**Recommendation system for H & M Personalized shopping**

Group 5-Data Blazers

Team Members:

Sai Teja Jonnala,-EDA, Content-Filter Based Recommendation

Yawo Eklou-Collaborative-Filter Based Recommendation Default, Feature Engineering

Suneeth Sreedharan-Collaborative-filter Recommendation Surprise, Feature Engineering

Course: DSBA 6156

Instructor: Dr. -Depeng Xu

Date Completed:12/11/2022

# Executive Summary:

## Business Problem Statement

*  Hennes & Mauritz (H&M) is a Swedish multinational clothing-retail company that provides fashion clothing for men, women, children, and teenagers.
* Opened in 1947 now operates in 187 countries
* 4801 stores in 74 countries
* Sales in 2021 were 23 billion worldwide
* H & M sales have been stagnant since 2017 and revenue has decreased due to shrinking margins, increased overhead, supply chain issues caused by world events, and an abundance of products that are not selling.
* They have a large group of consumers that are members of their fashion club that are not contributing to sales on a regular basis and who because we already have captured their information would be easy to market to.
  + What do we market to them?
  + How could we lower cost and free shelf space for these additional recommendations?
* We worked towards a previous Kaggle competition hosted by H&M Group to develop product recommendations based on data from previous transactions and customer and product metadata.

## Our Approach-Road Map

After we are done with EDA, we would perform Content-Filter Based Recommendation based on Article Descriptions, using an Unsupervised Method. This would help us gain evidence of recommendations and help us build a Machine learning Collaborative filtering Based recommendations model. Prior to performing Supervised based recommendations, we would perform feature engineering to transform the Transaction’s Time series data into a Normalized recommendation problem.

## Data Profile

Transaction time series data were collected from Kaggle Competition, between the period of September 2018 to September 2020:

* Total transaction records: over 28 million
* Total unique customers: over 1.3 million
* There are 105,542 unique items in H&M’s inventory

# Project Report

## Exploratory Data Analysis

We have got 4 classes of data files from Kaggle competition:

1. Articles Catalog data
2. Customers' profile data
3. Transactions data (time series)
4. Images of each item from Article catalogs

## Article Catalog Data

This table contains all h&m articles with details such as a type of product, a color, a product group, and other features. See appendix for Article Data description.

## Customers' profile data

See appendix for customer Data description.

* + 1. Transactions’ profile data

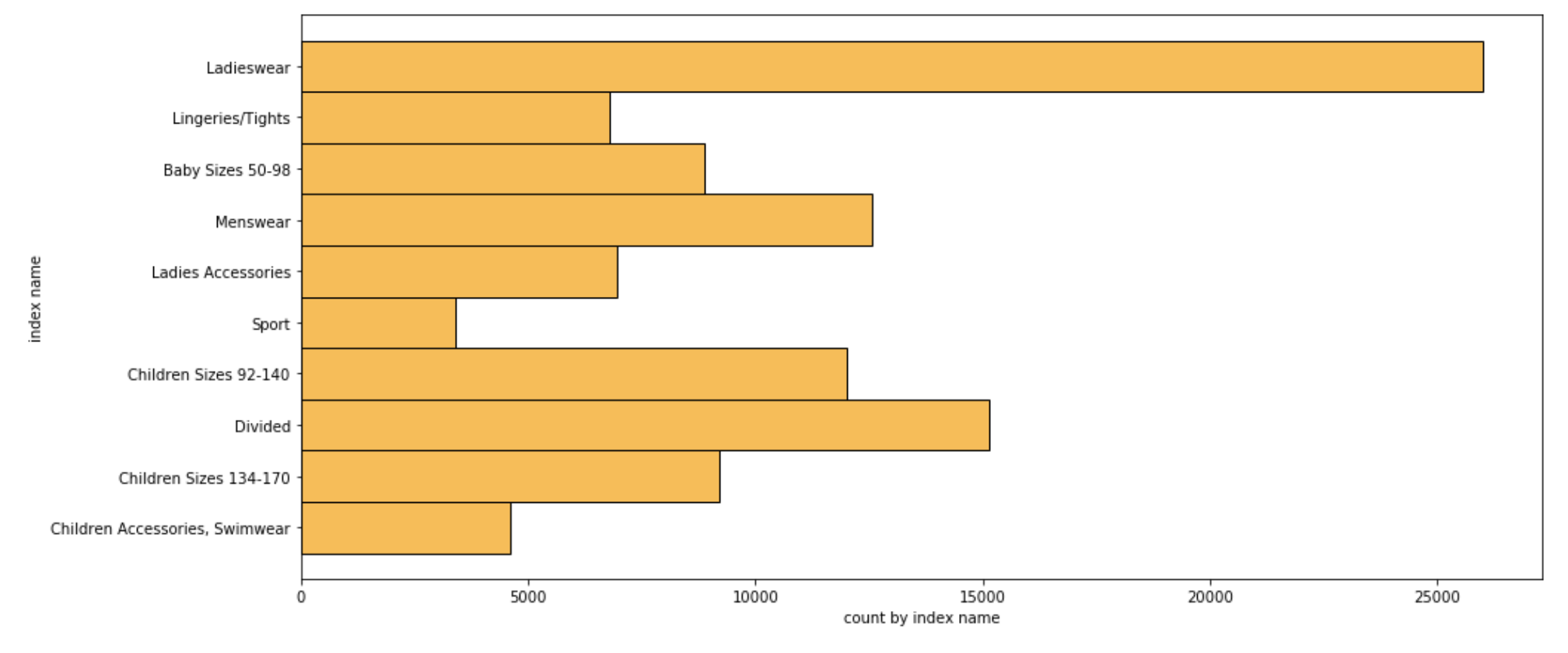
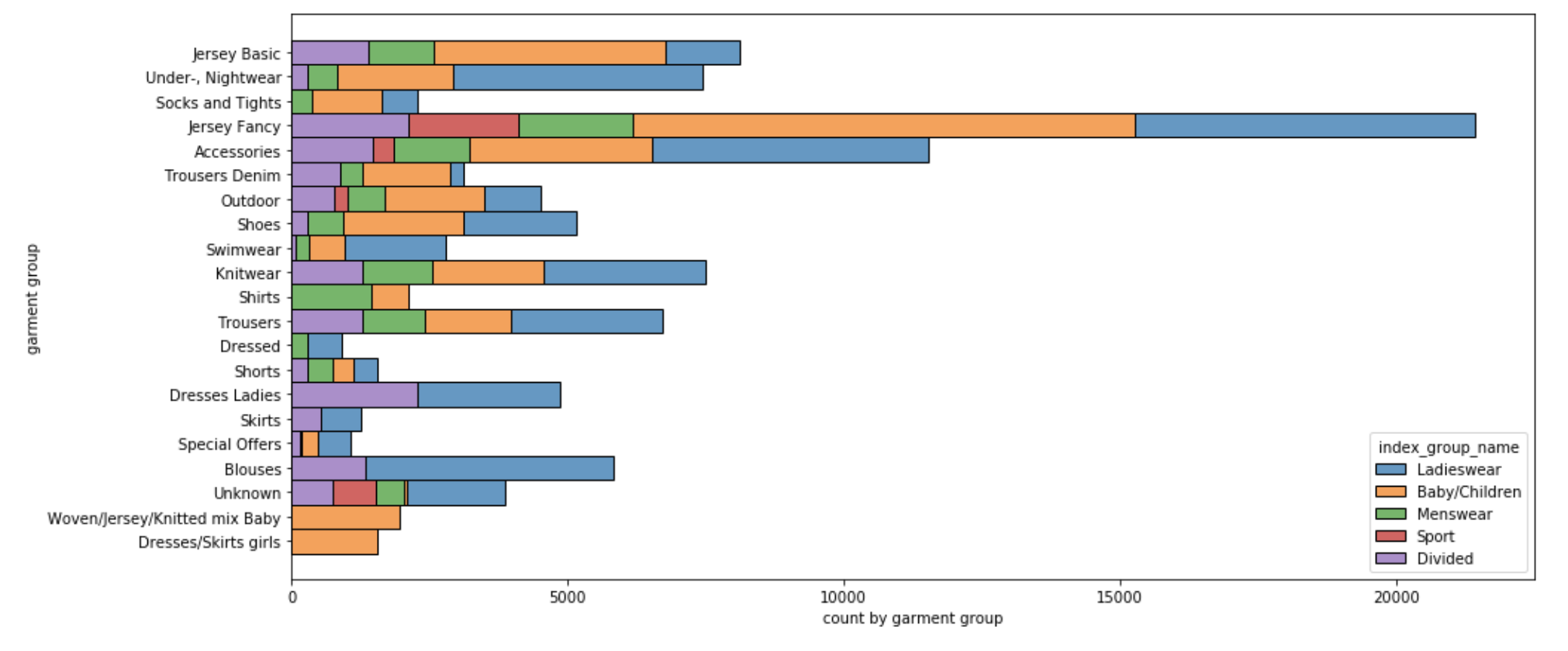
See appendix for Transaction Data description.

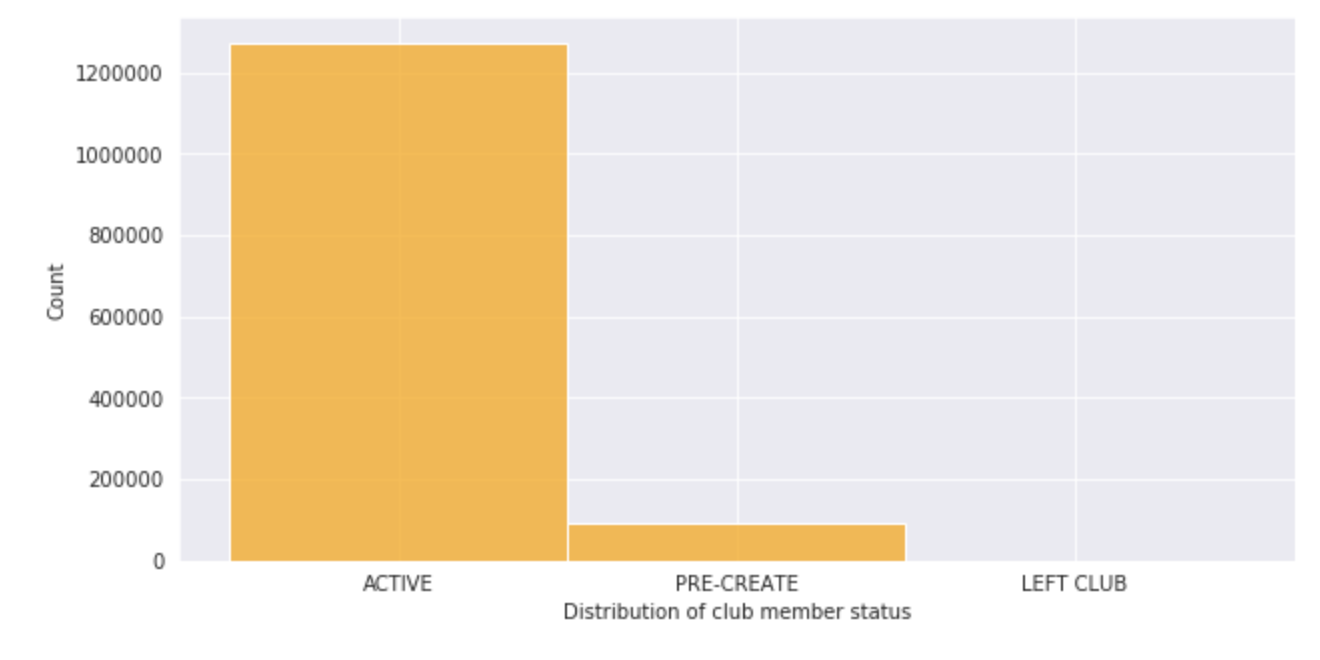
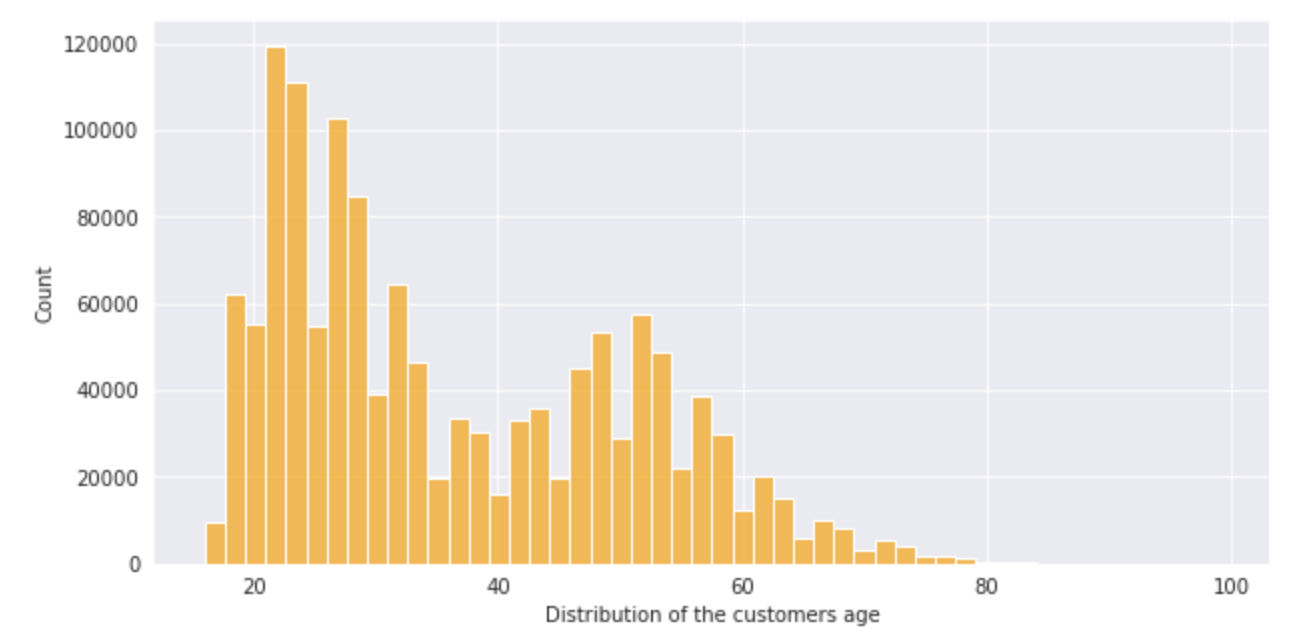
* + 1. Exploratory Data Analysis

Ladieswear has the highest portion of all dresses.The smallest percentage is in sportswear.

The garments grouped by index: Jersey fancy is the most frequent garment, especially for women and children. The next category by quantity is accessories.

The most common age is about 21-23. Almost every customer has an active club status, and some of them begin to activate it (pre-create).

****



Since we have more active club members, we have reason to believe we would have information on a good majority of customers and their purchase history, good enough to make a recommendation system.

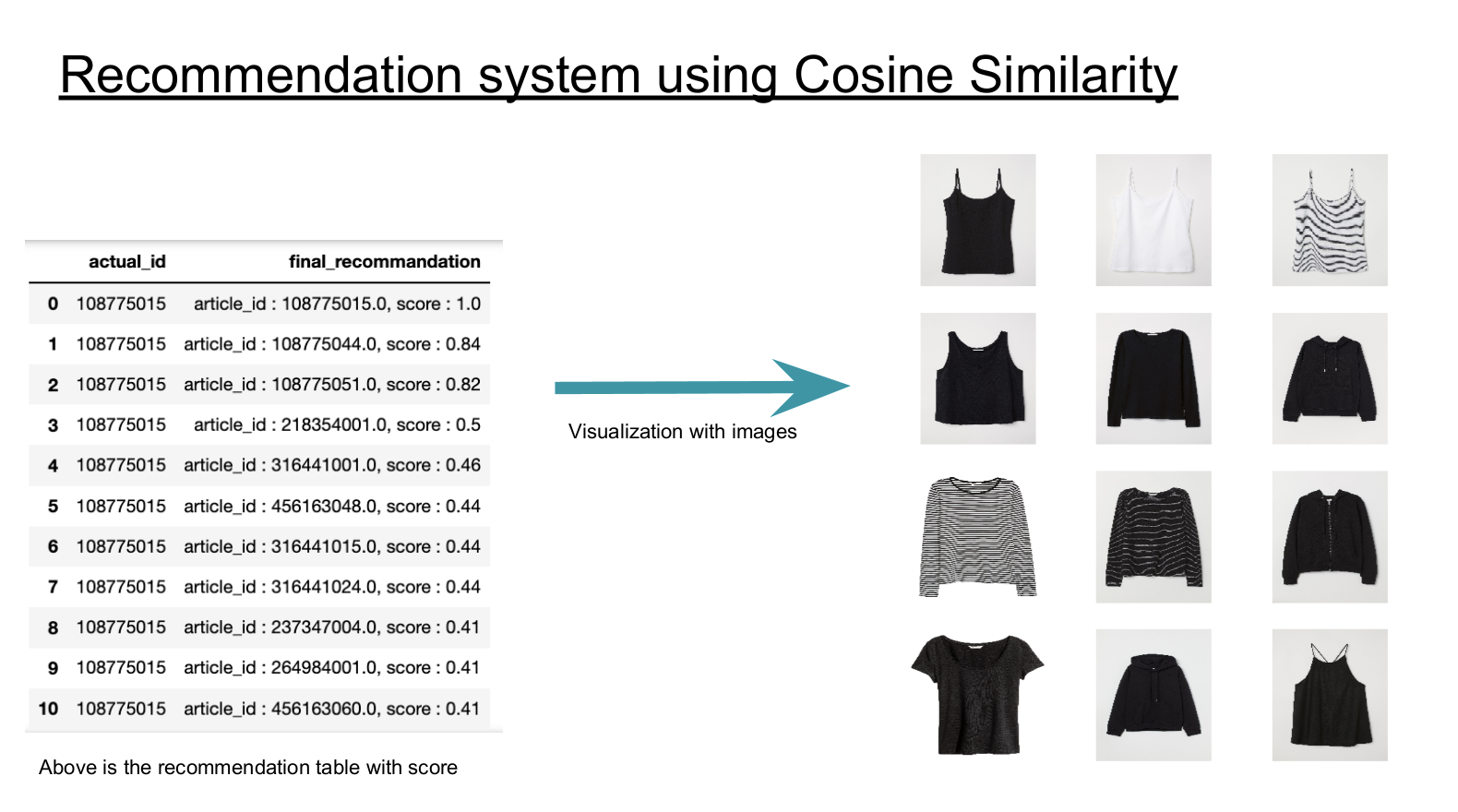
## Content Filter-Based Recommendation using Unsupervised Cosine similarity Method

In this project, each item is represented by an article. For example, in order to calculate the similarity between denim and Jeans I created two vectors with all the content information for each article and then calculated the vector’s cosine similarity.

Cosine Similarity is a measurement that quantifies the similarity between two or more vectors. We have chosen Cosine similarity because even if the two similar data objects are far apart by the Euclidean distance because of their size, they could still have a smaller angle between them.

The vector contains several 0 values to fill in null values. We have merged the information from variables like prod\_name, product\_type\_name, product\_group\_name,graphical\_appearance\_name, colour\_group\_name, perceived\_colour\_value\_name, perceived\_colour\_master\_name, department\_name, index\_name, index\_group\_name, section\_name, garment\_group\_name, detail\_desc to form a metadata of the customer. We have used the metadata to understand the similarity between various articles.

We calculated the cosine similarity between all the articles in the dataset and created an API to fetch the top 10 recommendations by passing the article id. Here in the below image, shows that the actual id is 108775015 and the next column indicates the score associated with each and every prediction.





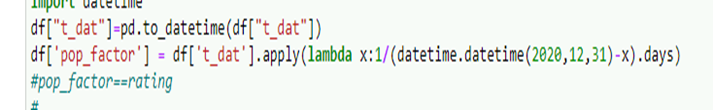
Hence the content filter-based unsupervised learning results is promising enough for us to conclude that there is enough evidence of possible recommendations, and thus we are proceeding towards supervised learning methods

## Feature Engineering

* + 1. Normalizing the Time series, and creation of Feature– Popularity factor.

We have come up with a feature based on the recency of the purchase made, the latest the purchase would be higher the value would be. This would help us avoid the time series data for our study.

Logic:



Find in the appendix the details of this Logic.

* + 1. Limiting data and splitting it into Training, Test, and validation sets based on transaction date

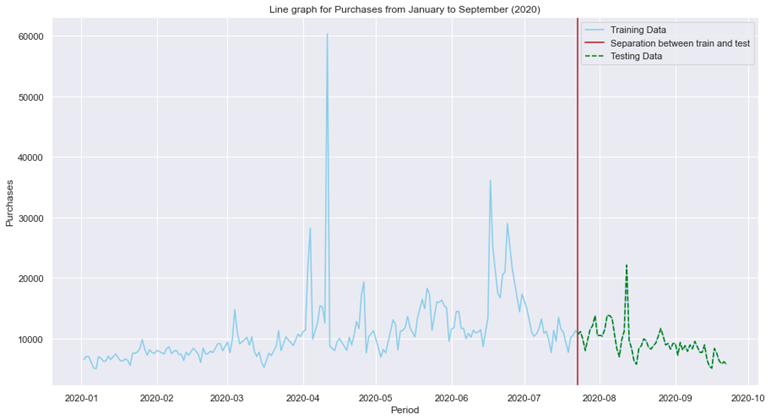
We would consider in our study the transaction of the year of 2020(the latest year in the dataset). This is in consideration of computational power and given the project time. Also, we would be running several algorithms to evaluate the result hence has to limit the data.

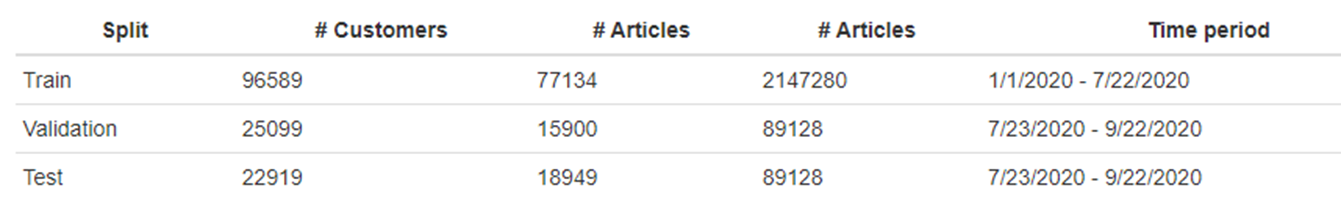
Another limiting criterion we adopted is, considering transactions for

* Articles with a minimum of 100 purchases.
* Customers with a minimum of 10 article purchase

Time period of transactions of Train data-1/1/2020 to 7/22/2020

Time Period of Transactions of Test Data- 7/23/2020 to 9/22/2020





Also in order to avoid over-fitting, for our tests, we have limited only 50% of the overlapping customer between Train and Test data.

## Collaborative Filter-Based Recommendation using Default Matrix Factorization method

## 

The Collaborative Filter-Based model is a machine learning predictive model used to make recommendations based on users' similar behavior. The principle is: “If User A likes article ‘1’ and ‘2’, but dislikes article ‘3’, and User B likes article ‘1’, then probably, User B may like article ‘2’ and dislikes article ‘3’ as well. Article ‘2’ will be recommended to User B while article ‘3’ won’t”. The metric used in our particular project to measure a customer's interest in an article is the popularity factor defined earlier in this project. In some cases, the users’ rating is the metric used. In this particular case, we used the default matrix factorization method in Sklearn API.

The following are the steps taken to build our model:

The Number of customers and articles corresponding are being filtered and result in the number of customers: 18610 and the number of articles: 7996, based on some specific criteria. **[Step1]**

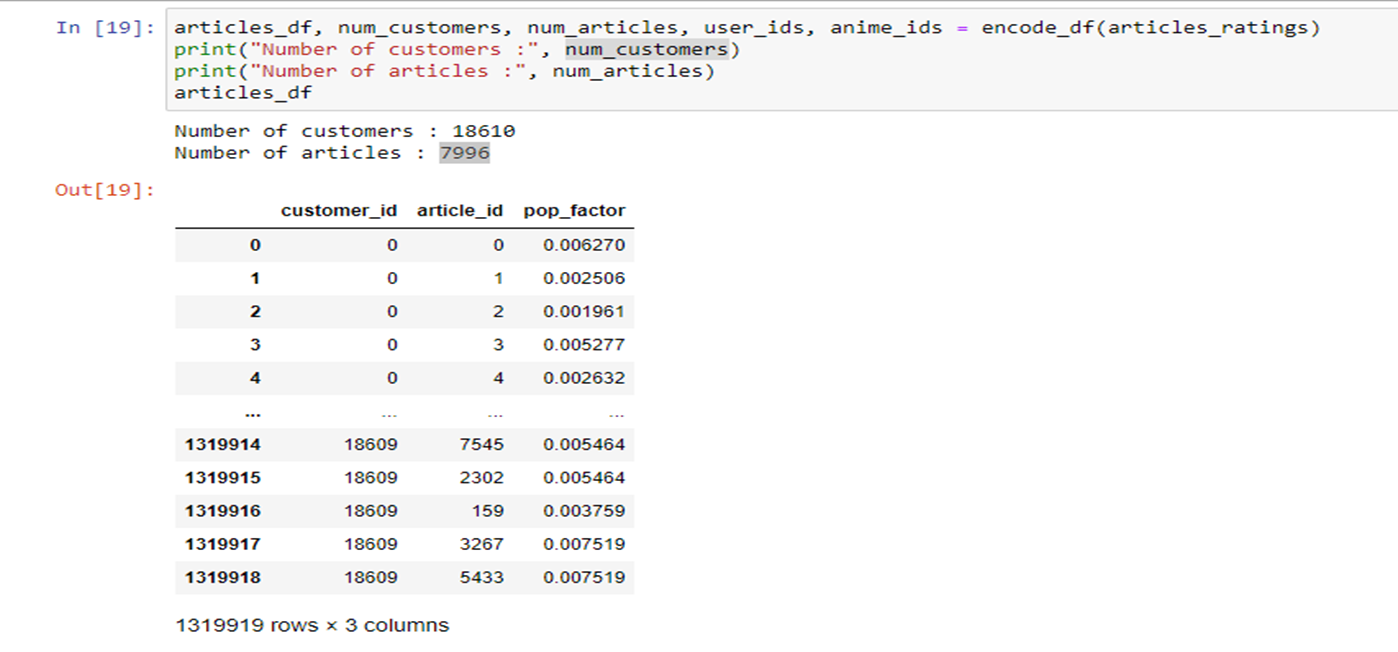
A Factorization Matrix is then implemented using the filtered data that combines the customer\_id, the article\_id, and the created popularity factor columns. **[Step2]**

A Python code that uses the Nearest Neighbors model in the Sklearn package is being written to create the top 10 article recommendations for one customer and the average popularity factor corresponding. **[Step3]**

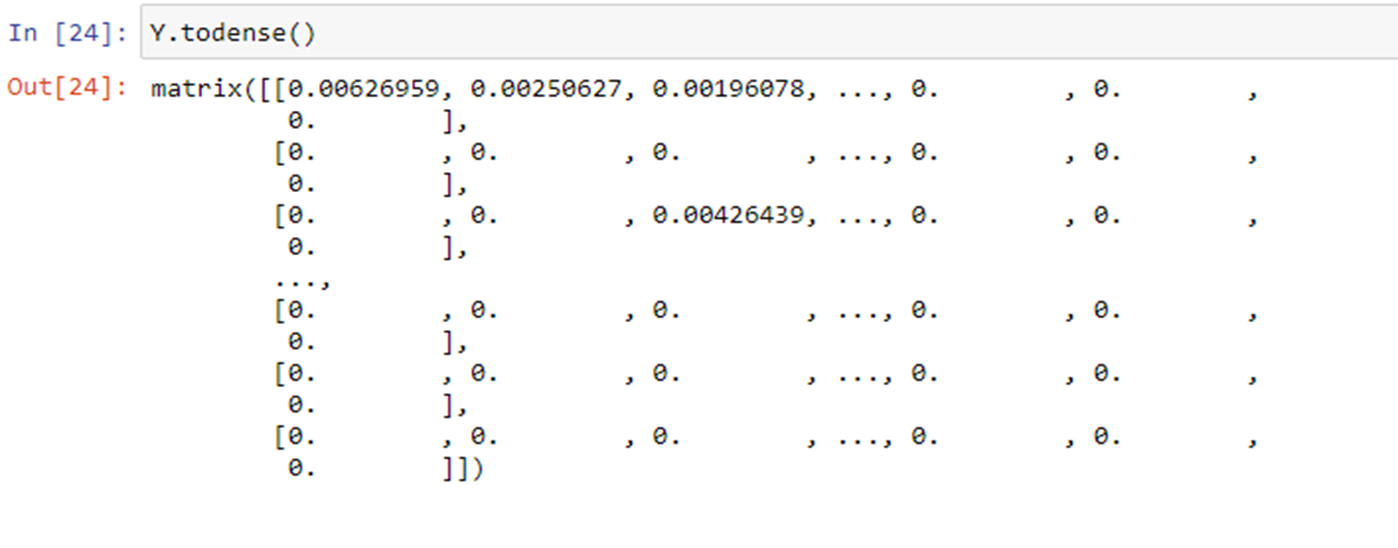
The top 10 article recommendation for one customer code above is then being inserted in a For Loop that ranges from 1 to number of customers to output the recommendation for each customer and the average popularity factor corresponding. **[Step4]**

The Visualization section is finally added to visualize the prediction results. **[Step5]**

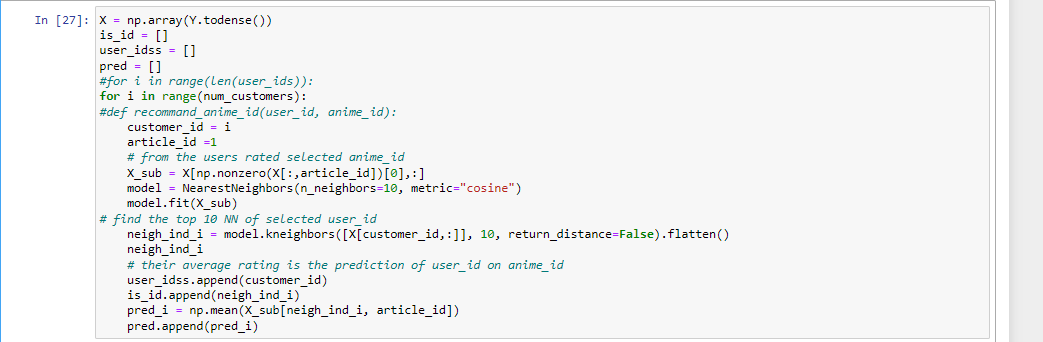
1- Data used after filtering the customer\_id column ( Number of Customers: 18610, Number of articles: 7996) **[Step- 1]**

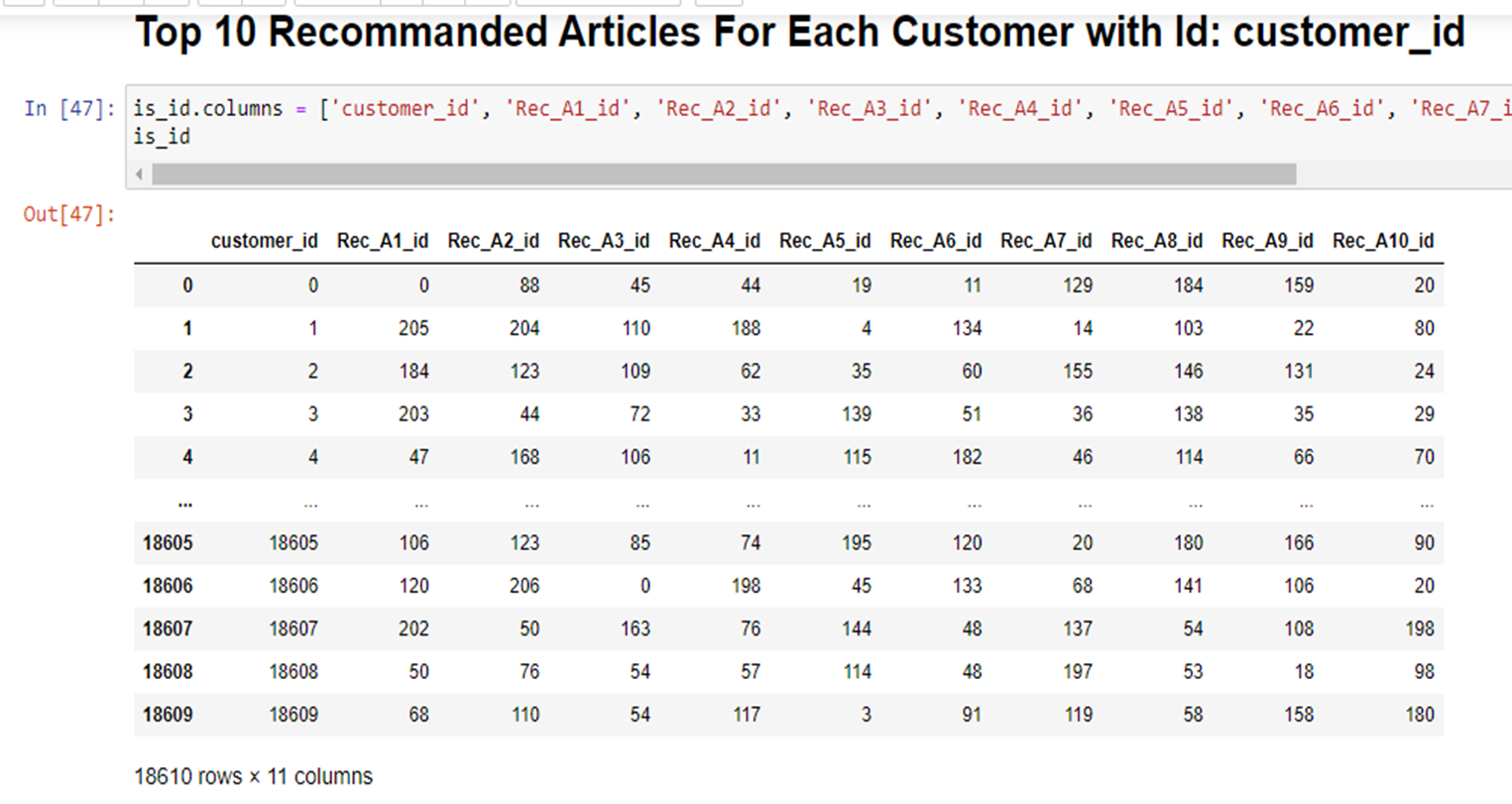


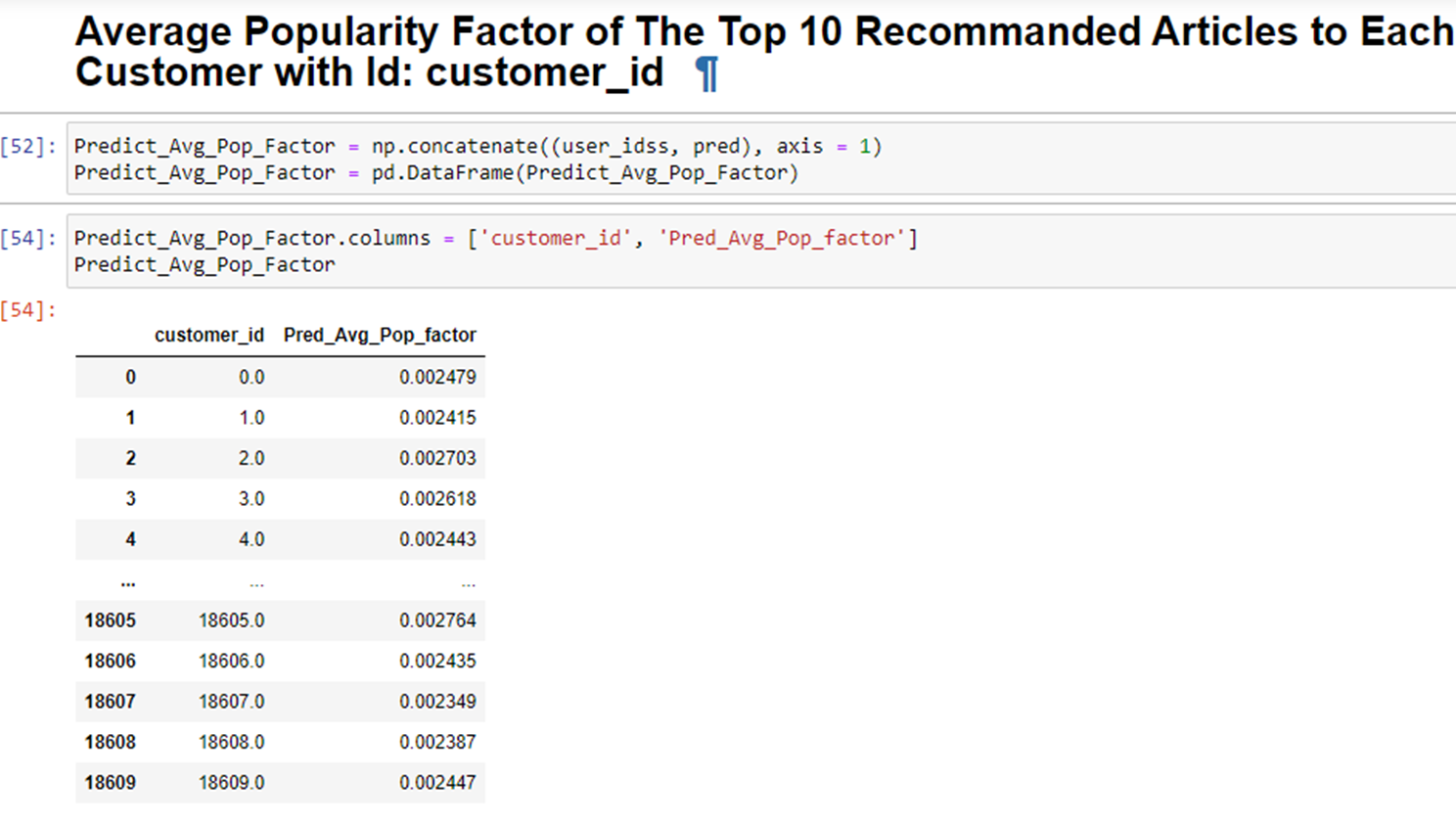
2- Matrix Factorization Base on 7996 Articles and 18630 Customers (Features)- **[Step-2]**



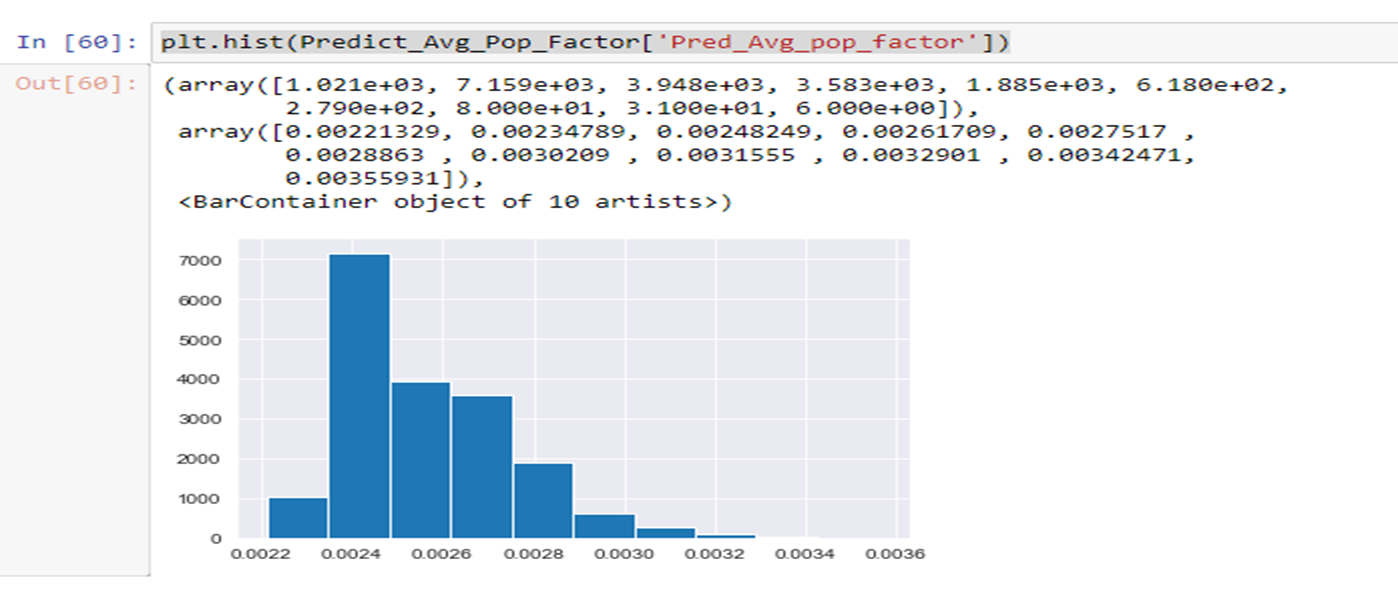
3- Python code to make recommendations to all customers **[Step 3-4]**

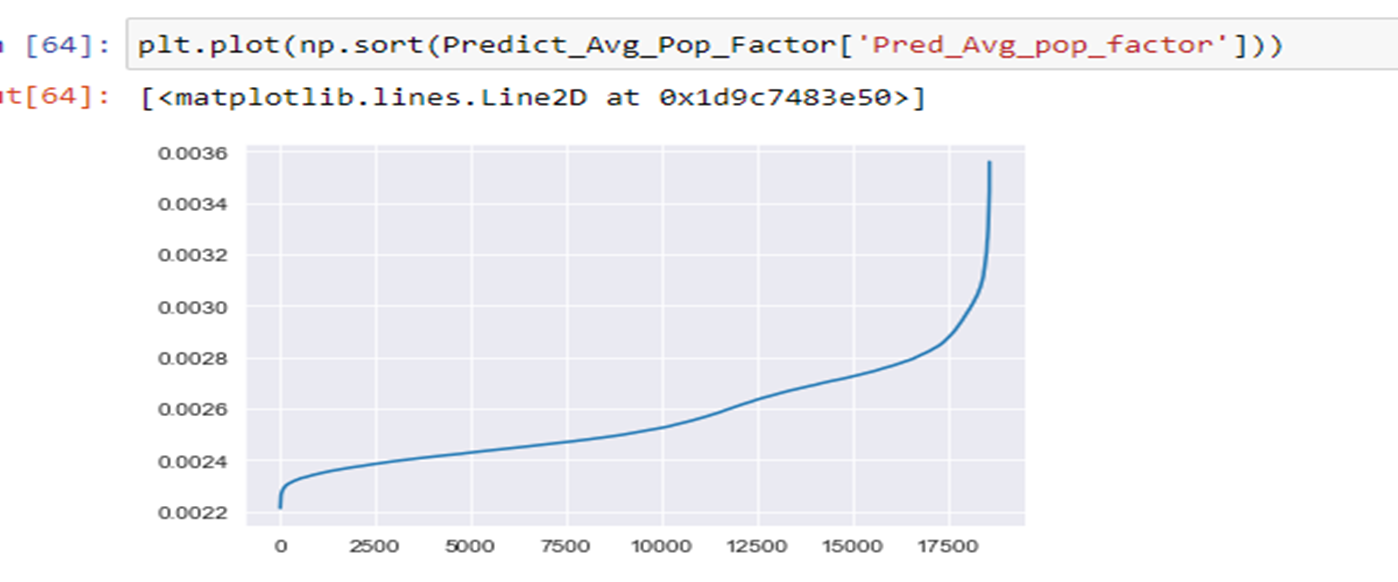






4- Distribution of Average Popularity Factor **[Step 5]**



Most of the recommended articles have an average popularity factor of **0.0024**. 

The higher the article popularity factor, the more the recommendation to customers.

## Collaborative filter-Based Recommendation using Machine Learning Model with ‘Surprise’ Package

Collaborative Filtering makes predictions based on user similarity. Its underlying principle is that if user A and user B have similar tastes, then articles bought by user A will probably be bought by user B too, and vice-versa. We used the train splits for user data from above and joined them with the transactions completed for the time range considered.

We used the Surprise Package to fit the Machine learning model based on algorithms:

1. Baseline
   * Surprise build-in algorithm based on Alternating Least Squares (ALS).
2. NMF Basic
   * We explored the `Non-negative Matrix Factorization (NMF)` from the `surprise` library. In NMF, specific step size is set for the stochastic gradient descent process of regularization. This is done to ensure that all user and item factors are kept positive. While this model can be prone to overfitting, this can be mitigated with steps to reduce the dimensionality of our data/factors.
3. NMF Tuned
   * We Used grid search to get best-fit parameters
4. SVD Basic
   * Also, we explored Matrix Factorization-based Algorithm -SVD as popularized by [Simon Funk](https://sifter.org/~simon/journal/20061211.html) during the Netflix Prize.
5. SVD Tuned
   * We used grid search to get best-fit parameters
6. SVD++ Basic
   * Further taking a look into the `surprise` library, we tried and utilized SVD++ algorithm, which is quite similar to SVD, however, the difference is that SVD++ attempts to add an extension onto the base SVD that uses implicit rating as well as explicit. In other words, it infers the action of rating an item as a latent factor regardless of the rating value given to the item while also factoring in the actual rating value. We thought this might further improve the RMSE.
7. SVD++ Tuned
   * We used grid search to get best-fit parameters

Also, we evaluated the performance of these models using RMSE.

Below is the evaluation result, based on the RMSE, the SVD++ Tuned is the best model and we choose it as the final ML recommendation model. Best parameter for SVD ++ Tuned algo-{'rmse': {'n\_factors': 1, 'n\_epochs': 50, 'reg\_all': 0.1}}.

It’s to be noted the Tuning of the SVD++ takes a long time to run-nearly 60 mins.

## Results

We have developed a Naïve SVD++ tuned-based model using the Surprise package, which would recommend, for a customer for his/her previously purchased combination of product type name, product group name, index name, index group name, and garment group name, would make 2 article recommendations per combination. In the example provided in Appendix the customer, based on her/his previous purchases has 5 combinations of product type name, product group name, index name, index group name, and garment group name. Hence in the recommendation set for the customer, we would have 10 articles(2\*5).

## Conclusion

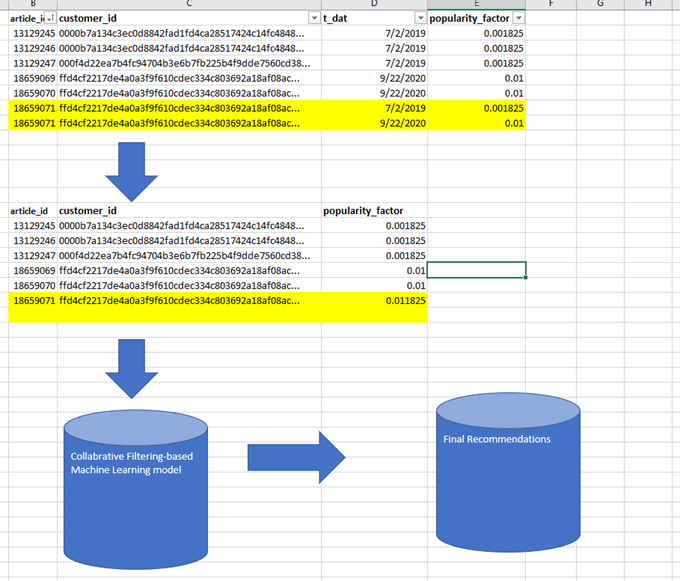
We were able to successfully come up with an ML model based on a Collaborative filter using the Surprise package. Also, we would like to note that the feature engineering of transforming/normalizing the time series was very crucial for the project.

# Appendix

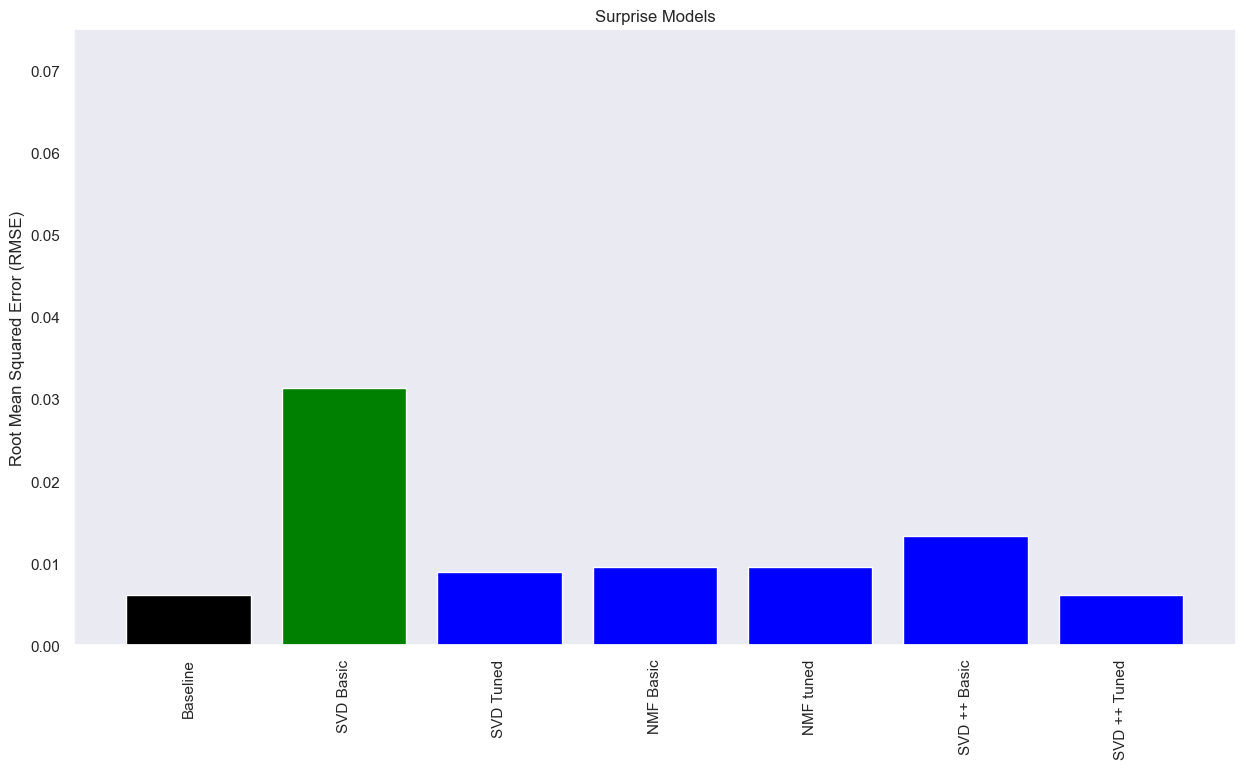
* 1. Code Notebook- EDA, Content-Filter Based Recommendation <https://drive.google.com/file/d/1t65rv8rTuxTdPZKmlRNA6LdmKy1PbhEa/view?usp=share_link>
  2. Code Notebook- Collaborative-Filter Based Recommendation Matrix Factorization, Feature Engineering <https://drive.google.com/file/d/17QfOi5uJaukzRq0cD3qJgMUh5AiEZFOG/view?usp=share_link>
  3. Code Notebook- Collaborative-filter Recommendation Surprise, Feature Engineering

<https://drive.google.com/file/d/17fyE1RLLTMFPAMbRvJ-OMCgikOG0Nnvr/view?usp=share_link>

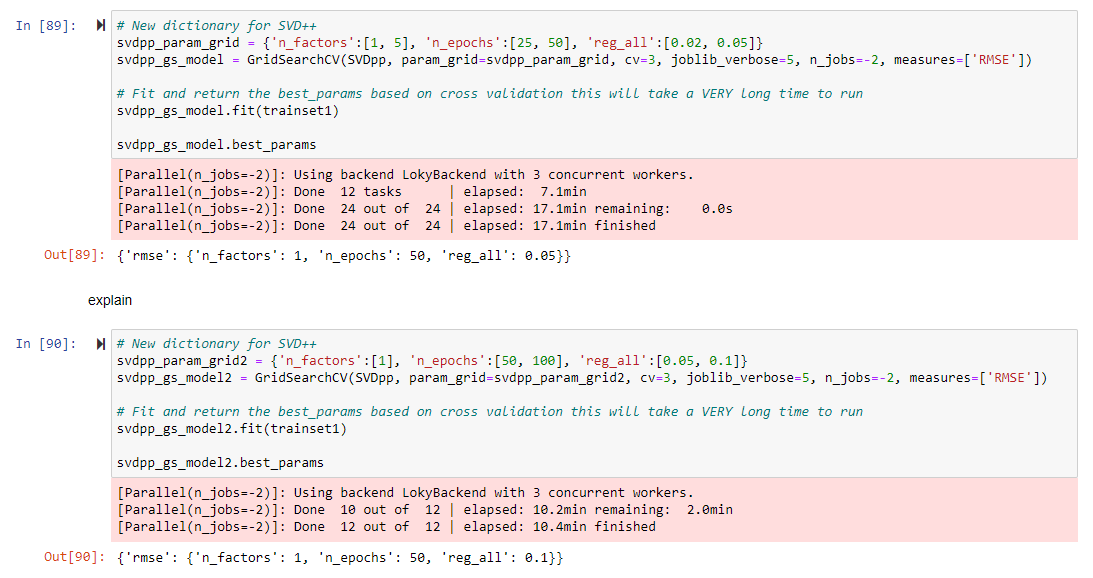
* 1. Kaggle Competition - <https://www.kaggle.com/competitions/h-and-m-personalized-fashion-recommendations>
  2. Article data description:
* article\_id : A unique identifier of every article.
* product\_code, prod\_name : A unique identifier of every product and its name (not the same).
* product\_type, product\_type\_name : The group of product\_code and its name
* graphical\_appearance\_no, graphical\_appearance\_name : The group of graphics and its name
* colour\_group\_code, colour\_group\_name : The group of color and its name
* perceived\_colour\_value\_id, perceived\_colour\_value\_name, perceived\_colour\_master\_id, perceived\_colour\_master\_name : The added color info
* department\_no, department\_name: : A unique identifier of every dep and its name
* index\_code, index\_name: : A unique identifier of every index and its name
* index\_group\_no, index\_group\_name: : A group of indices and its name
* section\_no, section\_name: : A unique identifier of every section and its name
* garment\_group\_no, garment\_group\_name: : A unique identifier of every garment and its name
* detail\_desc: : Details
  1. Customers data description:
* **customer\_id : A unique identifier of every customer**
* **FN : 1 or missed**
* **Active : 1 or missed**
* **club\_member\_status : Status in club**
* **fashion\_news\_frequency : How often H&M may send news to customer**
* **age : The current age**
* **postal\_code : Postal code of customer**
  1. Transactions data description:
* **t\_dat : A unique identifier of every customer**
* **customer\_id : A unique identifier of every customer (in customers table)**
* **article\_id : A unique identifier of every article (in articles table)**
* **price : Price of purchase**
* **sales\_channel\_id : 1 or 2**
  1. Feature Engineering -logic to create a feature -Popularity factor



* 1. Collaborative Filter based method using Surprise Algorithm -evaluation performance:

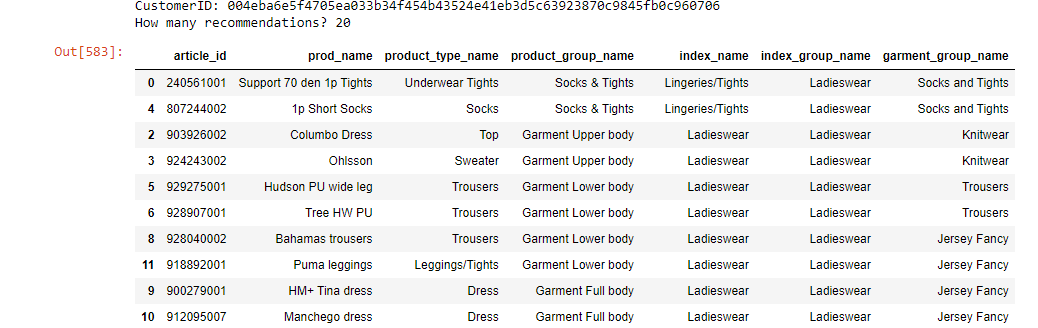


* 1. Collaborative Filter based method using Surprise Algorithm -SVD++ tuning



* 1. Collaborative Filter based method using Surprise Algorithm -Recommendation Example

### Collaborative Filtering using Surprise Package-Example for a Customer



The articles previously brought by the sample customers:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **article\_id** | **prod\_name** | **product\_type\_name** | **product\_group\_name** | **index\_name** | **index\_group\_name** | **garment\_group\_name** | **Article Image** |
| 372860002 | 7p Basic Shaftless | Socks | Socks & Tights | Lingeries/Tights | Ladieswear | Socks and Tights |  |
| 372860024 | Basic 7p Shaftless | Socks | Socks & Tights | Lingeries/Tights | Ladieswear | Socks and Tights |  |
| 875856001 | HM+ House | Sweater | Garment Upper body | Ladieswear | Ladieswear | Knitwear |  |
| 797892001 | ED Primo slacks | Trousers | Garment Lower body | Ladieswear | Ladieswear | Trousers |  |
| 815471001 | SPORT Heaven shape HW tights | Leggings/Tights | Garment Lower body | Ladieswear | Ladieswear | Jersey Fancy |  |
| 294008002 | HM+ Cora tee | Costumes | Garment Full body | Ladieswear | Ladieswear | Jersey Fancy |  |
| 294008005 | HM+ Cora tee | Costumes | Garment Full body | Ladieswear | Ladieswear | Jersey Fancy |  |

Recommendation for the sample customer

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **article\_id** | **prod\_name** | **product\_type\_name** | **product\_group\_name** | **index\_name** | **index\_group\_name** | **garment\_group\_name** | **Article Image** |
| **240561001** | Support 70 den 1p Tights | Underwear Tights | Socks & Tights | Lingeries/Tights | Ladieswear | Socks and Tights |  |
| **807244002** | 1p Short Socks | Socks | Socks & Tights | Lingeries/Tights | Ladieswear | Socks and Tights |  |
| **903926002** | Columbo Dress | Top | Garment Upper body | Ladieswear | Ladieswear | Knitwear |  |
| **924243002** | Ohlsson | Sweater | Garment Upper body | Ladieswear | Ladieswear | Knitwear |  |
| **929275001** | Hudson PU wide leg | Trousers | Garment Lower body | Ladieswear | Ladieswear | Trousers |  |
| **928907001** | Tree HW PU | Trousers | Garment Lower body | Ladieswear | Ladieswear | Trousers |  |
| **928040002** | Bahamas trousers | Trousers | Garment Lower body | Ladieswear | Ladieswear | Jersey Fancy |  |
| **918892001** | Puma leggings | Leggings/Tights | Garment Lower body | Ladieswear | Ladieswear | Jersey Fancy |  |
| **900279001** | HM+ Tina dress | Dress | Garment Full body | Ladieswear | Ladieswear | Jersey Fancy |  |
| **912095007** | Manchego dress | Dress | Garment Full body | Ladieswear | Ladieswear | Jersey Fancy |  |