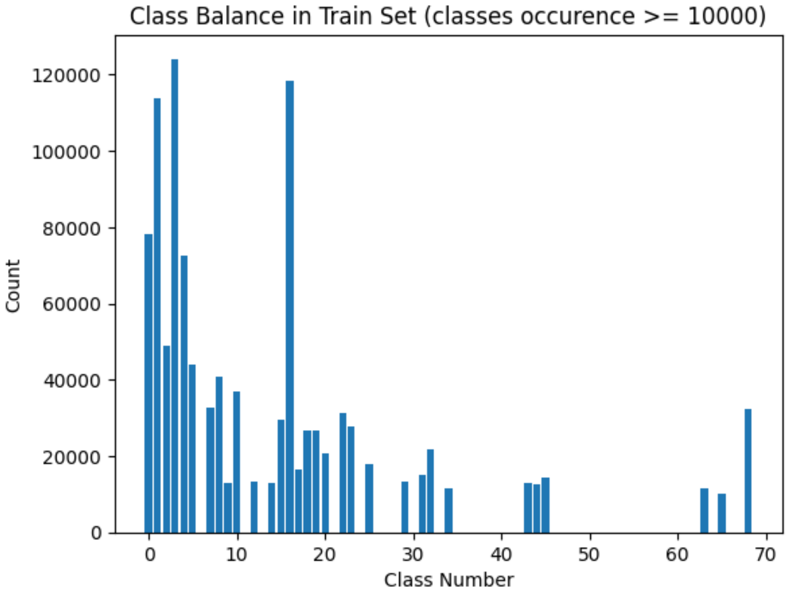


Class Balance in Test set





It is observed that most customers buy products mainly from the first clusters. Since the most likely clusters that can occur are those that have appeared more than 10000 times, then it makes sense to delete the samples, in the training set, that correspond to the classes that appear less than 10000 times. Therefore the training set now has a number of samples, 1101678 and thus the following illustration is obtained



Note: Absolute class balancing using over-sampling creates repetition of data patterns and due to over-fitting, produces worse classification results, so was not performed.

5 Experimental Results

Four classification models were implemented in the experiments, namely, K-Nearest Neighbors (KNN), Decision Trees, Logistic Regression, Light Gradient Boosting Machines (LGBM) and Linear Discriminant Analysis (LDA)

To select the best model, k-fold cross-validation was used on a sample of the 22033 crowd training set (for speed of operations), thus producing the following validation set accuracy results:

- KNN: 0.0414, time: 0.016
- Decision Trees: 0.0452, time: 0.136
- Logistic Regression: 0.0581, time: 1.292
- LGBM: 0.0754, time: 5.788
- LDA: 0.0738, time: 0.025

The LGBM model produces the highest accuracy, so this will be used as the base model. However, it is worth noting that the LDA model had little deviation in accuracy from LGBM and in a much lower time, which means that when training even larger sets it might be a better choice as a base model. The LGBM model was trained on the training set in 77 seconds and the following results were produced for the test set.

- Accuracy : 0.0955
- F1-Score : 0.0425
- Recall : 0.0955
- Precision : 0.0466

The model was then trained on the entire data set in 85.7 seconds. The $MAP@12$ metric for the extracted file is equal to zero which implies that the model failed to make good recommendations based on the proposed metric.

6 Critical Evaluation of Results

The metric accuracy = 0.0955 means that only 9.55 % of the predictions made by the model were correct, while the remaining 90.45 % were incorrect. This suggests that the model had difficulty accurately classifying each category for the data.

The metric F1-score = 0.0425
indicates a very low level of accuracy. This implies that both precision and recall have quite low results. The metric Precision = 0.0466 indicates that only 4.66 % of the cases predicted as one category were actually correct. This indicates a

high misclassification rate, as the model tended to incorrectly assign the class label in many cases.

The metric Recall = 0.0955 indicates that the model could accurately identify only 9.55 % of true positive cases, while missing a significant portion of them. This means that the model had difficulty capturing the true nature of each category, leading to a high rate of false negatives.

Finally, the metric $MAP@12 = 0$ means that none of the predicted items for a customer in ranks 1 through 12 were relevant or correct labels. In other words, the model did not make accurate predictions for that particular customer. The $MAP@12$ metric measures the average accuracy across all clients, taking into account the accuracy at each rank and the relevance of the predicted elements. A rating of zero indicates that the model failed to identify any relevant information about the customer, indicating a poor performance in making accurate purchase predictions.

The poor performance of the model can be attributed to several factors, the quality of the clustering performed using LSH may have affected the predictive performance negatively if the clusters formed do not effectively group similar elements together or if they introduce noise into the data . It is also critical to ensure the availability of a sufficient amount of diverse and representative training data. Unfortunately, the data does not have enough balanced instances for each class and this can lead to overfitting, where the model becomes too specialized in the training set and fails to generalize well to new data inputs.

7 Conclusions

In conclusion, the results obtained from the model using LSH and LGBM clustering were highly unsatisfactory, showing remarkably low values for precision, F1-Score metric, recall and precision but also for the evaluation metric $MAP@12$. It is evident that the model faced significant challenges in accurately predicting the classes in the provided data set. Several potential factors are identified that may have contributed to this underperformance. The feature selection process may have been inadequate, failing to capture critical information necessary for accurate predictions. In addition, the clustering performed using LSH may have created noisy clusters, hindering the model's ability to distinguish significant patterns. Additionally, the choice of the LGBM algorithm may not have been optimized for this particular problem. In order to improve the results, the need to reconsider the feature selection, to improve the clustering methodology, and to use alternative models than those used or even different approaches is recognized.