

**CDS513 PREDICTIVE BUSINESS ANALYTICS**

**SEMESTER 2, ACADEMIC YEAR 2022/2023**

**Assignment 1**

**Walmart Recruiting: Trip Type Dataset – Recommender Systems**

**Name: Looi Kah Fung**

**Matric no: P-COM0049/22**

**SCHOOL OF COMPUTER SCIENCES**

**UNIVERSITI SAINS MALAYSIA**

Table of Contents

[1.0 Background of the problem domain 3](#_Toc135225802)

[1.1 Description of the problem 4](#_Toc135225803)

[2.0 Data understanding & Integration. 4](#_Toc135225804)

[2.1 Exploratory Data Analysis (EDA) 6](#_Toc135225805)

[2.2 Data munging 7](#_Toc135225806)

[3.0 Recommender systems 8](#_Toc135225807)

[4.0 Discussion & Analysis 9](#_Toc135225808)

[4.1 Collaborative Filtering 9](#_Toc135225809)

[4.2 Matrix Factorization – Singular Value Decomposition (SVD) – extension to Collaborative Filtering 11](#_Toc135225810)

[4.3 Content-based recommender system 12](#_Toc135225811)

[5.0 Application of recommender system 13](#_Toc135225812)

[6.0 References: 13](#_Toc135225813)

# 1.0 Background of the problem domain

The Walmart Recruiting: Trip Type Classification challenge involves predicting the type of shopping trip a customer made based on the items they purchased. This problem is significant to Walmart as understanding trip types can help them improve store layout, product placement, and inventory management. By knowing what customers are looking for, Walmart can tailor their product offerings and marketing strategies to better meet their customers' needs.

The dataset provided for this challenge contains many customer visits with the Trip Type included. The challenge is to predict the Trip Type for each customer visit in the test set. The dataset contains several features that describe the customer visits, such as Visit Number, Weekday, Upc, Scan Count, Department Description, and Fineline Number. The Visit Number is an ID that corresponds to a single trip by a single customer, and the Weekday feature indicates the day of the week the trip was made. The UPC feature contains the Universal Product Code (UPC) number of the product purchased, while the Scan Count feature indicates the number of the given item that was purchased. A negative value in Scan Count indicates a product return.

The Department Description feature provides a high-level description of the item's department, while the Fineline Number feature provides a more refined category for each of the products, created by Walmart. The Trip Type feature is the ground truth that needs to be predicted, and it represents the type of shopping trip the customer made.

Walmart has already categorized the trips contained in this data into 38 distinct types using a proprietary method applied to an extended set of data. The challenge is to recreate this categorization/clustering with a more limited set of features. This could provide new and more robust ways to categorize trips.

In summary, the Walmart Recruiting: Trip Type Classification challenge aims to categorize shopping trip types based on the items that customers purchased. This is an important problem for Walmart as it can help them improve store layout, product placement, and inventory management.

## 1.1 Description of the problem

The Walmart Recruiting - Trip Type Classification problem can be solved using machine learning algorithms such as content-based recommender systems or collaborative filtering.

The content-based recommender system approach would involve analysing the product data provided in the dataset, such as UPC number, department description, and fine line number, to create a profile of each customer's shopping history. The system would then recommend products based on the customer's history and preferences.

On the other hand, collaborative filtering approach would involve analysing the patterns of purchases made by different customers to recommend products to new customers. Collaborative filtering can be further broken down into two categories: user-based and item-based. User-based collaborative filtering recommends products to a user based on the purchase history of other similar users, while item-based collaborative filtering recommends products to a user based on the similarity of the items they have purchased in the past.

The objectives of the Walmart Recruiting - Trip Type Classification problem can be listed as follows:

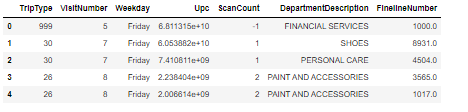
* Analyse the dataset with Collaborative filtering recommender system to predict the VisitNumber based on the ScanCount of other customers with similar preferences.
* Analyse the dataset with Matrix Factorization for prediction ratings and evaluating the model’s performance using accuracy score, AUC score and precision score.
* Analyse the dataset with Content based recommender system to recommend VisitNumber based on their features such as department description to customers who have made similar purchases in the past.

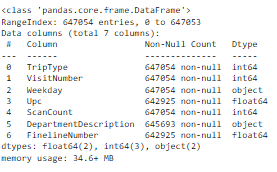
# 2.0 Data understanding & Integration.

The Walmart dataset contains 2 csv files, which are train.csv and test.csv. After a thorough study on both datasets, particularly test.csv does not have trip type information. We have decided not to integrate both files under 2 considerations. Firstly, the torrent of loaded data might get us into memory error, as the recommender system required an expensive computational resource. Secondly, there is no value added when both data have integrated as train.csv can be used to make deduction on the recommender system model and equipped with full fledged of column info.

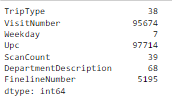
Interpretation of attributes:

1. TripType (association) – a categorical ID representing the type of shopping trip the customer made.
2. VisitNumber – an ID corresponding to a single trip by a single customer.
3. Weekday – the weekday of the trip
4. UPC – the UPC number of the product purchased.
5. ScanCount – the number of the given item that was purchased. A negative value indicates a product return.
6. DepartmentDescription – a high-level description of the item’s department
7. FinelineNumber – a more refined category for each of the products, created by Walmart.

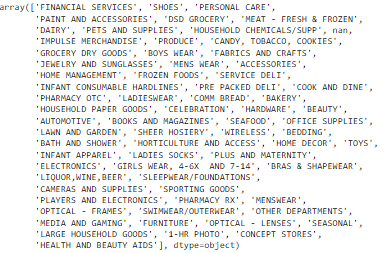




The cardinality and uniqueness of each column info in the dataframe. It helps to understanding the variety and diversity of data present in the dataset. VisitNumber, UPC and FinelineNumber have a high cardinality feature and it provides insights into possibility of getting into cold start when there is no sufficient info of the input. In our case, we are not concerned too much when performing user based collaborative recommender system because of the high cardinality of VisitNumber.

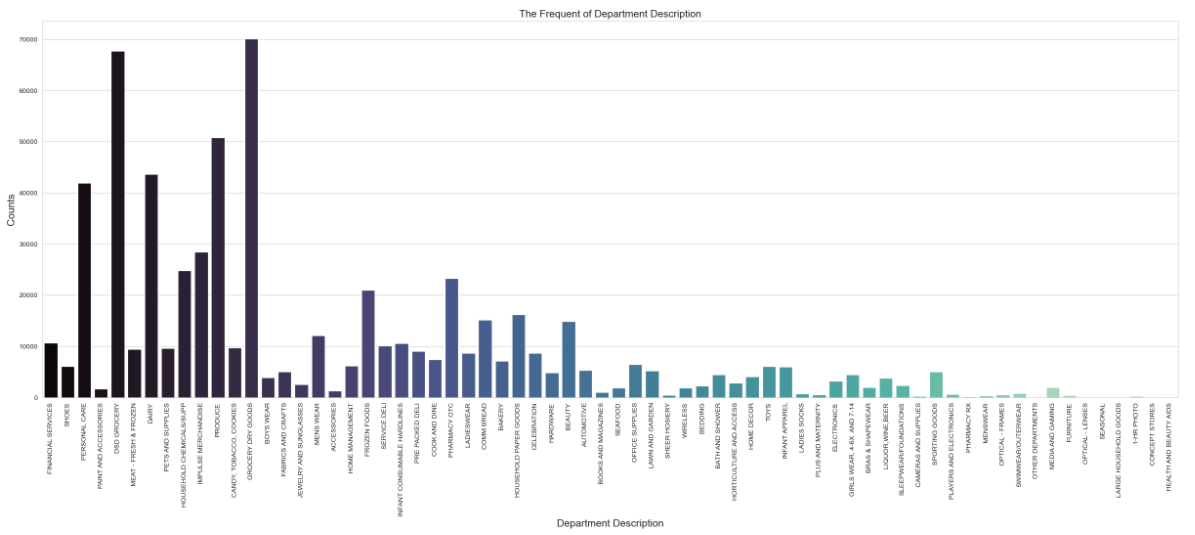


The array of department description. It has 68 unique info. The column info department description represents item in context of recommender systems.

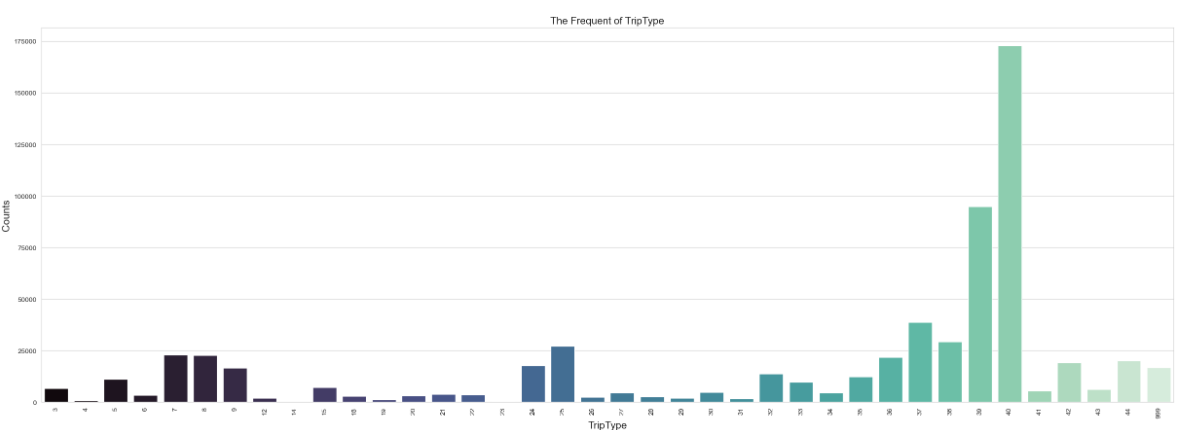


## 2.1 Exploratory Data Analysis (EDA)

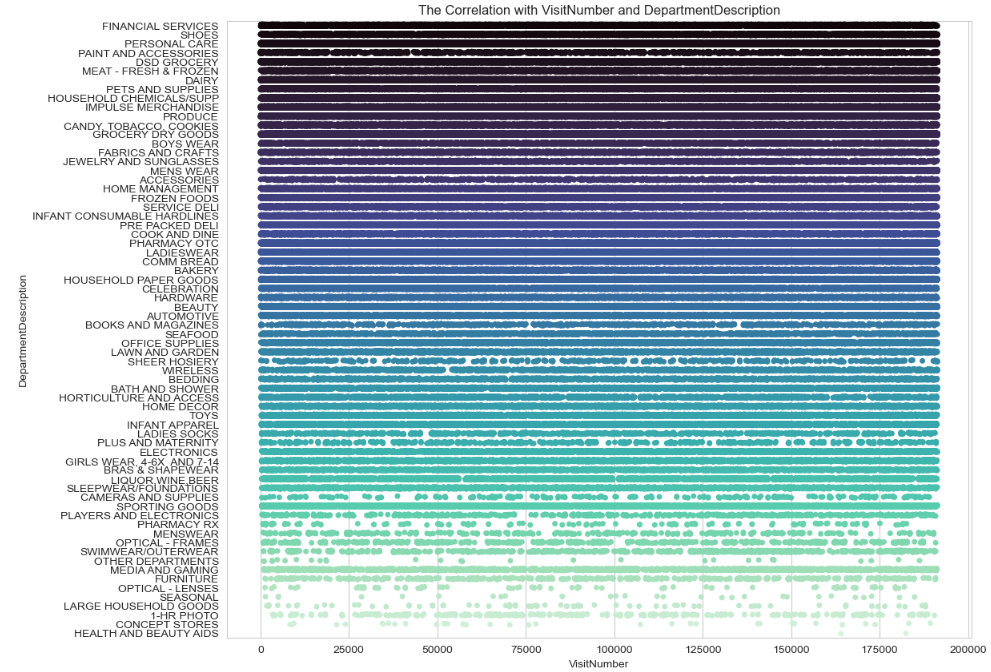
The count plot is used to visualize the frequency of each department description in the dataset. The plot displayed in the EDA using the Seaborn library. The x-axis represents the department description, while the y-axis represents the counts of frequency of each department description. It provides an overview of the distribution of department descriptions and can be used to identify the and least frequency departments. The most frequent departments are centralized at the grocery dry good, dsg grocery and produce whereas the least frequent departments are centralized at seasonal, health and beaty AIDS, concept stores and so on. Based on the distribution, we would anticipate the recommender system to be leaned toward the grocery related department because of the high count and may it take into consideration of VisitNumber and DepartmentDescription matrix.



The count plot is also used to visualization the distribution of the trip type. It shows the highest count of trip type is 40. Trip type is associated with the Visit Number, we can roughly gain insight most of the visit number visited the Walmart under trip type 40.



The strip plot visualization shows the correlation between the VisitNumber and Department Description variables in the dataset. It helps to explore the relationship between the VisitNumber and DepartmentDescription variables, allowing for insights into the distribution and patterns of departments visits.



## 2.2 Data munging

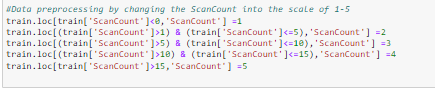
The data preprocessing step above involves changing the values of the 'ScanCount' column in the 'train' dataframe to the scale of 1-5. This is done to transform the explicit feedback provided by customers into a more usable form for building a recommender system model.

The 'ScanCount' column represents the number of times a particular item was scanned in a transaction. By changing its values to a scale of 1-5, we are essentially converting the count of scans into a rating of the item. For example, a 'ScanCount' of 1 is equivalent to a rating of 1 star, while a 'ScanCount' of 5 is equivalent to a rating of 5 stars.

This is useful for building a recommender system model as it allows us to treat the explicit feedback provided by customers as ratings. In collaborative filtering, these ratings are used to build a user-item matrix, where each row represents a user, and each column represents an item. The values in this matrix represent the rating that a user has given to an item.

By converting the 'ScanCount' values into ratings, we can use them to populate this user-item matrix, which can then be used to train a collaborative filtering model. The model can then make predictions on unseen data, by using the user-item matrix to find similar users or items and recommending items to users based on these similarities.







# 3.0 Recommender systems

Building personalized recommender systems plays a critical role in providing customized suggestions for end-users tailored specifically towards their interests’ using techniques like collaborative user-based filtering, matrix factorization and content-based filtering approaches. Each of these approaches has its unique underlying principles and techniques that this discussion aims at discussing comprehensively while highlighting their strengths and limitations. Collaborative user-based filtration approach utilizes the similarities established between different clients by analysing previous patterns of item preferences they have exhibited previously which gives useful insights into predicting future tastes. User-item similarity metrics such as cosine similarity help identify those with comparable inclinations allowing easier recommendation on likes or highly rated items by similar users. This method is outstanding in capturing user preferences that may not be explicitly stated and can provide serendipitous recommendations; however, it faces challenges when dealing with large datasets due to scalability issues. It is also susceptible to cold start problems for new items or users with limited data, and the filter bubble effect wherein users get recommended items like their current preferences, leading to a limited range of exposure from diverse products. [1], [2]

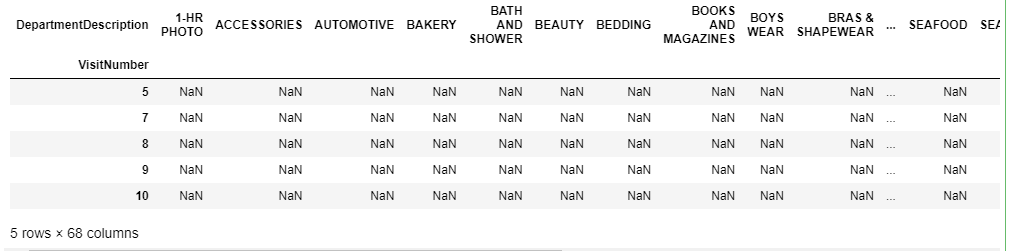
Content-based filtering takes a different approach by considering the features or characteristics of items. It builds item profiles based on attributes such as genre, actors, director, or keywords, and recommends items that are like the user's past preferences in terms of these features. Content-based filtering is particularly useful when dealing with the cold start problem, as it does not rely on historical user-item interactions. It can provide personalized recommendations based on item attributes alone. However, content-based filtering heavily relies on accurate and comprehensive item features, and it may struggle to capture complex user preferences or recommend serendipitous items outside the scope of the available features. [1]

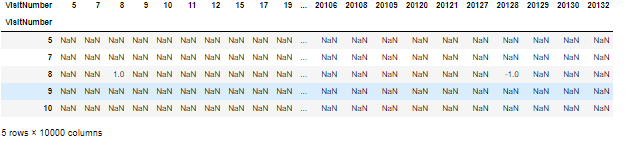
In comparing the three approaches, collaborative user-based filtering excels in leveraging user preferences and generating diverse recommendations. Matrix factorization, with its ability to handle sparsity and capture complex patterns, provides accurate predictions. Content-based filtering, on the other hand, focuses on item attributes and is effective in the absence of user-item interaction data.

# 4.0 Discussion & Analysis

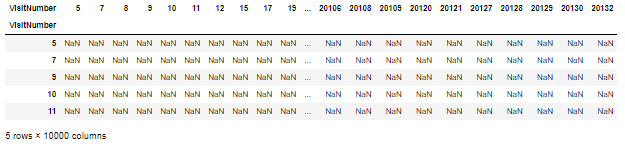
## 4.1 Collaborative Filtering

To create a user-item matrix, we use the pivot table function. The matrix will have VisitNumber as the index, DepartmentDescription as the columns, and ScanCount as the values. The user-matrix need to be normalized to account for variations in VisitNumber and DepartmentDescription popularity. We centre the data around zero by subtracting the row means, representing the relative preference of each user for each item compared to their average preference.

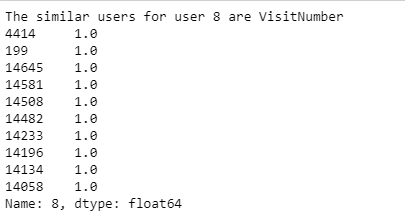




We have selected VisitNumber 8 as the input for the recommendation process. To provide personalized recommendations, we remove VisitNumber 8 from the candidate list. The similarity score is calculated using Pearson correlation coefficient.



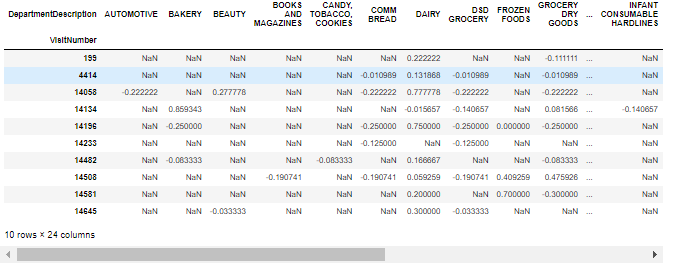
By analysing the user\_similarity, it allows to identify users who are most similar to VisitNumber 8 in terms of the ScanCount. These similar users can serve as a basis for generating personalized recommendations for VisitNumber 8. The outcome shows 4414, 199, 14645, 14581, 14508, 14482, 14233, 14196, 14134, 14058 top 10 similar users for VisitNumber 8 with pearson coefficient of 1.0.



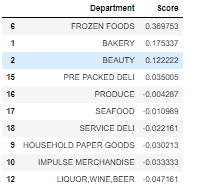
We focus on extracting the department information associated with the VisitNumber 8. The resulting subset provides a focused view of the departments visited or interacted with during VisitNumber 8. We can gain insights into the department that user engaged with during VisitNumber 8. From the dataframe, we can deduce that VisitNumber 8 often visiting dairy, paint and accessories and pets and supplies with a +ve of Pearson coefficient.



To have an overview of the user-item matrix in relation to VisitNumber 8 preferences, we display in a matrix form of VisitNumber as index and DepartmentDescription as column info.

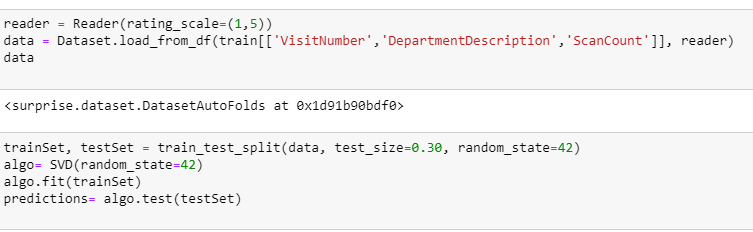


We will decide which department to recommend to VisitNumber. The score is weighted by the similarity scores, so the VisitNumber with higher similarity get higher weights. Frozen foods, bakery and beauty are recommended to VisitNumber 8.



## 4.2 Matrix Factorization – Singular Value Decomposition (SVD) – extension to Collaborative Filtering

Matrix factorization technique such as singular value decomposition (SVD) is used in collaborative filtering-based recommender systems. The collaborative filtering relies on the implicit or explicit feedback of users to make recommender, therefore the ScanCount attribute has been munging into the scale of 1-5. Matrix factorization decomposes the user-item rating matrix into low-rank matrix, capturing latent factors and patterns in the data.



This decomposition allows the model to predict ratings for unseen user-item pairs and make personalized recommendation. The evaluation metrics RMSE and MSE provide insights into the accuracy and prediction error of the SVD-based recommender system. In the context of Walmart dataset, SVD predicted the RMSE value of 0.32 and MAE value of 0.195. It suggests, on average, both metrics with low value signify better accuracy and smaller prediction error. In addition, accuracy score, AUC score and precision score has been calculated to evaluate the model’s performance. The predicted ratings have been binarized using a threshold value of 0.5 because of multi-classification conundrum. Moreover, binarization is necessary because it allows us to assess the model’s ability to correctly classify positive and negative recommendation. The calculated metrics measure the proportion of correct binary predictions and rule to true ratings. After a series of binarization, the model obtained a high score of 0.89, 0.5 and 0.794 correspond to accuracy score, AUC score and precision score respectively.

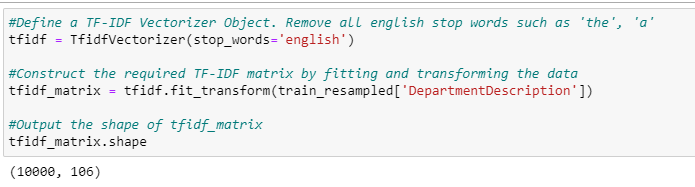
|  |  |
| --- | --- |
| Root mean squared error (RMSE) | 0.3207906748193675 |
| Mean absolute error (MAE) | 0.19590240790147145 |
| Accuracy score | 0.8913815246786606 |
| Precision score | 0.7945610225384536 |

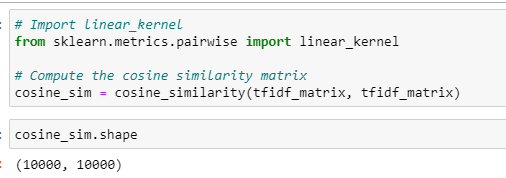
The evaluation metrics such as mean average precision (MAP), Precision at K and Recall at K for a recommender system. It iterates over each user in the testSet and generates top-k recommendations using the trained model. In our case, k = 10, we have done an analysis at 5, 10 and 15. The result does not deviate from a lot. Higher values of MAP, precision at 10 and recall at 10 indicating that the recommender system is effectively generating relevant recommendations for users and the matrix factorization is paired with collaborative filtering.

|  |  |
| --- | --- |
| Precision at 10 | 1.0 |
| Recall at 10 | 0.9915019694862997 |
| Mean average precision | 1.0 |
| AUC score: | 0.5 |

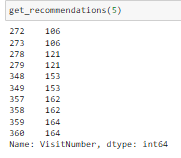
## 4.3 Content-based recommender system

The content-based recommender system analysis focuses on the attribute of the features of department description. To achieve that, the shape of the TF-IDF matrix is outputted, providing an overview of the dimensionality of the matrix. The shape of the TF-IDF matrix reveals the number of department descriptions and the number of unique words in the corpus which is crucial for understanding the complexity and coverage of the recommender system. While the core of the content-based recommender system lies in the calculation of cosine similarity. In this case, TF-IDF matrix is used to calculate cosine similarity to indicating how closely related the department description is based on the wording. Visit number 5 has been chosen to get the recommendation, the recommender model exhibits 106, 121, 153, 162 and 164 visit numbers are closely related to visit number 5 and having similar shopping behaviour.

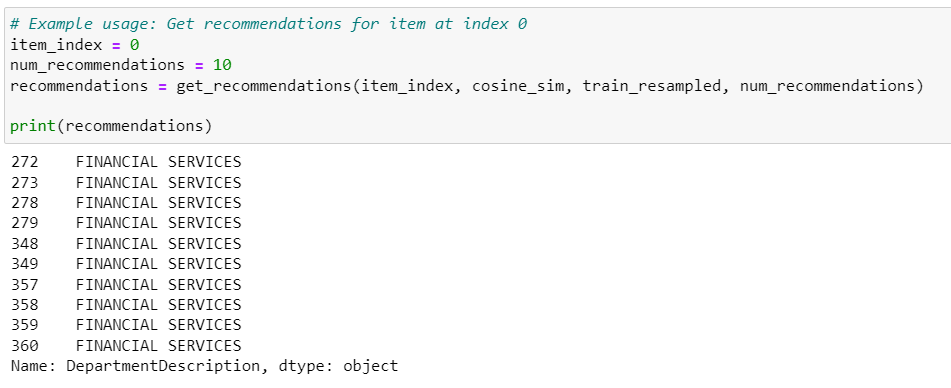




The result is accordance with the similar VisitNumbers. Content based recommender system can be reversed to recommend user-user, in the meanwhile it can be used to recommend user-item as well. It has been displayed in the figure. By having this, we can discover other VisitNumber that have shown similar patterns and what they opt to during the shopping.



The result of the recommendation has taken from the item\_index and num\_recommendations is 10. The resultant is financial services been corresponding to the VisitNumber 272, 273, 278, 279, 348, 349, 357, 358, 359, 360.



# 5.0 Application of recommender system

Our background of the problem domain is centralized at the retailer by recommending the users with the preferences they had liked in the past. In addition, there are other recommendation engine studies in retailer domain. In the context of personalizing item recommendation, Soumya Wadhwa et al. experimented price-related user-item signals into the recommender system to personalize the output. They were using the historical user-item data to predict user price affinity to re-rank the item recommendation anchored on a user preference. That is one of the many applications that using the recommender systems. [3] There are examples of application domain in healthcare, telecom, manufacturing etc. All in all, the choice of recommender system approach depends on various factors such as the characteristics of the dataset, availability of user-item interaction data, item attributes, and the desired recommendation goals. Collaborative user-based filtering, matrix factorization, and content-based filtering each offer unique advantages and limitations. By understanding the underlying principles and analysing their performance metrics, we can make informed decisions when designing and implementing recommender systems to deliver personalized recommendations to users.[3], [5], [6]

# 6.0 References:

[1] F. O. Isinkaye, Y. O. Folajimi, and B. A. Ojokoh, “Recommendation systems: Principles, methods and evaluation,” *Egypt. Informatics J.*, vol. 16, no. 3, pp. 261–273, 2015, doi: 10.1016/j.eij.2015.06.005.

[2] D. Roy and M. Dutta, “A systematic review and research perspective on recommender systems,” *J. Big Data*, vol. 9, no. 1, 2022, doi: 10.1186/s40537-022-00592-5.

[3] S. Wadhwa, A. Ranjan, S. Xu, J. H. D. Cho, S. Kumar, and K. Achan, “Personalizing item recommendation via price understanding,” *CEUR Workshop Proc.*, vol. 2697, 2020.

[4] R. De Croon, L. Van Houdt, N. N. Htun, G. Štiglic, V. Vanden Abeele, and K. Verbert, “Health recommender systems: Systematic review,” *J. Med. Internet Res.*, vol. 23, no. 6, 2021, doi: 10.2196/18035.

[5] J. Lu, D. Wu, M. Mao, W. Wang, and G. Zhang, “Recommender system application developments: A survey,” *Decis. Support Syst.*, vol. 74, pp. 12–32, 2015, doi: 10.1016/j.dss.2015.03.008.

[6] T. S. Kaya, M. Gezer, and S. Gülseçen, “Application of Recommender System for Spending Habits Based Campaign Management,” p. 7, 2021, doi: 10.3390/proceedings2021074007.

Source Code:

1. <https://colab.research.google.com/drive/1cN44RlIEaB28FTD30qFiHkN3rqcDgcng?usp=sharing>

2. <https://www.datacamp.com/tutorial/recommender-systems-python>