

BOĞAZICI UNIVERSITY



**PERSONALIZED PRODUCT RECOMMENDATIONS ON SECOND
HAND PLATFORMS**

Eda Kocakarın, Nilüfer Çetin, Ömer Anıl Usta, Yusuf Uluçoban

Project Advisor: Assistant Professor Mustafa Gökçe Baydoğan

Industrial Engineering Department

June 2022

BOGAZICI UNIVERSITY

**PERSONALIZED PRODUCT RECOMMENDATIONS ON SECOND
HAND PLATFORMS**

Submitted as the Final Project Report
for the Degree of Industrial Engineering
of Boğaziçi University

by

Eda Kocakarın, Nilüfer Çetin, Ömer Anıl Usta, Yusuf Uluçoban

Industrial Engineering Department in Engineering Faculty

June 2022

Abstract

Consumers have been switching to shopping from online platforms for quite some time now as more and more people found out they can reach to wider range of products for more affordable prices with fast delivery options. One trend in e-commerce is the second-hand platforms. These platforms present an environment for people to sell their items to other users. Inherent issue with these platforms is the difficulty in finding a product with suitable style and size because of the uniqueness of each product. Recommendation systems are being utilized in different platforms. These engines could optimize the experiences of users in a second-hand platforms by suggesting the most suitable items to the user. There are some obstacles related to the recommendation engines as well as the problems stemming from the nature of the second-hand platforms. This study aims to implement a recommendation engine for a second-hand platform to achieve increase in user-item interaction volume. Building and testing the approach, historical click-stream data of users have been utilized. Item-to-Item and Model Based Collaborative Filtering methods are used in the design of the proposed recommendation engine which is composed of two stages: Candidate Generation and Candidate Ranking. Three different models are presented and evaluated via different performance metrics. At final, the model with better performance and higher explainability has been proposed for implementation in the live ecosystem of the second-hand platform.

Özet

Git gide daha fazla kullanıcı, daha geniş ürün yelpazesine daha uygun fiyatlarla ve hızlı teslimat seçenekleriyle ulaşabileceğini düşünerek çevrimiçi platformlardan alışveriş yapmaya geçiş yapmaktadır. E-ticaret olarak adlandırılan bu alanda bir başka trend ise İkinci El Platformlar olarak görülmektedir. Bu platformlar, insanların eşyalarını diğer kullanıcılara satmaları için bir ortam sunmaktadır. İkinci El Ticaret Platformlarındaki temel sorun, her ürünün tek bir örneği olması nedeniyle uygun stil ve boyutta bir ürün bulmanın zor olmasıdır. Öneri motorları, farklı alanlardaki platformlarda kullanılan bir teknolojidir. Bu sistemler, kullanıcıya en uygun öğeleri önererek İkinci El Platformlarındaki kullanıcıların deneyimlerini optimize etmek amacıyla da kullanılabilir. Bu noktada, İkinci El Platformlarının yapısından kaynaklanan sorunların yanı sıra tavsiye motorlarının oluşturduğu kimi engeller de bulunmaktadır. Bu çalışma, kullanıcı-öğeler etkileşim hacminde artış sağlamak adına İkinci El bir platformda çalışacak bir öneri motoru üretmeyi amaçlamaktadır. Yaklaşımın oluşturulması ve test edilmesi sürecinde kullanıcıların geçmiş tıklama akışı verileri sıklıkla kullanılmıştır. Aday Oluşturma ve Aday Sıralaması olmak üzere iki aşamadan oluşan önerilen öneri motorunun tasarımında Öğeden Öğreye ve Model Bazlı İşbirlikçi Filtreleme yöntemleri kullanılmıştır. Sonuç olarak sunulan üç farklı model farklı performans ölçütleri göz önünde bulundurularak değerlendirilmiştir. Son olarak, İkinci El Platformun canlı ekosisteminde uygulanması için daha iyi performansa ve daha yüksek açıklanabilirliğe sahip model önerilmiştir.

Contents

Abstract	i
Abstract in Turkish	ii
Contents	iii
List of Tables	v
List of Figures	vi
1 Introduction	1
1.1 Summary	1
1.2 Overview	3
1.3 Improvements Expected by the Study	4
1.4 Summary of Conclusions	5
2 Detailed Overview of the Problem	7
2.1 Problem	7
2.2 Causes of the Problem	8
2.3 Requirements	10
2.4 Limitations	12
2.5 Data Gathered for Identification	13
2.6 Context Diagram	16
2.7 Performance Criteria	17
3 Analysis for Solution and Design	19
3.1 Literature Overview	19
3.2 Alternative Approaches	20

3.3	Assumptions.....	22
3.4	Summary of Selected Approaches	23
3.5	Relation to Industrial Engineering	24
4	Development of Alternative Solutions	26
4.1	Stage 1 - Item to Item Collaborative Filtering	26
4.2	Stage 2 - Candidate Ranking based on Scoring	30
5	Comparison of Alternative Solutions	46
5.1	Evaluation Procedure and Results.....	46
5.2	Proposed Model.....	51
5.3	Assessment of Solution	52
6	Suggestions for Implementation	54
6.1	Implementation	54
6.2	Integration	55
6.3	Revision of the Model	55
7	Conclusion and Discussion	57
7.1	Use of Industrial Engineering Concepts	57
7.2	Merits and Significance.....	58
7.3	Impacts of Design.....	59
8	Appendices	60

List of Tables

8.1	Color Group Mapping	60
-----	---------------------------	----

List of Figures

1.1	E-Commerce Sales in Retail Percentages	1
1.2	A sample Page from E-Bay	2
1.3	A Page from Dolap with Ranking Algorithms.....	3
2.1	Product Page and Recommendations.....	8
2.2	Sources of Clicks in Gathered Data	9
2.3	Several Categories in Dolap	11
2.4	A Snapshot of Color and Brand Catalog Data	14
2.5	Different Actions in Application	14
2.6	A Snapshot of Interaction Data.....	15
2.7	Histogram of Various Interactions in a Month.....	15
2.8	Context Diagram	16
2.9	One Proposed Evaluation Metric	18
4.1	Image showing Matrix Approximation in Collaborative Filtering from Project Pro: Recommender Systems Python-Methods and Algorithms	27
4.2	Alternating Least Squares Algorithm by Ghosh et al	28
4.3	Ratio of Likes, Bids and Comments Preceding Purchase	29
4.4	Histogram of Paid Amount	34
4.5	Sigma Limits in Normal Distribution from MathBitsNotebook	35
4.6	Distribution of Log Transformed User-Item Scores in Training Set.....	39
4.7	Mechanism of Tendency Model	39
4.8	Mechanism of Linear Regression Model	40
4.9	Change in Adjusted R-Squared Values in Training	41
4.10	Summary of Linear Regression Models	41
4.11	Change in RMSE Values in Training.....	42
4.12	Histogram of User-Item Interaction Scores	43

4.13	Change in Accuracy Values in Training.....	44
4.14	Summary of Logistic Regression Models.....	44
5.1	Snapshot demonstrating the Sparsity in Test Data.....	47
5.2	Rank Correlations for 5 Rankings	48
5.3	A Successful and Unsuccessful Set	49
5.4	The Percentage of Successful Product Recommendations	49
5.5	The Percentage of Products Interacted	50

Chapter 1

Introduction

1.1 Summary

Trading goods, services and monetary assets through internet is named as e-commerce [1] and is gaining popularity day by day. The percentage of online sales in total retail sales has seen a jump and is expected to be around 22% at the end of 2022 as forecasted by Statista.

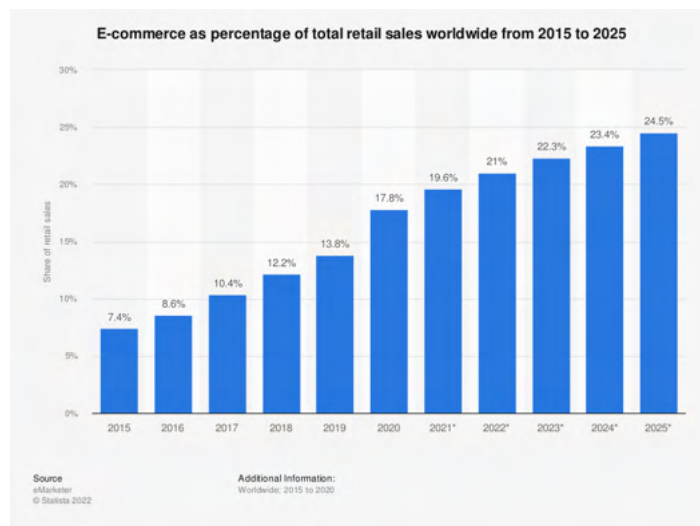


Figure 1.1: E-Commerce Sales in Retail Percentages

Baubonienė et al. (2015) [2] found out that people are opting to shop online for various reasons such as finding cheaper options of a wider range of products more easily and quickly.

A new trend in e-commerce is the second-hand platforms which are one of the commonly known examples to consumer-to-consumer (C2C) marketplaces in the e-commerce business models list of Nermat (2012) [3]. E-bay, Taobao and Dolap can be given as examples to the second hand e-commerce platforms. Cheaper prices, environmental concerns, more original options and feelings of refusal to become a part of capitalist system are found to be some of the causes for consumers to shop from second hand platforms according to Roux et al, 2008 [4].

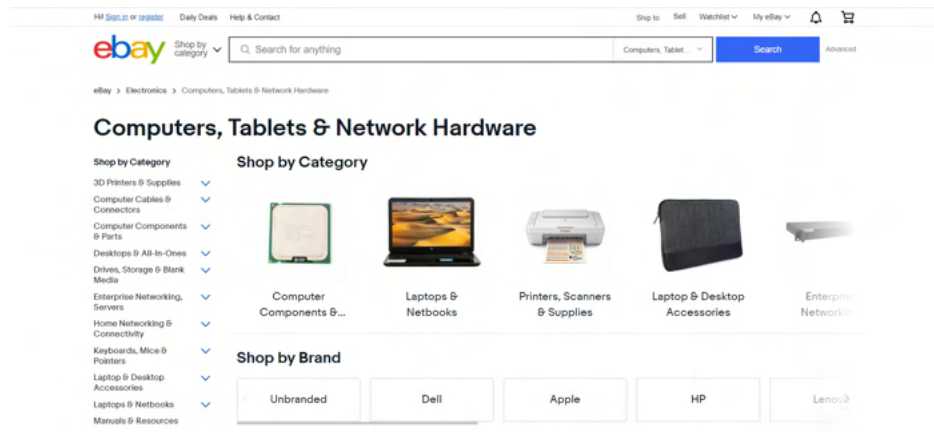


Figure 1.2: A sample Page from E-Bay

The main output of this study is a recommender system which is tailored for personalization using big data. It can be stated that personalization is now a vital part of various strategies utilized by e-commerce platforms [5]. With the number of interactions and transactions conducted through internet skyrocketing, more data related to consumer behavior is readily available. One way of personalization using the big data is presented can be regarded as recommender engines. These systems are frequently made use of in mail offerings, movie, song and other such media entertainment engines as well as e-commerce platforms.

The aim of this study is to devise a personalized recommender system for a second hand e-commerce platform in Turkey, namely Dolap. The final model will be able to recommend a number of available products under product pages according to past user behavior and preferences based on the predicted likelihood of user-product interaction. For this reason, click-stream data entailing user's browsing and event history on the platform will be used along with information on product's condition, color and brand as provided by the seller. Since massive data is included, only two sub-categories of product data of interactions of

users with products in a one month time horizon is gathered. The data is exploited through a series of data mining and mathematical models commonly used in the literature and final models trained with suitable hyper-parameters are presented and compared based on several performance criteria developed.

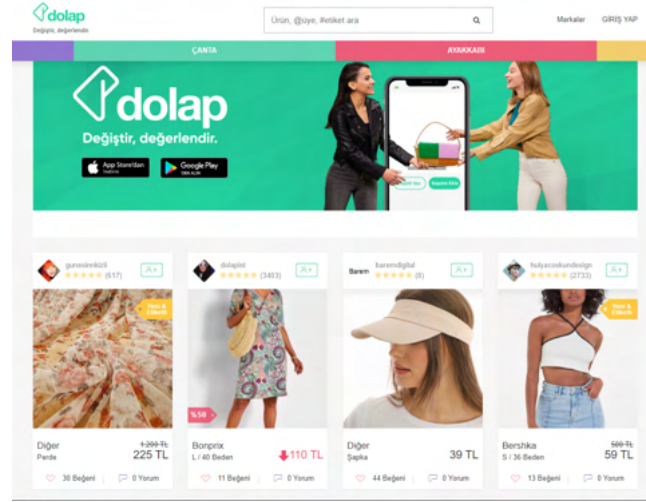


Figure 1.3: A Page from Dolap with Ranking Algorithms

1.2 Overview

There are two main categories of recommender systems that are utilized heavily in recent e-commerce and media platforms. The first category is termed as Collaborative Filtering Methods including a wide range of statistical and algebraic models on which the original implementations of recommender systems are built. The other partition can be demonstrated as the Neural Network Based methodologies which are novel and designed utilizing a series of neural network and back-propagation models.

Memory based collaborative filtering methods include product-user vectors, matrices and algebraic methods such as principal component analysis to yield a value quantifying the user's possible interest in the item which is then converted into a probability taking confidence and preference metrics into account as tried out by Hu et al. (2008) [6]. Furthermore, these methods are able to calculate and report on the similarity of different products and users based on the scoring methods to be used in filling product-user matrices.

Wang et al. (2021) [7] asserts that memory based approaches face with problems related to very low number of interactions between user and item pairs especially for new users

and products. Consequently, model based approaches for recommender systems have been gaining popularity as well. Trying to find a suitable distribution of user-product interaction probability based on user-item features to be extracted, some of these methods utilize machine learning methods as regression, clustering, naïve Bayes and ensemble trees [8] others exploit graph approaches and horting [9]. Though more robust and easily interpretable, these methods are subject to issues stemming from the vast number of computations for large user-item sets and over-specialization with elevated number of features to be extracted.

With the increased computational power, the use of Neural Networks in the field of recommender systems has also become popular. Among these, Covington et al. (2016) [10] has utilized two neural networks for candidate generation and ranking in Youtube video recommendations. Furthermore, Wang et al. (2021) [7] have embedded user and item features using deep learning approaches to overcome the problems of traditional collaborative filtering.

The progress of Natural Language Processing methodologies contributed to increase in the use of originally text-oriented approaches in product recommendation. Utilizing the Continuous Bag of Words and Skip Gram methods introduced by Mikolov et al. in 2013 [11] to model browsing history of customers as sentences with words denoting to products, more refined models specifically tailored for product recommendation tasks in e-commerce platforms have been offered by the works of Pfadler et al. (2020) [12] and Chen et al. (2019) [13]. Nonetheless, such models are not interpretable which decreases the generalization and flexibility when in use.

1.3 Improvements Expected by the Study

The range of studies utilizing click-stream data for recommendation engines are quite limited especially in the context of second hand e-commerce platforms. While this is the case, most of such studies deal with either incorporating more information into memory based collaborative filtering models or developing advancements on the neural nets or natural language processing models.

With these limitations being stated, considering the academic aspects this study is conducted in an aim to propose a well-developed exploitation of click-stream data via statistical data mining models in the context of recommender systems in second hand platforms. Furthermore,

the range and meaning of attributes that will be embedded into the model based collaborative filtering methods are all explainable, statistically significant and carefully selected by taking user and category trends as well as economical and aesthetic preferences into account. Lastly, the work is robust in the sense that it is one of the few number of studies in category to propose several two-staged models and compare them based on several criteria.

From an industrial point of view, with subsequent improvements to the proposed model and generalization into different categories, the recommender system can be employed by Dolap to personalize and thus optimize the user experience according to past user behaviors. It is expected for customers to spend more time in a platform which can understand the user needs and answer them readily in no time with minimal user effort devoted, while the sellers can benefit from reaching potential buyers in smaller time in the comfort of their homes. The user satisfaction can bring increased amount of traffic and purchases in the system in the form of expanded number of buyers and sellers who are increasingly loyal.

1.4 Summary of Conclusions

To sum up, it can be stated that the recommender systems have the potential to be adapted to wide range of platforms from different domains exploiting the advantages of big data and increasing ability of computation in the form of robust and flexible models.

Though past purchasing behavior of the users is frequently incorporated into recommender engines in e-commerce market places, the uniqueness of products impairs the strength of such approaches to be adapted in second hand platforms. Instead with limited information on user demographics, click-stream data in the form of liking, bidding, visiting and commenting on products become much more vital to be used in analyses and the models. Nonetheless, this data includes huge amount of information and is almost always highly unbalanced and sparse. For further clarification, it can be stated that there are few users with too many interactions, some with enough number and a large number of customers with no to too little interactions within the platform making it a hard problem to derive conclusions about user behavior and user-item suitability.

Recommendation engines in e-commerce platforms is a topic that is becoming known and studied by more recently. It is approached as a statistical modeling and learning problem due

to vast amount of data being utilized in the process. Furthermore, it can be claimed that there are different models with advantages and disadvantages. In the light of these information, several approaches will be devised on the upcoming chapters of this study.

Chapter 2

Detailed Overview of the Problem

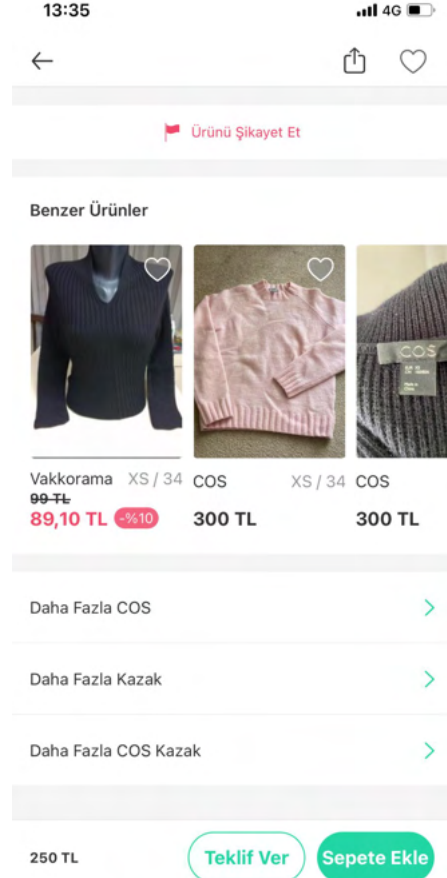
2.1 Problem

The purpose of this study is to devise a personalized recommender system in a second hand e-commerce platform that can suggest additional products fitting the user needs and taste under the product pages that the user visit. With this aim being stated, this study incorporates a number of Industrial Engineering related topics from mathematical modeling, statistical learning models, management and exploitation of big data.

There are certain problems recommender systems operating in different platforms have. A part of these problems arise from the amount and ambiguity of the available information which are related to massive data, some stem from the computational complexity of suitable mathematical and statistical methodologies that can yield improvement and some are domain specific to second hand e-commerce platforms.



(a)



(b)

Figure 2.1: Product Page and Recommendations

2.2 Causes of the Problem

Pfadler et al. (2020) [12] states that recommender systems for e-commerce platforms are subject to certain limitations such as cold-start, sparsity and scalability. Cold start problem refers to the fact that there is only limited amount of data about a product or a user at the beginning which impairs the effectiveness and accuracy of prediction models. As the number of items and consumers on e-commerce platforms is huge, it is almost always the case that most users do not have any interaction with most of the products. As a consequence, there might be inadequate information on product-user interaction database and such matrices are usually highly sparse to construct an all-inclusive recommender model. Finally, the huge amount of data of users and products puts a strain on managing possible models termed as the problem of scalability.

Tailoring a recommender system specifically for a second-hand e-commerce platform makes the problem unique as well. Since the business model is customer-to-customer, most of the products offered in second-hand platforms are singular. This situation requires elimination of certain models that segment the users and offer products based on buying history of the segment. With one of the most powerful indicators of preference being partially left out, the importance of other interactions in click-stream data can be made clear. Furthermore, with the unique products, the items suitable for the user is limited to a certain extent. For example size is an important determinant for a consumer to buy clothing. Nonetheless, since only one size is available of each item in second-hand platforms, the number of items that can be recommended to the user is substantially decreased. Furthermore, specific to this platform, there is already a recommendation engine operated by the team behind the application. Existence of such engine may have created a bias for the past user interactions that are used in training as well as model evaluation stages where the proposed recommender systems are tested based on past user-item interactions.

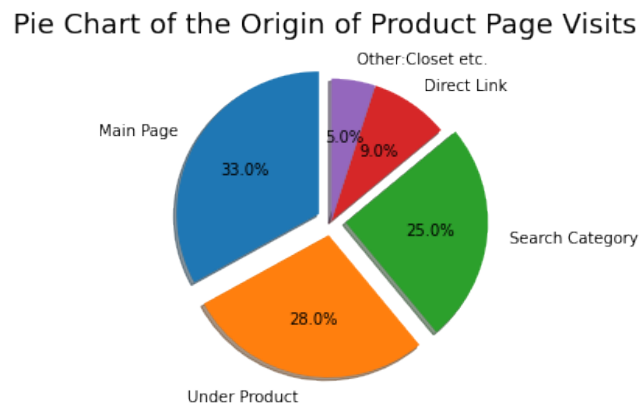


Figure 2.2: Sources of Clicks in Gathered Data

The data left by the user on e-commerce platforms may provide insight about user preferences and can be classified into two categories, explicit and implicit feedback data. The click-stream data obtained for this study include bidding, liking, commenting, viewing, visiting and buying products found in the second-hand platform and most of these actions can be categorized as implicit feedback. Nonetheless, there are certain drawbacks of using the implicit feedback data. Hu et al. (2008) [6] argues that implicit feedback data is non-negative, noisy, requires its own scaling and differs from the explicit feedback data in the sense that numerical values yield confidence but not preference in beliefs.

These aspects can be explained in more detail. As can be understood from the events included, the implicit feedback do not give any information about user distaste. In other words, it cannot be understood whether a consumer is not interacting with certain products because of dislike or other factors such as those products not being offered readily to the consumer. For similar reasons, implicit feedback data is noisy. Since the observer does not have any clue about the motivation of the user to interact with specific products, it cannot be argued that every interaction indicates preference. For example, the consumer might have visited a product page because of a simple clicking mistake. The fact that only 2% of e-commerce visits turn into purchases can make it easier to illustrate this problem. Moreover, implicit feedback data is not standardized compared to rating data. To give a concrete example, the thresholds for consumers to press the like button is usually different from one another. Thus, it could be the case that both users might have liked the same product while one liked it so much that he/she is planning on buying and for the other, the product is slightly above his/her average liking threshold. Consequently, handling data on implicit feedback actually calls for merits. Lastly, the numerical values in click stream data usually indicate frequency such as the total number of times a product page is visited by a user. Increased frequency in this case means more confidence in the hypothesis that the product is liked by the consumer but there is still no data providing evidence on the amount of this liking [6].

Finally the amount of traffic, data and monetary value being transacted makes this problem a special one under the category of Industrial Engineering problems. E-commerce sales revenue is expected to increase from \$4.28 trillion in 2020 to \$5.4 trillion in 2022 leaving the annual nominal GDP of \$4.87 trillion of Japan behind, which has been the 3rd country with biggest economy in 2017 according to Worldometers. Furthermore, Statista estimates that there should have been 2.14 billion people around the world buying products and services online which makes nearly 30 percent of world population and around 46 percent of world inhabitants that have access to internet.

2.3 Requirements

The system to be proposed will be utilized by the Dolap team for the active use of consumers in the second-hand platform. Though the consumers will be interacting with the engine,

sellers can also benefit in the form of increased purchases and revenue. Thus, the users in platform, platform managers and the group that conducts this study can be summarized as the parties that are of interest for the engine.

First of all, once put online the Dolap team will be responsible for the maintenance of the engine. Maintenance may include running models regularly with new data, gathering and evaluating data on engine-user interactions and testing for and fixing any faulty or inadequate behavior. For these reasons, the team demanded the models to be scalable and easy to interpret. Since the amount of user and item into and out of the application is a lot, the team would like to keep the engine updated in rather small periods. As a result, it is expected that the data manipulation and training phase of the engine should be short. Furthermore as the team will be responsible for the further developments of system, they desire to operate with an engine that is built on easily understandable models. Lastly the generalization power of engine is also important for the Dolap team. Though the models will be constructed with data from two categories, the system will be working under many categories and product pages. Hence, it should be paid close attention to not train the models with data or insights that are specific to these two training categories namely the women bag and the women pullover.

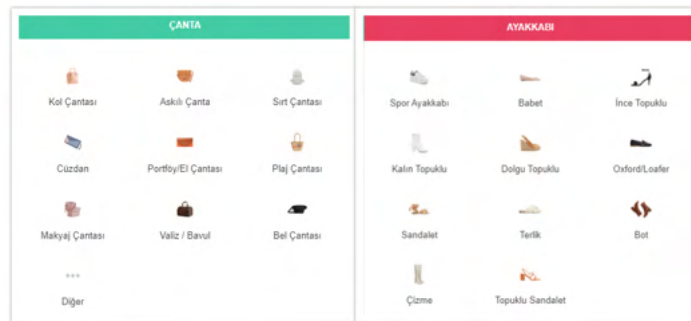


Figure 2.3: Several Categories in Dolap

Secondly, for the users the smoothness in interaction with recommender system is vital. The engine should be designed in a way that requires as little as possible tasks from user such as filling in a survey or rating an item and the product recommendations should be existent under all of the product pages in no time with maximum accuracy. It can be regarded as easy to require minimum from the user, as the data that is fed into the models encompass the actions taken by users during a normal session. To recommend items in almost no time calls for a model that is trained fully or to the most before the user-item interaction, for this

purpose a method to incorporate user-item interaction into the model at last point in minimal time may be developed.

Finally, it should be noted that this study is conducted as a final project for Industrial Engineering bachelor's degree. This implies that the methodologies and models to be utilized should be related to subjects such as mathematical modeling, data mining and statistical model interpretation. Originating from computer science domain, the previously mentioned neural network based strategies have little correspondence to Industrial Engineering context. Consequently, such strategies will be disregarded at least during the main system development process.

2.4 Limitations

A recommendation engine is an intangible mathematical model in operation compared to other possible models and systems that can be constructed within Industrial Engineering context such as distribution networks, facility layouts and queuing systems. As a result, the physical and geopolitical limitations imposed by this study are almost non-existent.

With that being stated, there can be listed a number of legal and ethical issues that this study is subject to. The click-stream data encompasses the actions taken by users while browsing through the application. Collection of such data requires consent of the user which is taken whence the user starts using the application. Though the consent is obtained by the platform, there does not exist any other personal data such as the actual name, gender, age and place of the user. As can be understood, the anonymity of the users is ensured. Furthermore, even the usernames are kept confidential by the platform and are provided as user ids for this study. Hence, it is impossible to match the data to any actual identity and track for further user actions.

Secondly, the use of gathered data for any other purposes with third parties is strictly prohibited by the legal agreement signed by both parties beforehand. Publication of the data in any platform is illegal and the data will not be accessible by study conductors after the end of the project duration without the consent of Dolap team.

Environmental effect of the computational costs in the form of energy to be spent during

updating the models with new data should also be paid special consideration. It is of the researcher's responsibility to decrease the number of unnecessary computations in the final stage and minimize the size of the recommendation model to adequate measures. Nevertheless, as will be presented in detail the training of user-item matrices in first stage of models are optimized by a function and a library in Python and takes very minimal time to handle. However, it is possible with more users, items and categories to be incorporated, it may require more time and energy to build and update the models. Nonetheless, as stated the methods are chosen with regard to minimizing the computational costs and time to train models.

A final limitation specific to this study, has been the lack of time and means to test the proposed models in the platform. Normally for the projects that are designed for industrial use, online A/B testing is a stage that yields important results to evaluate the models and decide on further improvements. On the other hand, because of limited time and the academic nature of the study purpose, this evaluation has not been possible. As a result, coupled with the limited number of active users and popular products; the designed methods have only been evaluated on a selected portion of past user and items. Consequently, it can be stated that the conclusions are more suitable for active users and frequently visited products while testing the models on a wider range of user and product types might yield more useful information and enable a more comprehensive evaluation.

2.5 Data Gathered for Identification

For model building purposes click-stream data including events such as bidding, commenting, liking, visiting page and viewing products is provided by Dolap team. Furthermore, to more closely grasp the user behavior, purchasing data has also been gathered. Lastly, to help with feature extraction and manual testing, information on products in the form of coloring, conditions, brands and brand segmentation are also taken.

id	title	colour_id	title
404650	Berakids	47	Mürdüm
368900	Smafolk	44	Somon
403050	Egos	49	Şeffaf
171800	Bilgi Yayınevi	-4	Renksiz
386850	PatPat	12	Altın
386851	la&vetta	50	Zebra
383751	Visco House	51	Puantıye
322550	Arma House	40	Dore
397950	Nas Bag	-4	Renksiz
393801	Electro-Harmonix	-3	Ekoseli
306600	Casilda Home		
304552	Bose		
338700	Stone Island		

Figure 2.4: A Snapshot of Color and Brand Catalog Data

Since the platform is widely used, the amount of daily and even hourly traffic is massive. As a result, taking data from all shopping categories could result in a size that is impossible for non-commercial computers to handle. For that reason, the problem is partitioned on the basis of recommending products from the same sub-category to the one the user is viewing. Consequently, the scope of the study is limited to two sub-categories. Nevertheless, it is believed that the final models are explainable which makes them easier to be generalized into other sub categories with minimal effort.

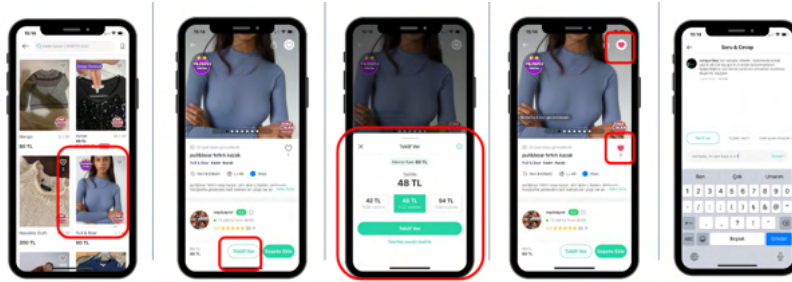


Figure 2.5: Different Actions in Application

It has been demonstrated that most of the interactions in application occur under woman clothing categories. Taking this into account, from the popular categories women pullover, a seasonal item, and women bag, a non-seasonal item, are selected as of interest. The click-stream, purchasing and product data from these two categories spanning one month from February 20_{th} 2022 to March 20_{th} 2022 are gathered.

ts	channel	event	event_subtype	pid	sid	ip	x_id
1 2022-02-20 00:00:00 UTC	iphone	pageview	product_detail	79655531-D44C-40B8-B7A3-109558A57F13	C3706F12-A907-4CF2-B620-0CA4449433A8	37.154.136.194	ff605e4-5212-4acf-83f3-b97502048047
2 2022-02-20 00:00:00 UTC	iphone	pageview	product_detail	7CE4852D-18BD-43E7-9F3A-834A64F8780E	AF1800DA-B40F-4E44-8BC2-C784A708FC35	88.238.138.187	66f47ec3-16e9-4acf-a5a4-06428030e1d4
3 2022-02-20 00:00:00 UTC	android	pageview	product_detail	275dab75-03df-44fc-afe9-175854262234	f295569-1a4a-4a11-89a3-f2cacc6a4a3	88.231.130.128	096c225d-4a6e-4f22-9883-ade360ef2cd6
4 2022-02-20 00:00:00 UTC	iphone	pageview	product_detail	622C9B51-688F-4D88-BE77-6E26A6C091	A7E4283D-8904-48BF-A3B3-38547703FAA4	176.33.71.178	8634a565-6cbc-490e-9661-c31e8da0067

productid	sellerid	shipmentterm	brandid	categoryid	colorid	condition	hasbidding	myproduct	price	productgroup	productstatus	quality	sellertype
172337378	39232749	BUYER_PAYS	214	96	1	GENTLY_WORN	N/A	N/A	45	10	APPROVED	LOW	USER
146599219	13524849	BUYER_PAYS	339	96	34	NEW_WITH_TAGS	N/A	N/A	160	10	APPROVED	MEDIUM	USER
181534443	39097001	BUYER_PAYS	214	96	16	GENTLY_WORN	N/A	N/A	45	WOMAN	APPROVED	LOW	USER
81327910	9226131	BUYER_PAYS	107	96	8	LIKE_NEW	N/A	N/A	26	10	APPROVED	LOW	USER

Figure 2.6: A Snapshot of Interaction Data

As a next step, the browsing habits of users are analyzed. It is found out by summing the average number of times of each interaction that on average a user takes 24 actions in the application within a month including bidding, comment, liking, product visit or ordering. Four of these interaction histograms can be seen below. Due to the fact that recommender systems require data about user preferences, the users with too few data, those that took less than 8 actions in the two categories within this one month are eliminated at the very first step inspired by the method proposed in Yarar (2022).

Various Histograms of User Behavior in a Month

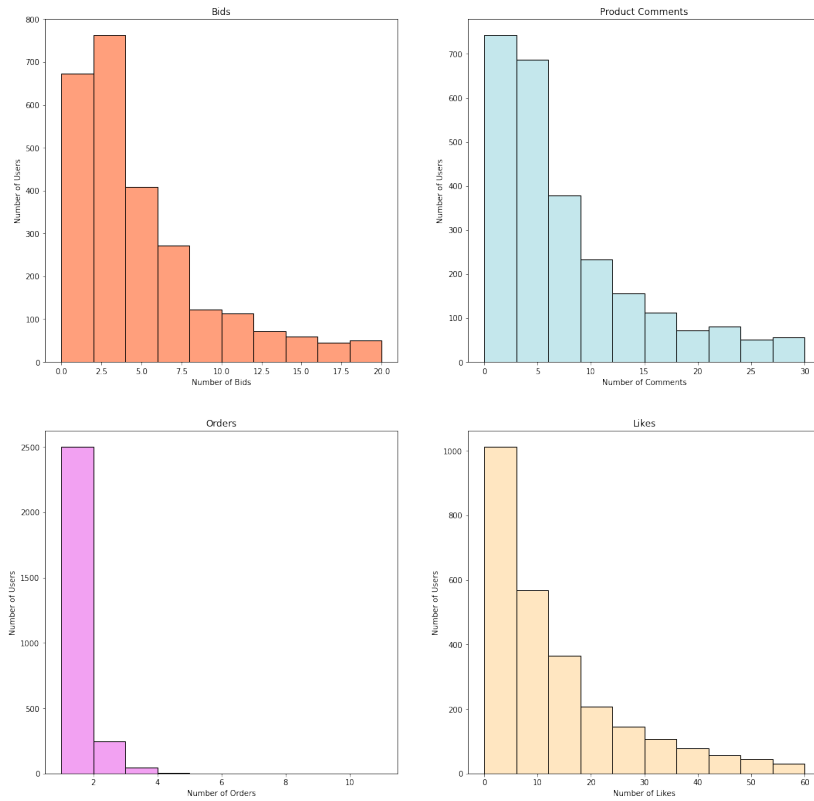


Figure 2.7: Histogram of Various Interactions in a Month

2.6 Context Diagram

From the systems perspective, there can be listed three different parties that have relation to the recommender system to be produced namely the users of Dolap application, the Dolap team and conductors of this study. The possible interactions that the parties can take are demonstrated in the context diagram.

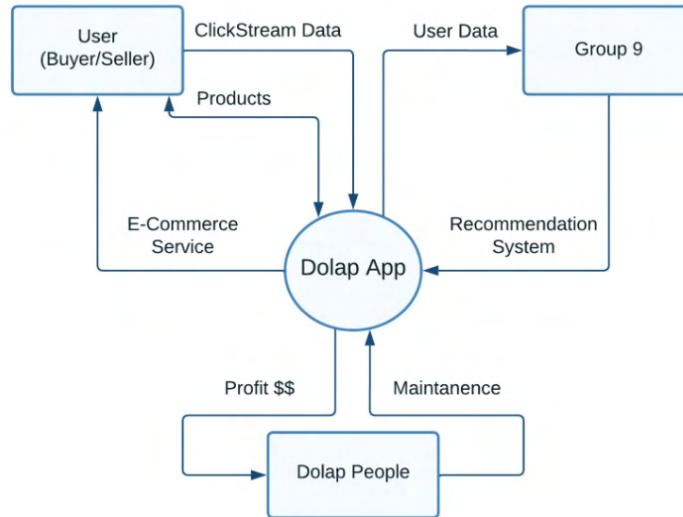


Figure 2.8: Context Diagram

Dolap application provides users with a platform where they can trade products as both sellers and buyers. Thus, it can be stated that the product catalogue is actively updated by the users in form of either product listings or purchases. Not only the purchases, but all the other actions taken by the users within the application are collected and named as the click-stream data.


In this study, the click-stream data is gathered, analyzed and exploited to effectively build a recommender system to operate in Dolap application in a way that personalizes product listings under the product pages. The application team is responsible for the maintenance of the platform as well as the recommender engine to be built.

2.7 Performance Criteria

It is ideal to test the models to be proposed in operation in the actual Dolap platform with a number of customers being selected as the target group and check for changes in number of interactions and purchases under different product and user groups. Nonetheless, this has not been possible due to the time constraints. Instead two offline testing methods performing evaluations based on past user and recommended item interaction and another evaluation metric providing the amount of agreement between the developed alternatives and past data are constructed.

The amount of agreement between the results produced by the models are calculated on the basis of the ranking of the products they have offered to the customer. As a consequence of the design, all solution methods rank the same 25 items produced by the same first stage using personal past data to recommend 8 or 10 items. To measure the amount of bi-variate correlation between two ranking or ordinal scales, a metric called rank correlation is widely utilized. By this metric, the amount of similarity between the outputs of three proposed models are calculated. Furthermore, it has also been checked how similar the rankings are to the actual rankings of interaction scores of items and users that has taken place in the one month. Lastly, it has been claimed by many that the e-commerce platforms rank and show products heavily based on their popularity in the platform. A final ranking ordering the 25 candidate products based on their popularity within the data collected has also been devised and it has been examined whether the proposed alternatives recommend products different than that of sole popularity which can be regarded as the amount of personalization enabled by the methods. The other two metrics are constructed in order to yield information about the models' ability to accurately grasp the past events in the data. Firstly, the percentage of times the first 8 items ranked by the models have covered all the 3 of the items that the user has interacted the most in the candidate set of 25 products are calculated separately for all three models as if they are operating under the two different categories. The final evaluation procedure reports and compares the percentage of items that has actually been clicked by the user for different number of items to be recommended changing from 3 to 5 to 8 to finally 10.

product_id	product_id
9774	9774
15115	15115
10853	3789
34762	
27873	
6237	
81456	
3789	



product_id	product_id
9774	27873
15115	6237
10853	16416
34762	
27873	
6237	
81456	
3789	




Figure 2.9: One Proposed Evaluation Metric

Though meaningful, these evaluations are not enough to fully predict the outcomes that can be enabled by the models when in use. Part of this situation is attributable to the offline testing stage missing out on the partially active users and less popular products. The best improvement upon these would be to test the models separately on similar user and item sets during the same time period. A possible offline evaluation strategy would be to make use of time windows and sliding them. Including data up until a certain point in time for training the model and predicting the actions in the upcoming day could be a more robust and less biased method of evaluation. Nonetheless, with the sparse dataset and limited scope of time, even the most active users may not be recommended the products that they have interacted with. In the case of user not having been interacted with most of the candidate products that can be recommended, it is not possible to rank the interaction scores and use them to evaluate the agreement with rankings the models have produced.

Chapter 3

Analysis for Solution and Design

3.1 Literature Overview

Sarwar et al. (2001) [8] mentions that one of the oldest recommender system has been Tapestry. These original implementations of recommender systems have been heavily based on the methods of collaborative filtering. Though there exists user based collaborative filtering methods which utilizes the similarity between user purchases and clustering approaches, it is put forward by many works such as Linden et al. (2001) [14] that item-to-item methods are more scalable and better in practice. Item based collaborative filtering makes use of similarities between different products/services offered by the platform based on various related features such as product category, type and color as well as popularity, clicking and rating variables that are extracted from users' interaction with the product. Collaborative filtering methods include product-user vectors, matrices and matrix methods such as singular value decomposition and principal component analysis to output a value quantifying the user's possible interest in the item which is then converted into a probability taking confidence and preference metrics into account as experimented by Hu et al. in 2008 [6]. Furthermore, there are also hybrid approaches as constructed by Melville et al. (2002) [15] that consider both the user and the item features and augment the existing user data to predict the popularity of new items in catalogue.

Though these hybrid approaches alleviate some of the problems encountered by Collaborative Filtering methods, Wang et al. (2021) [7] contends that collaborative filtering is subject to

issues related to sparsity, scalability and cold-start. Because the product/service and user sets are usually large, only very few interactions are recorded between user-item pairs which presents itself as missing data. Furthermore, the new users and items fed into the system face with a cold-start problem meaning that there is no data reflecting the preferences and ratings of these users or items in user-item space. Taking these into account, model based approaches for recommender systems have gained popularity in the recent years. While some of these methods utilize machine learning methods as clustering, naïve Bayes and tree methods (Sarwar et al., 2001) [8] others exploit graph approaches and horting (Aggarwal et al., 1999) [9]. In addition, deep learning models have also been proposed. Among these models, Covington et al. (2016) [10] has utilized two tower neural networks for candidate generation and ranking in Youtube video recommendations. Furthermore, Wang et al. (2021) [7] have embedded user and item features using deep learning approaches to overcome the problems of collaborative filtering.

In both of the mentioned deep learning models, text mining methods have been used to make benefit of comment and title texts. With the progress of Natural Language Processing methodologies, the use of text in product recommendation has become popular. Barkan et al. (2016) [16] and Grbovic et al. (2016, 2018) [17], [18] have adopted the Continuous Bag of Words and Skip Gram methods introduced by Mikolov et al. in 2013 [11] in product recommendation to find the similarities between products as embeddings and named the adoptions as item-2-vec, product-2-vec. More refined models specifically tailored for product recommendation tasks in e-commerce platforms have been offered by the works of Pfadler et al. (2020) [12] and Chen et al. (2019) [13]. Though these methods may yield better performance in practice, they have lack of explainability and generalization power and have little to do with the Industrial Engineering context.

3.2 Alternative Approaches

Taking the reviewed literature into account, it can be stated that the applicable models can be classified into several categories. Nevertheless, the separation between collaborative filtering methods and neural network based approaches are the most distinct. In general, the collaborative filtering methods make use of algebraic and/or statistical models including

matrix factorization, regression, random forests and different clustering techniques. Due to the fact that this study is designed as a final project in Bachelor's Degree in Industrial Engineering, the neural network based approaches are excluded from consideration in an aim to employ the concepts learned during the education.

Under the broad category of collaborative filtering, two classes can be defined as memory and model based collaborative filtering techniques. An alternative for the problem at hand can be to make use of the memory based collaborative filtering methods which originate from the idea of finding the most similar vector and matrix representations of items, users or item user pairs in space. In many studies such as Sarwar (2001) [8] and Melville (2002) [15], user ratings of items are input for construction of such algebraic entities/ However with the click-stream data, there are not any ratings or numbers. Thus, type of user-item interactions in time can be evaluated using a series of formulas as experimented by Hu (2008) [6] in order to fill for user-item interaction scores instead of the ratings. As the recommender system will be operating under the product pages, it is not sensible to devise a user-to-user collaborative filtering model which is more of a user segmentation method. Though more suitable strategies for the task at hand, direct use of item-to-item and hybrid methods are weak in the sense of more effectively personalizing the item recommendations. Thus, use of these methods for reducing the candidate item space and incorporating a scorer model or supervised learner for the ranking of candidates come out to be more robust.

The second class encompasses a wide variety of statistical learning models from Bayesian to tree methods. With the time dimension of click stream data, range of possible interactions and different product and user qualities such as information about brand, color, shipping method; opting for such models is also plausible. The ability of these models to incorporate real-life features into design increase their strength and flexibility. Furthermore, they are tightly coupled with the concepts learned through Industrial Engineering curriculum such as statistical modeling and multivariate data analysis. Thus, two of these methods namely the linear regression and logistic regression are selected to operate on the second stage in an aim to rank the candidate products utilizing user and item features.

3.3 Assumptions

Click-stream data is composed of implicit feedback. Such information is comparably different from the explicit feedback in multiple aspects. Since the preference of the consumer about product is not directly taken, the feedback cannot be thought of as part of a standardized merit system. As put forward by Hu (2008) [6], different users demonstrate their interest in different ways and increase in number of interactions increases the confidence in the assumption that the user is interested in the specific product but does not necessarily imply that he/she might be liking the product more than another user with less number of interactions with the same product. Nonetheless, it can be argued that for very low number of interactions, more interactions can actually be resulting from increase in preference. Encompassing one month of interactions the number of interactions are usually low for most of the users. Hence, this problem is ignored and preference has been assumed as a cause of heightened number of interactions. However, for the version of the models that can be actually used in the platform that cover larger amount of user data, correction of confidence and preference as put forward by Hu (2008) [6] can be performed.

Secondly, with the click-stream data the intention behind interactions are not clearly known. Consequently, it might be the case that some of the interactions can be result of wrong clicks or may have not occurred out of interest but for example to save a product for a friend or family member. To eliminate bias that can be created by such situations, users and items with number of interactions above a certain threshold are taken into consideration in the data. For these remaining interactions that are considered, the motivation is assumed to be personal interest and purchasing.

Along with the intentions not being clear, the effects of Dolap's own recommendation and ranking models on the collected click-stream data cannot be ignored. As put forward previously, existence of this ranking creates a bias in the observations and as the evaluations are made using the data that is result of this recommendation system, the performance evaluations are subject to bias as well. To negate the effects of this situation, the view data has been left out of consideration as they are mostly result of Dolap's own ranking and recommendation systems. In the offline evaluation process, it has been expected from the model outputs to be different than the past scores to some extent since being the same would indicate that the

models are not producing any output beyond those of the current recommender models that are heavily popularity based.

Finally since the data provided is only in two subcategories, the complete information about user behavior is not gathered. Moreover, for the sake of obtaining clear and detailed evaluations in the testing stage, users and products with very high number of interactions are taken into account. This fundamentally limits the amount of conclusions and insight that can be drawn from the data as consumers might be demonstrating different behavior in different categories, they might probably behave differently compared to the most active users or some users may not yet be actively making purchases in a category he/she would like to interact with in the future. Thus, it has been presumed that it should be adequate to predict consumer preferences of the most active users within a specific product group from the past consumer interactions with the products that can be classified as popular in the same group but not any other products. Furthermore taking two categories, it is principally assumed that the categories of interest are representative of most of the standard user behavior and the conclusions to be derived can be easily generalized into other product categories.

3.4 Summary of Selected Approaches

After careful consideration of the limitations, constraints, approaches and assumptions; three models have been selected to be suitable for the aim and scope of this study. First, an item-to-item collaborative filtering has been applied on the products' space in order to generate 25 candidate under-page items for each item followed by three different personalized scoring or regression models that aid in ranking and selecting from the candidates.

For all models, the candidate generation and personalization steps are separated for the purpose of scalability and ease of use. For the same reason and to decrease the vast number of computations, utilizing item-to-item collaborative filtering is preferred over a hybrid model in the first stage. As a starting point, the user-item matrix is generated using scores resulting from the past user-item interactions. At this stage, different interaction types are scored differently as they show varying degree of interest. At the very core, the scoring scale is constructed by taking into account the percentage of times a type of event is followed by purchase and the meaning of each interaction. In numerical terms, bidding event has given

score of 9, liking has been assigned a score of 6, followed by the score 4 of commenting and a score of 1 for visiting. Once the user-item matrix is obtained, the alternating least squares algorithm is run over the matrix to impute for the missing values, minimize the amount of scarcity and extract the most important latent features from the vast number of dimensions. The columns of the resulting matrix are denoted as product vectors and using the cosine similarity metric, the most similar 25 products to the product that the user has visited are set apart as candidate products to be recommended.

In the second stage, to incorporate personalization within the candidate space, the user's interaction score or probability with the candidate items are predicted using item and user-item features along with a tendency based scoring model, a linear regression model and a logistic regression model. These models incorporate the degree of match between the candidate product and the consumer's past brand, condition and color interest as well as taking the latest popularity trends of product into account. Finally, the products are ranked by their updated probability or scores and the first 8-10 of the products, that will be specified by the Dolap team in actual use, are shown to the user.

3.5 Relation to Industrial Engineering

This study encompasses wide variety of Industrial Engineering topics such as mathematical modeling, statistical analysis, aggregation of data and decision making especially depending on monetary factors.

First of all, the platform can be regarded as a system in which users interact with items in several ways. To grasp the nature of the events within the platform; concepts under decision making and system dynamics are utilized. Most importantly at the identification stage, descriptive analysis of the gathered data using data aggregation and variable transformations has been conducted.

The first stage of the solution is largely based on linear algebraic operations and algorithms. Developing a clear understanding of the meaning and the mechanics of the user-item interactions in higher dimensions has been vital to develop the suitable approaches in this stage. Concepts learned in Linear Algebra courses such as matrix factorization, dimension reduction and matrix approximation are employed to aid in solutions.

In the second stage, techniques learned via statistics, data mining, time series and even quality management have been exploited. The mathematical modeling and variable transformations to come up with user-item features are inspired by the basic concepts of Operations Research. Moreover, the basic ideas learned through economics and decision making courses are kept in mind while the features are created. To devise the models, data mining models powered by statistical modeling and methods are used.

Lastly, processing huge amount of data would not have been possible without the proper programming techniques some of which are learned in programming and statistics courses as R, Python and SQL.

Chapter 4

Development of Alternative Solutions

All three alternative solutions entail a two-staged design in which the same first stage procedure applies to all the alternatives. To summarize an item-to-item collaborative filtering model has been utilized to obtain the 25 most similar items to each product in the training set followed by the personalized ranking of these products via a tendency-based scoring model, a linear regression model and a logistic regression model.

4.1 Stage 1 - Item to Item Collaborative Filtering

Mathematical Background

Linden et al (2003) [19] illustrates the user-item matrix as a collection of user vectors that have dimensions as many as the number of total products. It is further stated that the elements in these vectors can be composed of scores indicating the amount of user's rating, interaction or purchases on the specific item. There are different methodologies to transform the entries into different scales to account for popularity and confidence bias as proposed by Linden et al (2003) [19] and Hu et al (2008) [6] as well. Nevertheless, the term always corresponds to a matrix that has customers in rows and products in columns and amounts quantifying interest of the customers in the products in the cells. As a result of each customer being paired by each product, such matrices are usually mostly sparse, meaning that the amount of cells with non-zero values are quite low compared to the total number of cells.

There are many reasons for the users to interact in a certain way with different products.

However, it can be hypothesized that the interactions or ratings that are filling the user-item matrices are not random. They are instead results of unseen attributes that are related to the user, item or the nature of relationships between the user and the item. These unseen attributes are termed as the latent features as mentioned in Rao et al. (2021) [20]. The objective of the memory based collaborative filtering models is to find similar users or products as indicated by these latent features. To this aim, Hu et al (2008) [6] states that the original user-item matrix can be separated into two matrices one of which carrying information of user-latent features and the other filled with information of item-latent features. Takacs et al (2014) [21] denotes the mentioned user-feature matrix as P and the item-feature matrix as Q , and finally the user-item matrix denoted by R can be approximated as the product of P and Q^T , that is $R \approx PQ^T$. The idea can be conveyed by the image below.

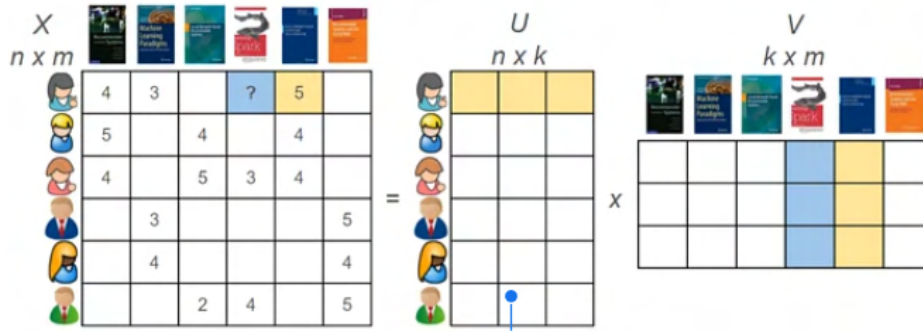


Figure 4.1: Image showing Matrix Approximation in Collaborative Filtering from Project Pro: Recommender Systems Python-Methods and Algorithms

Gosh et al (2021) [22] presents the formula to find the P and Q in the approximation using least squares as,

$$\underset{R_{ui}}{\operatorname{argmin}} \sum (R_{ui} - P_u^T Q_i)^2 + \lambda \left(\sum_u \|P_u\|^2 + \sum_i \|Q_i\|^2 \right)$$

λ is the regularization parameter to minimize the amount of over-fitting which is claimed to have a default value of 1 [22] while u and i denote to the index of the item and user. The conventional method to solve for P and Q is via Alternating Least Squares which is an iterative chicken-egg procedure. The algorithm starts with a randomly initialized Q . The elements of Q at this stage are small random numbers as stated by Takacs et al (2014) [21]. Next, as Q is fixed, the matrix P is updated such that the term provided above is minimized.

After that, the updated matrix P becomes fixed and an optimal matrix for Q to be updated with is calculated by taking the derivatives of the above term. With the updated P and Q , this procedure follows a number of iterations that can be specified each time the algorithm is used. Though it might take time to reach the exact optimal values for a large number of decimals, Gosh et al. (2021) [22] puts forward that the convergence to the global optimal is guaranteed as the problem becomes convex optimization once the values of P or Q are fixed.

Algorithm : Alternating Least Squares (ALS)

Procedure ALS
(P_u, Q_i)
 Initialization $P_u \leftarrow 0$
 Initialization matrix Q_i with random values
 Repeat
 Fix Q_i solve P_u by minimizing's the
 objectivefunction (the sum of squared
 errors)
 Fix P_u solve Q_i by minimizing the
 objectivefunction similarly
 Until reaching the maximum
 iteration
 Return P_u, Q_i
 End procedure

Figure 4.2: Alternating Least Squares Algorithm by Ghosh et al

It is vital to find a close-to-optimal value for the matrix Q as the remaining part of the problem exploits the information contained in this matrix. The columns of Q can be seen as product vectors filled with different elements for each latent feature that are in the rows. Once a proper Q has been found, the similarities of products can be found by calculating the similarity of their vectors. Cosine similarity, Pearson similarity and Adjusted Cosine similarity are the 3 common similarity measures used in the literature. [8] Within this study, Cosine Similarity has been utilized with formula,

$$similarity(A, B) = \frac{A \cdot B}{||A||x||B||} = \frac{\sum_{i=1}^n A_i x B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}}$$

The dot product of the vectors are taken to be scaled by the product of their norms. If each latent feature can be regarded as a dimension, this formula works in a sense that if both the elements in the same dimension are large; the contribution of their product helps the dot product grow larger. If one element in a dimension is large while another is small, the product does not get high in value and contribution to the total is rather small. In a highly

sparse environment, it is a good indicator of similarity for both vectors to contain large values in the same dimension which makes the product and the overall similarity score increase in value. The division by norms is necessary to prevent two not very similar vectors with large elements give rise to a large value of dot product and a large value of similarity. As a consequence, the division is necessary to scale and standardize the dot products in a range of 0 to 1 indicating the amount of similarity.

Implementation

For the successive stages of implementation, it might be suitable to clarify the weighted scoring scale of different interactions. Though the gathered data included 5 different actions that users can take, viewing has been left out of consideration since it is much less important compared to others and is possibly subject to biases that are the result of Dolap's own recommendation and ranking algorithms. For the remaining interactions, it has been observed that of all the purchases in the data 50% are associated with the purchasing user liking the product while approximately 35% are associated with the bidding and/or commenting on the product. Since bidding can be regarded as a more clear indicator of user's interest in the product and as the amount of bids are quite small compared to the amount of likes, bidding has been given more weight compared to liking and commenting. The ratio of weights between likes and comments are adjusted according to the mentioned statistics. Visiting, which is a quite weak interaction compared to others, has been given weight of 1, while a weight of 4 is given to commenting on the product. Bids are associated with a score of 9 and likes are associated with a score of 6.

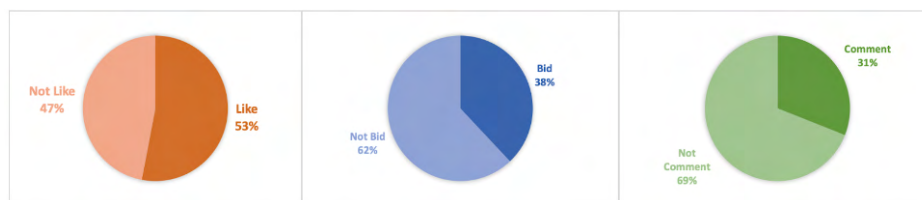


Figure 4.3: Ratio of Likes, Bids and Comments Preceding Purchase

After deciding on a scoring criteria, the space of users and products has been minimized in order to decrease the amount of sparsity within the data. To accomplish this aim, users with a weighted score of interaction below 15 are eliminated which decreased the number of users from 1 million to approximately 318000. In a similar sense, products with a weighted

score of interaction under 11 are left out of consideration which decreased the number of products from 1 million to nearly 170000 in each category. These values are selected by first evaluating the histogram of user and item weighted interaction scores. It has been observed that these distributions are highly unbalanced for small values of interaction scores and as the scores get larger, the number of users or items drop substantially. After the values 11 in the products histogram and the value of 15 in the users' histogram, the distributions become more uniform and hence, these values are selected as the cutoffs.

Although these manipulations reduce the sparsity to some degree, still there have been users with low amount of interactions and products that have been clicked by a small number of users. Nonetheless, more sparsity has been attributable to the user set. Thus, in a follow-up process approximately 35000 most active users and 20000 most interacted products in each category are selected and fixed to be included in the user-item matrix. The rest of the users and products are selected randomly to make up for 150000 users and 150000 items in both categories. Not all users and items separated in the first processing stage are included into the because of the remaining sparsity and the computational costs of running the Alternating Least Squares algorithm and calculating for similar products.

With the user and product set specified, the interactions of users and products until March 16th are gathered and the weighted scores for each pair of user and item are calculated. It should be stated that March 16th is chosen to gather enough data about user behavior and also to allow for extra evaluation data if it would have been possible to conduct the time sliding testing. Once the scores are calculated, Alternating Least Squares of function as implemented in the 'implicit' library of Python has been used with 50 features, a default value of 1 for λ and default value of 15 iterations.

4.2 Stage 2 - Candidate Ranking based on Scoring

Once 25 candidate items to show under the item page are generated in the first stage, one manual scoring model and two statistical learning models are devised in order to rank the candidate items for selection and ordering products under the product page.

For all the models in this stage, the same features of user and products have been used. Most of the selected attributes quantify the amount of fit between the user and the product which

can be categorized as user-item features, while a few that can be grouped as item based features quantify the popularity of the product and carry information about the recent trends in the categories and sub-groups the product is a part of. With these being stated, it might be suitable to provide the full list of features and their meanings in the context of the problem at hand.

Mathematical Notation for Features

All of the defined features are based on the item and most of them are also based on the user as well. Though most features evaluate the behavior on all of the one month, there are a few attributes that grasp the changes in behavior and trends from the last week or three days. Therefore, it might be useful to introduce an indexing of user, item and time horizon for each feature x , such that

$$x_{uit}$$

denotes to the value of the feature x for user u and item i obtained from the interactions of last t days in data. With these being stated, it might be easier to introduce the set of attributes and their descriptions.

It should be noted that for features not dependent on an index, the letter of the index is written in upper cases.

Item Features

BrandBidsRatio_{U7} (Brandbidlc): The ratio of the total number of bids of the products with the same brand that has taken place in the last week to the total number of bids of the products with the same brand

This feature measures the popularity of the brand in the form of bids during the previous week. The measure growing closer to the maximum value of 1, indicates that most of the bidding interactions within the brand took place in the previous week. Thus, the brand has performed extraordinarily better compared to its regular performance during the previous 7 days, which clearly shows that the brand is becoming trendy.

ProductMeanDailyVisit_{Ui3} (Avg1st): Average daily number of visits to the product in the last 3 days standardized via dividing by the maximum daily visits in the training set

This feature quantifies the amount of daily viewing trends of the product in the last days. More popular products get larger values while less visited products are penalized in a sense. This measure grasps the behavior of overall customers browsing in the application during the last days and increases the chances of the user to be recommended more popular products.

ProductMeanDailyLike_{Ui3} (Avg1kst): Average daily number of likes to the product in the last 3 days standardized via dividing by the maximum daily visits in the training set

This feature quantifies the amount of daily likes of the product in the last 3 days. While very similar to the previous attribute, page visits do not always end up with likes. As liking indicates a more sharp preference compared to visiting, this attribute is also statistically significant for the models as will be seen in the upcoming stages.

ProductCondition_{UiT} (Condition1): The condition of the product is entered by the seller to the system and has three categories: new with tags, like new and gently worn. It has been observed that most products either belong to the second or the third category. It has been observed from bids and likes that while users' preferences for the new with tags and like new groups are similar, the amount of traffic for gently worn products are not as high. Therefore, new with tags is given a weight of 1, followed by weight 0.7 for like new and 0 for gently worn products.

ProductBrandType_{UiT} (Bt1): There are four different brand types for the products that are segmented by the Dolap team and shared with the collected data. The segmentation has been done on the basis of pricing and brand types can be listed as economical, popular, lux and ultra-lux. It has been seen in the data that there is fluidity between economical and popular, lux and ultra-lux and a partially less fluidity between the popular and lux brand types in most user preferences. Furthermore, popular brands are interacted mostly by the users followed by economical followed by lux and ultra-lux. Taking the fluidity in user preferences and the popularity of brand types into account, popular brand types have been given weight of 0.8, economical ones are assigned 0.6, lux brands are given 0.5 weight and 0 extra weight has been assigned to the ultra-lux items.

ProductExtremePrice_{UiT} (Prex): Indicator variable that the price of the product is higher

than 900 TLs. Although expensive products are visited and sometimes even liked by the customers, in only 1% of the purchases the customer paid an amount above 900 TLs. Thus, it can be stated that irrespective of the product quality and brand price above a threshold absolutely diminishes the amount of interest the users can express in the product.

User-Item Features

Brand Behaviors

Brand is an important factor in determining the user's interest in the product. Brand loyalty is described in Wikipedia as the consumer's repeated interest to purchase the products or services of a certain brand irrespective of the deficient products and changes in trends. Taking these into consideration, the past interactions with a brand can be thought of as predictors of the future relation between the user and the products of the brand. Thus, the features indicating percentage of user's interactions with the brand of the product in the total interactions of the user have been developed. While a larger percentage demonstrates a more confident assumption that the user might get interested in the product, a smaller percentage contributes less to the total score or target value of the user-product interaction.

UserBrandBidsRatio_{ui30} (Brandbidst): The ratio of user's bids to the brand of the product during the past month to the total amount of bids of the user in last month

UserBrandVisitsRatio_{ui30} (Brandvisitst): The ratio of user's visits to the brand of the product during the past month to the total amount of visits of the user in last month

Between several thousands of products, part of customers in marketplaces are usually only interested in a specific type which might be determined by factors such as product category, brand and color. Though the above attributes with percentages are satisfactory in performance, other features indicating the extreme tendency of the user to a certain brand have also been observed to explain part of the user-item interaction scores in the training set. To devise these features, percentages for each action between the user and the brand in the past month are once again calculated. By local search based on improvement in the goodness of fit in regression models, threshold percentages are determined and the values of features below are changed from zero to one for user-product pairs with percentages above the threshold.

UserBrandBidsExtreme_{ui30} (Brandbidex): Indicator variable checking whether the percent-

age of user's bids to the brand of the product during the last month to the total number of bids of the user in the past month is greater than 0.75.

UserBrandLikesExtreme_{ui30} (Brandlikeex): Indicator variable checking whether the percentage of user's likes to the brand of the product during the last month to the total number of likes of the user in the past month is greater than 0.65.

UserBrandVisitsExtreme_{ui30} (Brandvisitex): Indicator variable checking whether the percentage of user's visits to the brand of the product during the last month to the total number of visits of the user in the past month is greater than 0.65.

Price Behaviors

On the basics, the optimal demand and supply is found through the optimal price point in microeconomics. Although the price is important to almost all the customers, different customers might be willing to pay different amounts. Even the same person, might be willing to pay different values for similar products, or the same product in different days. Though the price attitudes of one customer can be assumed to stay the same in short time periods, the variability in the relationship between different users with prices. This important phenomena is incorporated into the personalization stage through 3 different variables.

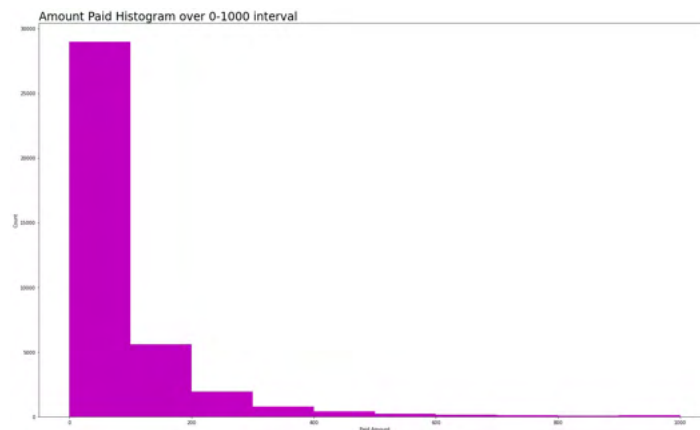


Figure 4.4: Histogram of Paid Amount

It can be seen in the histogram above that the prices of products that are interacted by the users are distributed similar to the normal distribution. This can remind about the distributions of product's attributes related to quality in the Quality Management. Within this area, sigma limits is an important concept to derive conclusions about extreme behaviors and out of

control situations. Inspired by sigma limits, a similar metric for the prices of the products that are liked by the user has been devised. The average value and the standard deviation of the prices in the products that the user has liked have been calculated personally for each user. Then, for each user-item pair, the z-score of item with respect to the distribution of the user's favored item prices has been obtained.

$$Z_{ui30} = \frac{Price_i - \mu_{LikePrice_{uI30}}}{\sigma_{LikePrice_{uI30}}}$$

If the price of the product is closer to the center of the personalized price distribution, it can be assumed that the user can be more interested in the product rather than another one that is too expensive for user's budget or too cheap for his/her quality and brand expectations. Hence, the products are rewarded if they are in 0.5 sigma limits, 1 sigma limits and 2 sigma limits to the mean of the personalized product distribution. While a more central product within 0.5 sigma limits has been rewarded via all three attributes, a product that is far away might have only been signed by being in 2-sigma limits.

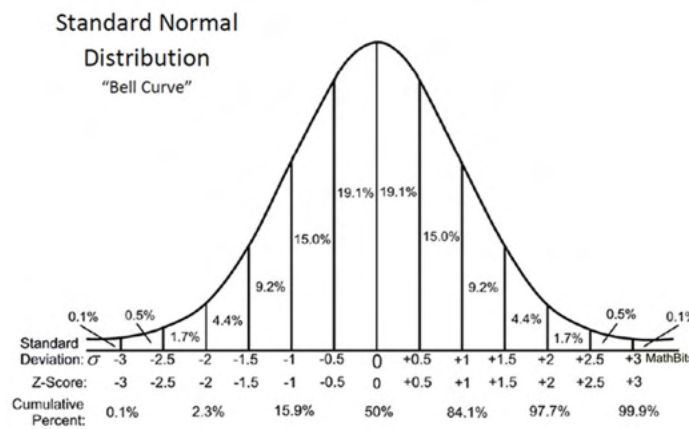


Figure 4.5: Sigma Limits in Normal Distribution from MathBitsNotebook

UserPriceZHalf_{ui30} (Zhalf): Indicator variable if the price of the product is within 0.5-sigma distance to the mean value of prices of the products user has liked in the past month

UserPriceZ1_{ui30} (Zin1): Indicator variable if the price of the product is within 1-sigma distance to the mean value of prices of the products user has liked in the past month

UserPriceZ2_{ui30} (Zin2): Indicator variable if the price of the product is within 2-sigma distance to the mean value of prices of the products user has liked in the past month

While the actual implementation of sigma limits in Quality Engineering concept include 1, 2 and 3-sigma limits; the percentage of products covered does increase in a very small amount from 2-sigma limits to 3. As a consequence, 3-sigma limits has been left out of consideration while the inclusion of half-sigma limits to the models has yield better performance and thus 0.5-sigma limits has been kept in the models.

Color Preferences

There are approximately 45 different colors in Dolap application which the seller can indicate as the color of product that is being put on the platform. It might be hard to grasp a clear idea of whether the customer has tendency to look for products of a certain color as the number of colors in catalog is quite high. Nevertheless, color preferences are the most important in shopping for clothes compared to other categories of shopping such as home appliances or exercising tools. Consequently, the percentage of user's interactions with the products of the same color in all of the past month interactions of the user has been devised as candidate features that can be added to the model. Once checked, one of such features has been proven as statistically significant.

UserColorBids_{ui30} (Colorbidst): The ratio of user's bids to the products with the same color of the product during the past month to the total amount of bids of the user in last month

Color Group Preferences

The number of colors being high might have created possible bias in the color preferences. It can be argued that the color preferences of consumers are not as sharp and can include a variety of close tones and hues as well such as having a tendency for pastel tones or preferring black and preferring dark bags. To account for this factor, the colors are grouped into 8 color groups manually depending on the similarity of them in user choices in clothing. A full version of color groups and included colors can be found in the appendices. It should be noted that a better and more specifically designed version of this categorization can be accomplished via more information and better research. Nonetheless, this topic is not quite related to the aim of this study.

The interaction percentages from the same color group to the total number of the same interaction by the user in a month is calculated and included in the models. By evaluating the results of statistical significance tests on the training set, the following features are decided

to be kept in the models.

UserColorGBids_{ui30} (Colorgbidst): The ratio of user's bids to the products within the same color group of the product during the past month to the total amount of bids of the user in last month

UserColorGLikes_{ui30} (Colorglikest): The ratio of user's likes to the products within the same color group of the product during the past month to the total amount of likes of the user in last month

UserColorGOrders_{ui30} (Colorgorderst): The ratio of user's orders to the products within the same color group of the product during the past month to the total amount of orders of the user in last month

Brand Type Preferences

As mentioned before, the brand types are divided into 4 groups in the data obtained from Dolap Team. Though different brand types have been weighted for further use in training the models, these weights are only to reflect the general user tendencies but not personal preferences. Thus, to personalize the scores with respect to the brand type of the user, similar attributes have been designed and utilized in the models based on their significance in explaining the scores within the training set.

UserBrandTBids_{ui30} (Brandtbidst): The ratio of user's bids to the products of the same brand type of the product during the past month to the total amount of bids of the user in last month

UserBrandTLikes_{ui30} (Brandtlikest): The ratio of user's likes to the products of the same brand type of the product during the past month to the total amount of likes of the user in last month

UserBrandTVVisits_{ui30} (Brandtvisitst): The ratio of user's visits to the products of the same brand type of the product during the past month to the total amount of visits of the user in last month

Training Dataset

To tune for the coefficients of the features in Linear and Logistic regression models, a training dataset has been obtained. Before delving deeper into the numerical details of the training

data, the mechanism and the intuition behind linear and logistic regression models proposed for this study should be explained.

Subject to high amount of sparsity and uniqueness in the item set, the number of purchases performed by each user is on average quite low in the second hand e-commerce platforms compared to online market places with limited range of products that can be bought by thousands of customers. The illustrated situation greatly diminishes the importance and informative value of purchasing as a predictor of future orders of the user. With these being stated, it is not suitable to utilize purchase as a target variable since only one user can have o_{iu} defined as an indicator variable whether the user has bought the product as 1 for the item i while the value will be 0 for all users regardless of their possible interest in the product. However, a product that is still existent in the platform is open to any kind of interactions with all of the customers. Taking these into account, interaction score of users and the products are considered as the variables to demonstrate the interest of a user in a product. The scoring scale is the same as the one constructed for user-item matrix in Stage1 as 9 points for bids, 6 points for likes, 4 points for comments and 1 point for visits.

For the linear regression to learn each kind of relationship between the user and the product equally well, a dataset with normally distributed scoring values has been gathered. By observing the overall scores of many user-item pairs, the possible scores are divided into 9 classes. Starting from score 0, the breakpoints for class changes are as follows: 0, 12, 16, 21, 26, 32, 40, 55 and 80. From each of the 9 classes obtained, 5000 user-item pairs are selected into the training set which in total resulted in 45000 user-item pairs with normally distributed interaction scores. These 5000 scores are coming from 5 different classes flagged by 5 randomly generated time stamps within the last three days of gathered. To calculate each thousand user-item pairing score, interactions from the start of the data period till the randomly generated time-stamp are taken into account. These five different time stamps are preferred over sampling from scores generated till one time stamp to prevent bias that are created from daily or hourly trends and speculations. It is believed that by taking five different time points into account, a more objective picture of user-item relations can be illustrated.

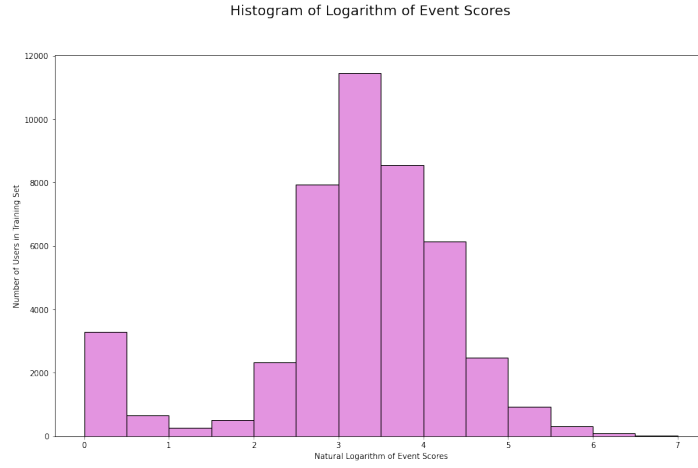


Figure 4.6: Distribution of Log Transformed User-Item Scores in Training Set

Tendency Based Scoring Model

This is the first model devised in order to rank the candidate products. The idea is to directly sum the features created by the user and the item pair and obtain a tendency score u,i . To illustrate a concrete example one row from the training data can be evaluated.

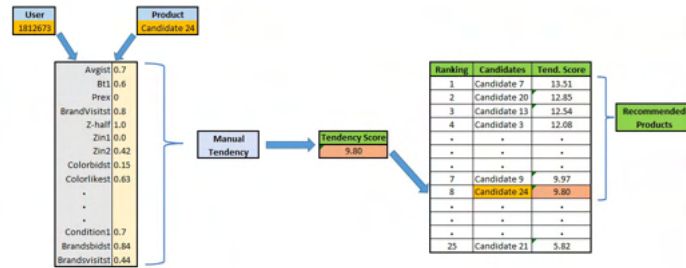


Figure 4.7: Mechanism of Tendency Model

The points obtained by the generated attributes can be seen in the figure. By directly summing these scores, a final potential score of user and item can be obtained as 9.80. The score of user with other 24 candidate products have already been calculated and demonstrated in the figure as well. In the final stage, the candidate products can be ranked on the basis of the total points obtained. According to the tendency based scoring model, the product should be shown as the 8_{th} product to the user on the previous product page.

Being a manual scoring model, there are not any coefficients of the features that are needed to be adjusted or learned. Hence, the tendency based scoring model does not require to be trained once a satisfactory and statistically significant feature set has been found. Nevertheless, it

is rather hard for this model to be generalized into a vast number of categories and different user groups since it does not account for the fact that features might differ in their relative importance depending on the type of products and customers.

Linear Regression Model

In this second model, the idea is to predict the natural logarithm of the score of user-item interaction using a formula with adjusted coefficients for each feature based on the training set. The natural logarithm is taken in order to prevent the effects of having too large values in the target variable. To adjust for the coefficients, more formally to learn the coefficients, a linear regression model exploiting the idea of least squares on the background is employed. Because this is a final project in Industrial Engineering, the mathematical model and foundation of linear regression has been omitted assuming that the reader has the sufficient knowledge about the concept.



Figure 4.8: Mechanism of Linear Regression Model

Two different regression models for the two product categories at hand have been generated using the 45000 user-item pair data via cross validation with 7 folds and 10 repeats. Adjusted R-squared value has been chosen as an appropriate metric to evaluate the performance.

$$R^2 = \frac{SSR}{SST} = \frac{\sum (\hat{y}_i - \bar{y})^2}{\sum (y_i - \bar{y})^2}$$

The performance has been similar in each repeat. Nonetheless, it has been observed in the experimental setting that the values of Adjusted R-squared are decreasing after around the six and seven folds so the training has been stopped. The changes in the Adjusted R-squared are demonstrated on the next page.

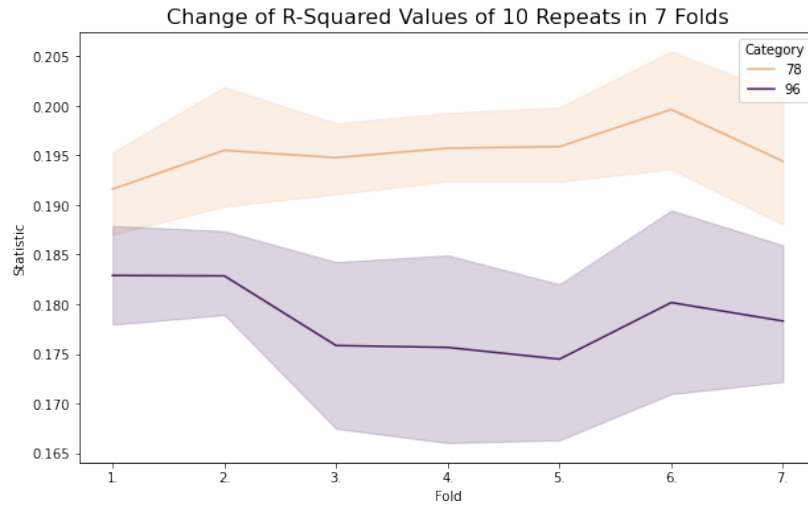


Figure 4.9: Change in Adjusted R-Squared Values in Training

The Adjusted R-squared values are not quite large compared to the ideal value of 1 which is reached when the features fully explain the variance in the observed target values. This is because there are bias in the data stemming from the fact that it is not possible for all users to interact with all products. Furthermore, there are unobserved features that might influence the interest of the user in the product such as size. It has been found out that the values are slightly higher for women pullover compared to those of women bag. This might be the result of a few features better explaining the behaviors under the pullover category than those under the bag category.

Call: lm(formula = .outcome ~ ., data = dat, family = "binomial")					Call: lm(formula = .outcome ~ ., data = dat, family = "binomial")				
Residuals: Min 1Q Median 3Q Max -4.2979 -0.4255 0.1247 0.6722 4.0853					Residuals: Min 1Q Median 3Q Max -4.2936 -0.3782 0.1080 0.6220 3.6326				
Coefficients: Estimate Std. Error t value Pr(> t) (Intercept) 2.25384 0.03476 64.841 < 2e-16 *** brandbidst 0.66474 0.04805 13.835 < 2e-16 *** brandvisist 1.15619 0.04632 24.962 < 2e-16 *** brandbidex -0.48736 0.03290 -14.815 < 2e-16 *** brandlikeex -0.17516 0.02246 -7.800 6.41e-15 *** brandvisitex -0.21375 0.02855 -7.487 7.28e-14 *** zin1 0.09582 0.02246 4.266 2.00e-05 *** zin2 0.11058 0.03068 3.605 0.000313 *** brandbidlc 0.12040 0.02165 5.562 2.69e-08 *** colorbidst 0.40006 0.04542 8.808 < 2e-16 *** colorgbidst 0.06226 0.04535 1.373 0.169817 colorglkeest 0.10280 0.02699 3.809 0.000140 *** colorgorderst 0.07249 0.05346 1.356 0.175087 colororderst 0.19332 0.05852 3.303 0.000957 *** brandtbidst 0.29607 0.02799 10.577 < 2e-16 *** brandtlikest -0.08152 0.02551 -3.196 0.001395 ** brandtvisitst -0.05109 0.03532 -1.447 0.148048 avglst -1.01790 0.34160 -2.980 0.002887 ** avglkst 1.63680 1.99019 0.822 0.410836 shalf 0.07394 0.01461 4.477 7.59e-06 *** conditionl -0.06498 0.01579 -4.115 3.89e-05 *** btl -0.07894 0.03065 -2.576 0.009999 ** prex 0.12102 0.03268 3.703 0.000214 *** --- Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1					Coefficients: Estimate Std. Error t value Pr(> t) (Intercept) 2.53043 0.06401 39.532 < 2e-16 *** brandbidst 0.54095 0.05656 9.564 < 2e-16 *** brandvisist 1.11124 0.05508 20.176 < 2e-16 *** brandbidex -0.53483 0.03926 -13.622 < 2e-16 *** brandlikeex -0.15180 0.02994 -5.071 4.01e-07 *** brandvisitex -0.19805 0.03543 -5.590 2.30e-08 *** zin1 0.14035 0.02599 5.400 6.75e-08 *** zin2 0.14776 0.04419 3.343 0.000829 *** brandbidlc 0.16088 0.02555 6.297 3.10e-10 *** colorbidst 0.36211 0.05408 6.696 2.20e-11 *** colorgbidst 0.12082 0.05102 2.368 0.017905 * colorglkeest 0.16934 0.03478 4.869 1.13e-06 *** colorgorderst 0.02419 0.05400 0.448 0.654199 colororderst 0.30053 0.06202 4.846 1.27e-06 *** brandtbidst 0.31173 0.03178 9.809 < 2e-16 *** brandtlikest -0.07608 0.02975 -2.557 0.010566 * brandtvisitst -0.13526 0.04302 -3.144 0.001670 ** avglst -1.53142 0.31033 -4.935 8.10e-07 *** avglkst 6.60556 1.68873 3.912 9.21e-05 *** shalf 0.05239 0.01791 2.926 0.003498 ** conditionl -0.04416 0.01858 -2.377 0.017464 * btl -0.34946 0.07990 -4.374 1.23e-05 *** prex 0.02213 0.12966 0.171 0.864487 --- Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1				
Residual standard error: 1.102 on 27262 degrees of freedom Multiple R-squared: 0.1966, Adjusted R-squared: 0.1959 F-statistic: 303.2 on 22 and 27262 DF, p-value: < 2.2e-16					Residual standard error: 1.042 on 17692 degrees of freedom Multiple R-squared: 0.1804, Adjusted R-squared: 0.1794 F-statistic: 177 on 22 and 17692 DF, p-value: < 2.2e-16				

(a) Women Bag

(b) Women Pullover

Figure 4.10: Summary of Linear Regression Models

As can be observed, 0-1 transformation of daily average likes of the product in the last 3 days, percentage of users visits to the same brand type and the same color group in the last month to his/her total visits and percentage of users bids to the same color group in the last month to his/her total bids are examples to the few number of features that are inadequate in explaining the variability. A final evaluation of the model using RMSE (root mean squared error) can be provided as well with the formula.

$$RMSE = \sqrt{\sum_{i=1}^n \frac{(\hat{y}_i - y_i)^2}{n}}$$

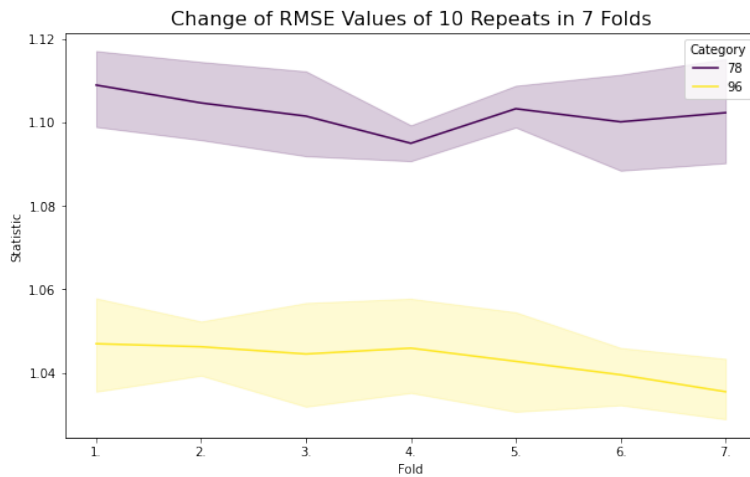


Figure 4.11: Change in RMSE Values in Training

In operation, the linear model outputs the predicted score of user-item using the learned regression formula. Once the scores are calculated for all 25 products that can be recommended to the user, the products are ranked on the basis of the forecasted score.

Logistic Regression Model

The objective of the last model is to predict the probability of user to be interested in the product more than an average user. A logistic regression model with the same features trained in 7 folds and 10 repeats for the reasons mentioned before has been developed for this aim. The fundamental difference of logistic regression from the linear regression is that logistic regression does not work on predicting a linear relationship between the features and the target variable of interaction score. Instead, the aim is to learn a formula of a classifier with feature coefficients.

To accomplish this aim, the training data needs to be separated into two classes. For the model to learn both classes equally well and find the optimal coefficients, the user-item pairs are segmented into two classes from the median score value of 27. The pairs with a score below 27 are marked as 0 while those higher than 27 are classified to have value 1. In this case, the model is run to predict whether the user will be interested in the item such that he/she can accomplish a score of at least 27.

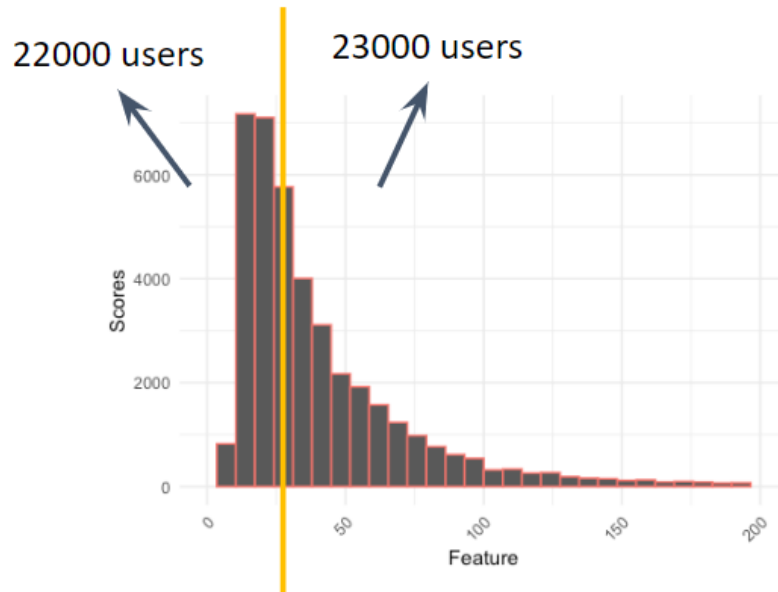


Figure 4.12: Histogram of User-Item Interaction Scores

The mathematical foundations of logistic regression and classification problem are not explained in this study as the aim of the study is to utilize such concept learned during the education. One of the default performance evaluation metrics for logistic regression is accuracy ratio. Accuracy ratio is utilized to calculate the percentage of times the trained model has correctly classified an observation belonging to class 1 by using only the features and the regression formula. The formula in this case outputs a probability value between 0 and 1 indicating the probability of the observation to be coming from class 1. In the default case, a value of 0.5 is selected as a threshold value above which the observation is stated to be belonging to class 1. Though not quite robust to include all aspects of a classification problem, accuracy has been employed in evaluating the performance of the model trained with cross validation.

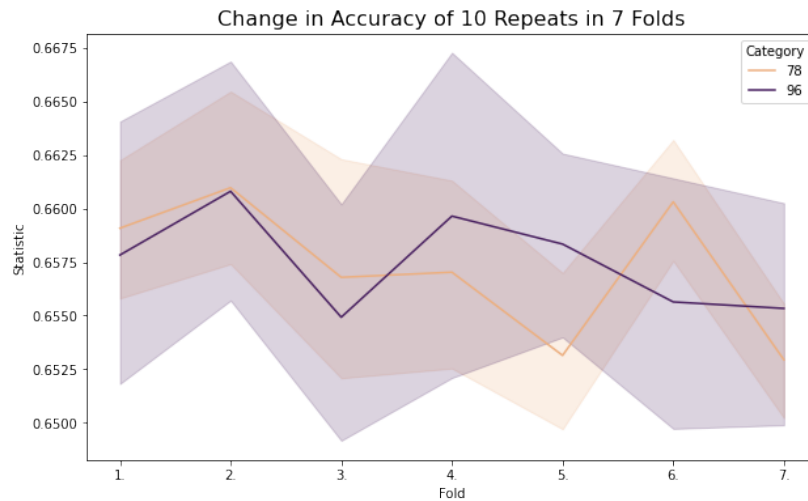


Figure 4.13: Change in Accuracy Values in Training

As can be observed, approximately 66 percent of the observations are correctly classified to be coming from class 1 in left out set. The performances are not quite different between folds and between two categories. The final coefficients and the statistical significance marks of the features for separate models of two categories are presented below.

```
Call:
NULL

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.3379  -1.0203   0.4955   1.0395   2.2935

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.04406    0.06886  -15.163 < 2e-16 ***
brandbidst   0.84332    0.09272   9.095 < 2e-16 ***
brandvisitst 1.66200    0.09075  18.315 < 2e-16 ***
brandbidex  -0.60106    0.06462   -9.301 < 2e-16 ***
brandlikeex  -0.26114    0.04401   -5.934 2.96e-09 ***
brandvisitex -0.23582    0.05586   -4.222 2.42e-05 ***
zin1         0.08724    0.04371   1.996 0.045921 *
zin2         0.15778    0.06070   2.599 0.009336 **
brandbidlc   0.18383    0.04271   4.304 1.68e-05 ***
colorbidst   0.75239    0.08718   8.630 < 2e-16 ***
colorgbidst  0.06534    0.08678   0.753 0.451521
colorglikest -0.14800    0.05297   -2.794 0.005206 **
colorgorderst 0.05263    0.10199   0.516 0.605866
colororderst 0.71674    0.11341   6.320 2.62e-10 ***
brandtbidst  0.30094    0.05441   5.530 3.19e-08 ***
brandtlikest -0.33895    0.05032   -6.736 1.63e-11 ***
brandtvisitst 0.06134    0.06945   0.883 0.377127
avglst       -0.75504    0.72312  -1.044 0.296423
avglkst      -15.08807   4.52829  -3.332 0.000862 ***
zhalf        0.14993    0.03218   4.659 3.17e-06 ***
condition1   -0.21472    0.03095   -6.938 3.97e-12 ***
btl          -0.15157    0.05990   -2.530 0.011394 *
prex         0.26118    0.06459   4.044 5.26e-05 ***

---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 37812  on 27284  degrees of freedom
Residual deviance: 33794  on 27262  degrees of freedom
AIC: 33840

Number of Fisher Scoring iterations: 4
```

(a) Women Bag

```
Call:
NULL

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.534  -1.012   0.508   1.044   2.030

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.933992    0.134140  -6.963 3.34e-12 ***
brandbidst   0.798938    0.116575   6.853 7.21e-12 ***
brandvisitst 1.662612    0.114378  14.536 < 2e-16 ***
brandbidex  -0.835597    0.083077  -10.058 < 2e-16 ***
brandlikeex  -0.202052    0.062033   -3.257 0.00113 **
brandvisitex -0.181111    0.073945   -2.449 0.01431 *
zin1         0.271275    0.053502   5.070 3.97e-07 ***
zin2         0.194890    0.093680   2.080 0.03749 *
brandbidlc   0.284440    0.054110   5.257 1.47e-07 ***
colorbidst   0.750906    0.110374   6.803 1.02e-11 ***
colorgbidst  0.218113    0.102777   2.122 0.03382 *
colorglikest -0.066051    0.072265   -0.914 0.36071
colorgorderst 0.031106    0.108348   0.287 0.77404
colororderst 0.884295    0.128528   6.880 5.98e-12 ***
brandtbidst  0.397680    0.065182   6.101 1.05e-09 ***
brandtlikest -0.308720    0.061941   -4.984 6.23e-07 ***
brandtvisitst -0.002606    0.089297   -0.029 0.97672
avglst       -1.705387    0.719074   -2.372 0.01771 *
avglkst      -0.183977   3.514546   -0.052 0.95825
zhalf        0.043786    0.036811   1.189 0.23425
condition1   -0.091461    0.038398   -2.382 0.01722 *
btl          -0.514776    0.166076   -3.100 0.00194 **
prex         0.100720    0.266933   0.377 0.70593

---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 24538  on 17714  degrees of freedom
Residual deviance: 21946  on 17692  degrees of freedom
AIC: 21992

Number of Fisher Scoring iterations: 4
```

(b) Women Pullover

Figure 4.14: Summary of Logistic Regression Models

It can be seen that the statistical significance of some features are weaker compared to the linear regression counterparts of these models. This can be attributed to the fact that most of the features devised are continuous and the classes in logistic regression model are not actual classes in real life but artificially created groups.

In spite of the model learning two artificially constructed classes, the logistic regression model in operation is quite similar to its linear regression counterparts. While linear regression yields a score for potential user-item interaction, logistic regression outputs a probability for user-item interaction. It can be expected from the logistic regression model to calculate higher probability of interaction for pairs with elevated interaction scores. Thus, it can be claimed that the probability output of the logistic model is similar to the score output of the linear model in a sense. Taking these into account, if the probabilities of interaction for the customer with all of 25 candidate products are calculated via learned formulas of logistic regression, the products can be ordered and shown to the user according to the found probabilities.

Chapter 5

Comparison of Alternative Solutions

5.1 Evaluation Procedure and Results

For the evaluation of the performances of the proposed solution methodologies, three metrics have been selected. While one of these metrics yield knowledge about the similarities between the rankings obtained by the devised models, the other two metrics provide a picture of the models ability to cover the amount of products that the customer has been interested in enough to interact with. Nevertheless, it should be kept in mind that the evaluation has not been performed online and an artificial evaluation scenario using the gathered data has been devised for the evaluation.

To conduct an evaluation, the 200 most active users in each category have been paired with the 500 most popular items in each of the two categories. It has been assumed for each user-item pair $u - i$ that, the user u is on the page of the item i . The task is to find the 25 most similar items to the product i and rank these items based on the personalized features of the user u .

The user-item matrices have been trained as the way it has been described in the Implementation of Stage1. After obtaining 25 candidate items for each user, the previously features have been calculated. The features are plugged into the three scorer models to gather three alternative rankings. Then, depending on the actual interaction score between the user and the offered products obtained from data; the performance of the models could be examined. Though this had been the actual motivation, it has been observed that most of the times, a

high portion of the 25 products to be offered has not been interacted by the user in data. This has reduced the testing stage data from 10000 samples for each category to 100 samples for each category. It can be stated that reduction in the data might have concealed some of the important aspects and outputs of the models as the models have only been evaluated for a very active customer and very popular product set.

	der_count	123 color_bid_count	123 color_order_count	123 colorgroup_likes_count	123 colorgroup_bids_count	123 colorgroup_orders_count
1533	[NULL]	1	[NULL]	11	4	[NULL]
1534	4	[NULL]	[NULL]	[NULL]	[NULL]	[NULL]
1535	4	[NULL]	[NULL]	[NULL]	[NULL]	[NULL]
1536	4	[NULL]	[NULL]	[NULL]	[NULL]	[NULL]
1537	4	[NULL]	[NULL]	[NULL]	[NULL]	[NULL]
1538	4	[NULL]	[NULL]	[NULL]	[NULL]	[NULL]
1539	4	[NULL]	[NULL]	[NULL]	[NULL]	[NULL]
1540	4	[NULL]	[NULL]	[NULL]	[NULL]	[NULL]
1541	4	[NULL]	[NULL]	[NULL]	[NULL]	[NULL]
1542	4	[NULL]	[NULL]	[NULL]	[NULL]	[NULL]
1543	4	[NULL]	[NULL]	[NULL]	[NULL]	[NULL]
1544	7	18	2	15	25	2
1545	[NULL]	55	[NULL]	185	55	[NULL]
1546	2	[NULL]	[NULL]	[NULL]	[NULL]	[NULL]
1547	2	[NULL]	[NULL]	[NULL]	[NULL]	[NULL]
1548	2	[NULL]	[NULL]	[NULL]	[NULL]	[NULL]
1549	2	[NULL]	[NULL]	[NULL]	[NULL]	[NULL]
1550	2	[NULL]	[NULL]	[NULL]	[NULL]	[NULL]
1551	2	[NULL]	[NULL]	[NULL]	[NULL]	[NULL]
1552	2	[NULL]	[NULL]	[NULL]	[NULL]	[NULL]
1553	2	[NULL]	[NULL]	[NULL]	[NULL]	[NULL]
1554	2	[NULL]	[NULL]	[NULL]	[NULL]	[NULL]
1555	2	[NULL]	[NULL]	[NULL]	[NULL]	[NULL]
1556	[NULL]	2	[NULL]	57	[NULL]	[NULL]
1557	[NULL]	2	[NULL]	57	[NULL]	[NULL]
1558	[NULL]	[NULL]	[NULL]	23	[NULL]	[NULL]
1559	[NULL]	[NULL]	[NULL]	23	[NULL]	[NULL]
1560	[NULL]	5	[NULL]	6	14	[NULL]

Figure 5.1: Snapshot demonstrating the Sparsity in Test Data

The first evaluation metric has been selected in order to calculate the amount of agreement between the results produced by the models while ranking the products offered to the consumer. Rank correlation is employed in order to quantify the amount of bi-variate correlation between two rankings. It can be assumed that similar models in operation should agree on most of the relative ranks of the products to be ordered.

Though the rank correlations have been calculated for the three alternative orderings, two other rankings have been devised in order to obtain a more informed outlook on the models. The amount of correlation between the rankings and the actual order of candidate items based on the interaction score with the user in the data have also been found. Lastly, it has been claimed by many that the e-commerce platforms rank and show products heavily based on their popularity in the platform. Keeping this in mind, the 25 candidate products are also ranked based on their popularity within the gathered data.

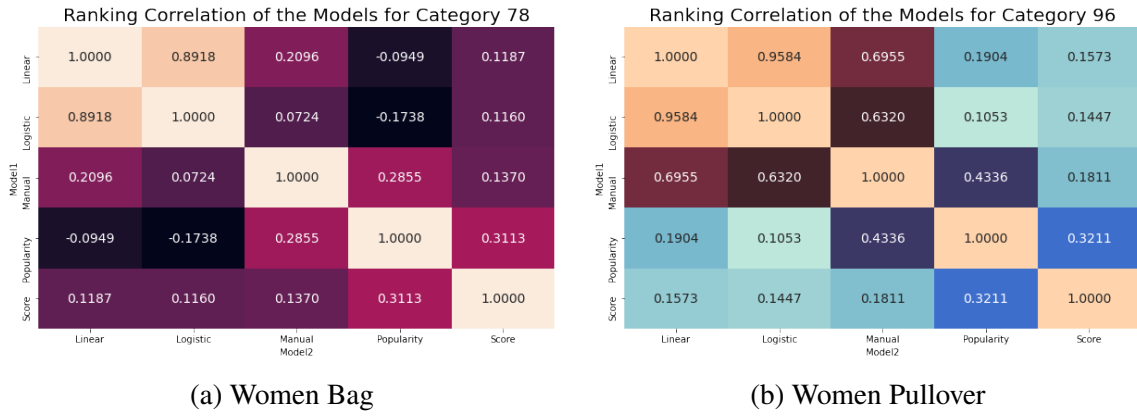


Figure 5.2: Rank Correlations for 5 Rankings

It can be observed that most of the times, Linear and Logistic Regression rankings are pretty much similar to one another. On the other hand, the agreement of Manual Tendency scores with these regression models is high in women pullover while being low on women bag. It could be the case that the ratio of calculated coefficients of regression models for the Women Pullover category could be close to those of the tendency models. However, for regression models the ratios change from one category to another which adds up to the generalization power of the models. Because of this change, the ratios of regression models might be quite different than those of tendency model in the Women Bag category which resulted in a low correlation.

The agreement between the actual scores and the models have always been quite low with correlation coefficients changing around 0.1. This indicates that the models offer different products than those user has already interacted within the data. It may not be sensible to evaluate this result as good or bad, as the scores are result of Dolap's own ranking algorithm. It can only be stated that the models operate differently compared to the currently operating ones.

Finally it can be stated that there has always been agreement between the popularity based ranking and the actual scores to some degree. This clearly shows that the popularity is an important factor in the product rankings and product offers of Dolap. On the other hand, the correlation coefficient of popularity based ranking with the regression models' rankings are quite small. Keeping it in mind that the popularity based ranking always ranks the products the same independent of the user, it can be stated that especially the ranking

algorithms accomplish some degree of personalization while ranking and recommending products which is the main objective of this study.

As a second evaluation procedure, the percentage of times the first 8 items ranked by the models have covered all the 3 of the items that the user has interacted the most in the candidate set of 25 products are calculated separately for all three models as if they are operating under the two different categories with the obtained testing dataset.

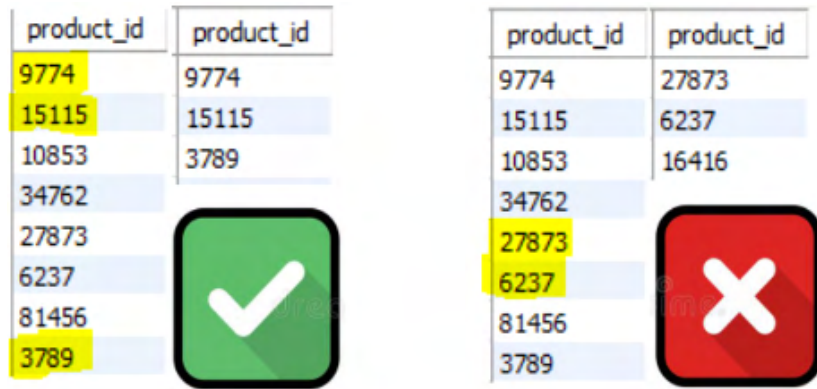


Figure 5.3: A Successful and Unsuccessful Set

The results are visualized as follows,

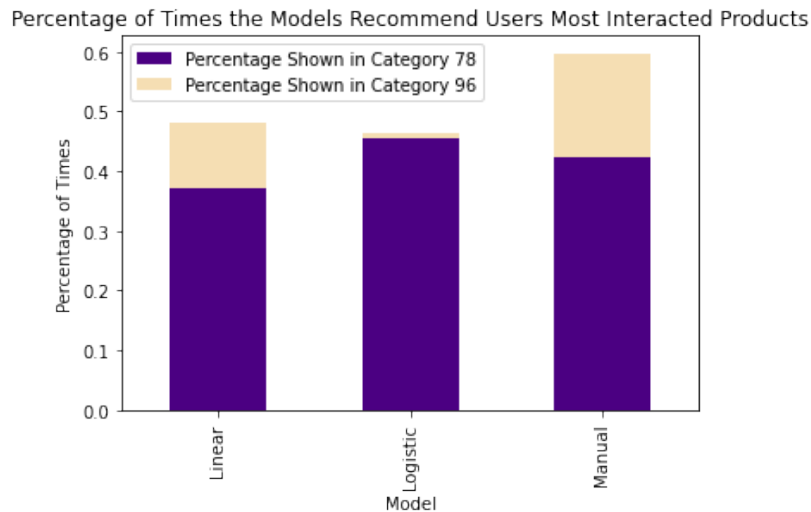


Figure 5.4: The Percentage of Successful Product Recommendations

There are not any benchmark values for these metrics set by any study in the field as they are specifically tailored. Thus, it might be a better idea to compare the obtained values with one

another instead of any ideal value such as 0.8 or 0.95. It can be seen that the performances are very similar and the order of models based on success rate are quite different between the two categories. The successful sample rate of Manual Tendency Model is much higher in Women Pullover category compared to Women Bag which points out the unbalanced performance of the model stemming from stable feature coefficients, ie. a direct sum. nevertheless, the performances of linear and logistic model are quite the similar and while one has performed better in Women Bag the other has performed better in Women Pullover. As a consequence, it might be hard to state that one regression model outperforms the other.

Lastly, the percentage of items that has actually been clicked by the user for different number of items to be recommended changing from 3 to 5 to 8 to finally 10 from the candidate set of 25 items are calculated for each three model separately working in two categories.

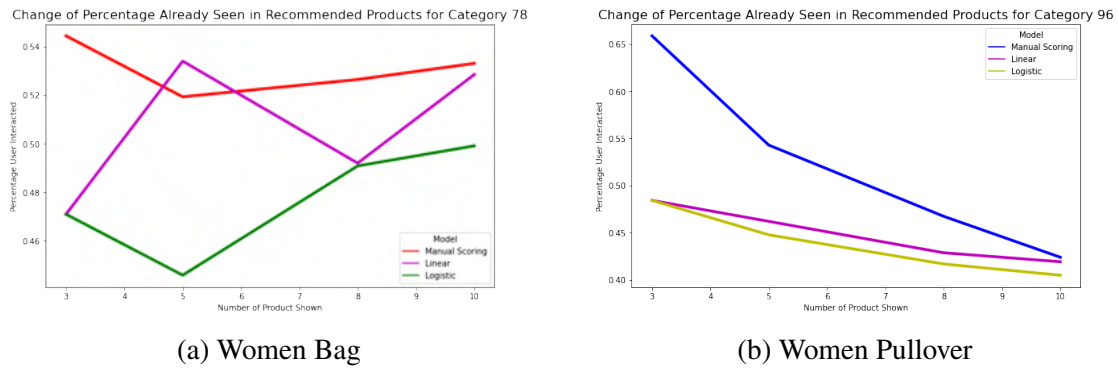


Figure 5.5: The Percentage of Products Interacted

It is interesting to witness that the amount is steadily decreasing in the Women Pullover category with increasing number of products shown to the user. Nevertheless, in the test data it has been observed that Women Pullover category had more sparsity compared to the amount that the Women Pullover category has. Therefore, the average number of products that are interacted by the customer in the set of 25 candidate products might be lower in Women Pullover category.

Once again, it can be seen that the Manual Tendency Model has significantly higher percentages in the Women Pullover category. In the same category, the performances of regression models are quite alike with Linear Regression Model having slightly larger percentage for all values of number of products shown.

The percentage values of Manual Model is also high in the Women Bag category. However,

it can be seen that the values fall far away from those that the model has accomplished in Women Pullover recommendation which demonstrates the unstable performance of this model. Linear Regression Model has been able to demonstrate a performance competing with that of Manual Tendency Model in recommending Women Bags. Though, Logistic Regression Model come closer to the values obtained from Linear Regression the statistics has always stayed as the lowest ones between all three models working for personalized recommendations in Women Bag.

5.2 Proposed Model

It is depicted that the performance of the Manual Tendency Model is quite higher compared to other models regarding the evaluation stage. Though the performance of this model is satisfactory enough, input attributes and their coefficients are not determined considering woman bag and pullover categories. Meaning of attributes might vary in different categories creating different coefficients across different product and user groups. In general the coefficients of the variables in this model can not be adjusted for different patterns in each of the categories which impairs the flexibility and limits the use of model.

The attributes inputted into both of regression models are statistically significant, easy to generalize and be updated in the light of novel data. Thus, it can be stated that these models are sensitive to trends and changes in consumer behaviour. Furthermore, the obtained results are explainable. Lastly, the regression models are shown to recommend different products than those offered by Popularity based ranking which might have high contribution in Dolap's own algorithm. This result points out that a certain degree of personalization has been established by the regression models which has been the principal aim of this study. Between the two, Linear Regression model can be regarded as slightly better compared to the Logistic Regression taking the performance metrics into account. Moreover, Logistic Regression requires threshold scores and probabilities to train increasing the number of parameters to tune. Taking these into account, Linear Regression Model has been selected as the final model to be proposed by the results of this study.

5.3 Assessment of Solution

Some of requirements to implement the proposed methodologies can be listed to require small amount of effort by users, to be calculated and put into action quickly under the product pages and to be able to be trained in a short time. At the same time, the algorithm should be statistically explainable, have logical features and be suitable for future changes in user behavior and product catalog. The approach offered is divided into 2 parts as candidate generation and candidate ranking. In Candidate Generation part, Collaborative Filtering works with a pre-trained matrix and only Candidate Ranking part is left as a final step when the user clicks on a certain product. Hence, it is a good indicator for system suitability that the Product Ranking using the proposed regression model is quite fast, so applying this algorithm provides an efficient and sustainable solution.

On the other hand, training the matrices can take place daily or twice a day. The training process for a category does not take more than at most 10 minutes. In addition, considering that the Dolap team has more powerful computers in terms of hardware and computational capability, it can be stated that the candidate generation part can be applied much more easily and smoothly.

Recommendation algorithm offered in this project is feature based which is meaningful considering the explainability and statistical significance. In different categories, dynamics and shoppers' behavior can be investigated more in detail to extract extra features for the candidate ranking stage. By utilizing information yielded by extra data, this model can be implemented on other categories in the future. Consequently, it can be claimed that the proposed method is robust.

Following the guidelines established in this study for the Candidate Generation part, Dolap team can train dataset and find similar items for each sub-category daily or twice a day. Under the pages of newly added products that have never been interacted with, the most popular products in the category can be suggested to the user. In Candidate Ranking stage, learning regression formulas and applying them are shown to not take a long time, and regression models do not need to be updated at short intervals. The required datasets are kept on Dolap's database and are updated regularly. Application of aggregation and group by functions in SQL for feature extraction is quite easy and lasts at most 5 minutes. Therefore, it is very easy

to keep and process the data in databases and operating the recommendation engine can be regarded as sustainable.

Chapter 6

Suggestions for Implementation

6.1 Implementation

The model consists of two parts, which are candidate generation and candidate ranking.

Candidate generation considers and analyzes the similarity between different items by modelling previous interactions as matrices. These interactions are gathered in a time frame from a specific time to the last run-time of the model which may occur once or twice a day. The working principle of this stage is able to be used for different categories and sub-categories. Also, it is feasible to implement matrix factorization and training in different categories at the same time with parallel processing, which may be done in the computers and servers of Dolap. Although “Matrix Creation” and “Alternating Least Squares” depends on the different characteristics of the data, the maximum time limit for the implementation is nearly 10 minutes, which is one of the major advantages of the proposed methodology. Additionally, one of the challenges in the project were the efficiency of the group by and aggregation functions because the data were in .csv formats and was not able to be used with SQL queries. Since Dolap uses SQL databases, the aggregation and group by processes may be much more efficient.

The second part of the model is candidate ranking. Candidate ranking algorithm should work when users are click on the page of the items. For the ranking algorithm to operate, the 25 most similar items to the product which the user is on the page of should be kept in memory. When the user opens the item page, the algorithm should calculate the features and

apply regression models to rank these 25 similar items in no time. Within this design, the creation of the features is done by using SQL queries which may not create any problems for Dolap team because of the computers with high computational power. Overall, the ranking is constructed when the user is on the item page by regression models that use users' features which is also quite fast.

6.2 Integration

The integration of the model to the whole Dolap system is only possible via implementation of the algorithms for each sub-category. However, in order to test the validity and performance of the model, A/B testing should be conducted with a number of selected users in the two sub-categories used in this study. This can be a first step for the system integration.

Behavior of the users, price attitudes, trends, seasonality and rate of change in the product catalogs may differ across the categories. As a consequence, before adapting the engines for use in different categories, the user-item interaction data under these categories should be analyzed deeply to observe patterns and requirements that are different compared to the data that is utilized in this study to come up with the proposed model. In addition, the regression models may be diversified across different user types by extracting new features depending on the user-item behaviors.

To sum up, the model can be integrated with the overall system after careful considerations and analyses on categories and various user groups.

6.3 Revision of the Model

Since the model uses features related to popularity and the trafficking of users and items, the user-item matrices needs to be updated at short intervals. In categories that are subject to higher rate in the change rate of product catalog, to prevent cold-start problems to occur, Candidate Generation models should operate once or twice a day for sufficient performance.

On the other hand, the attributes, which are the most important components of the Candidate Ranking algorithm, provide information about the special relation between user and product. There is no vital need to daily update these attributes to effectively to grasp the behavior of the

users, price attitudes, and trends in the category. This is because these behavior and trends can be assumed to be stable in the short run. However, it can be observed that the interacted products are mostly in clothing categories, which indicates various seasonality factors to be kept in mind for different products such as bikinis or coats. Therefore, the parameters and features that are used in the regression model should be updated in each 1-2 months considering the seasonality and if necessary, addition or deletion of new features should be regulated according to the current season or change in trends.

Chapter 7

Conclusion and Discussion

7.1 Use of Industrial Engineering Concepts

In this study many different methods of Industrial Engineering are utilized as a range of subjects that were investigated in the courses of the department are reviewed in the design process. To manage the gathered data and to create and present the final models the skills and tools covered throughout the degree program are exploited.

The data analysis aspect within this work is heavily based on mathematical modelling and aggregation of data. These operations have been conducted in order to the manipulate data at hand to derive various insights and analyses that could be used while deciding on the models. These manipulations have been crucial steps to obtain a better understanding of user-item interactions and user behaviour.

Furthermore, at the first stage of proposed method, matrix factorization and linear algebra operations related to Collaborative Filtering are carried out to achieve Candidate Generation. In an aim to numerically model the big and high dimensional data, dimensionality reduction techniques have also been opted.

In the second stage of the model, which aims to rank the previously generated candidates, Industrial Engineering methods were once again used. At this step new features are calculated based on the past user-item interactions through mathematical modelling. In the generation of new features, concepts such as normal distribution assumptions learned in Statistics courses

and variable transformations from Operations Research courses are employed. Also, robust data mining techniques that have been covered in the Data Mining syllabus have been implemented. Another approach utilized is testing for statistical significance during the comparison and evaluation of different metrics.

Finally, a topic this study touched upon has been the economical behavioral aspects of second-hand e-commerce platforms. Because the focus has been on a marketplace, when analysing consumer behaviour related to the price of items, related opinions and concepts on consumer-price relationship from the Economics courses have been used.

7.2 Merits and Significance

It has been a hurdle to get the big data provided to a format in which it is easy to play with the data and conduct full analysis. But even though the number of users and items at hand was substantial, there have still been issues related to the sparsity of interactions. The number of interactions by a user was so minimal compared to the item set obtained. Thus, utilization of algorithms for a smaller datasets has been considered. These algorithms proved to be efficient for big data as well. The models are easy to understand, based the general consumer behaviour logic underlying the data and can be implemented for other product categories. Another advantage of the created models is their fast progress in learning stage.

When compared to popularity-based models, the selected model recommends significantly different set of items. In this sense, it could be stated that the proposed model performs better based on personalization. Recommending items that are not popular, therefore will not be normally shown to the user, could imply better results for the distinct users. In this process, it has also been showcased that in order to use big data and create a recommendation engine, complicated models that are hard to implement and interpret are not necessary. Though the new trend of engines that are powered by deep learning methods can yield slightly better results, they are hard to decipher. The model proposed exploits the basic statistical tools that are based on observations which eases understanding the future data and adjust the models accordingly. The importance of deriving insights and producing attributes through data analysis is exhibited with an adequate performance by the design.

7.3 Impacts of Design

It can be argued that because the model is created using consumer data there might be problems from an ethical point of view. Nonetheless, Dolap collects this data with consent from its users.

From an environmental perspective, the model is created for a second-hand platform which promotes circular economy. Only negative implication of the recommendation model built can be related to the energy consumption for the powering of the computers for training phases. But with the learning time of the proposed model being relatively short, prevent environmental challenges have been tried to be prevented.

Economical consequences are actually the main focus of this study. It is believed through the proposed model that the prioritization of user experience would result in higher purchase rate and increased revenue. This is realized via recommending the best possible items to the respective user. This would result in increased customer satisfaction which in return would draw more customers to Dolap. More and more people that are switching second-hand items could also create economic benefits in terms of sustainability.

Chapter 8

Appendices

	colour	family	colour	family
1	50	pattern	13	red
2	40	metallic	14	yellow
3	-3	pattern	15	red
4	44	red	16	yellow
5	51	pattern	17	pattern
6	-4	clear	18	pattern
7	12	metallic	28	metallic
8	37	pattern	29	earth
9	47	red	30	earth
10	49	clear	31	black
11	1	black	32	blue
12	2	earth	33	yellow
13	3	light	34	red
14	4	blue	35	light
15	5	blue	36	light
16	6	black	38	red

Table 8.1: Color Group Mapping

Bibliography

- [1] Dr. Shahid Bhat, Keshav Kansana, and Jenifur Majid. “A Review Paper on E-Commerce”. In: Feb. 2016.
- [2] Aelita Skarzauskiene, Živilė Baubonienė, and Gintarė Gulevičiūtė. “E-COMMERCE FACTORS INFLUENCING CONSUMERS’ ONLINE SHOPPING DECISION”. In: *Social Technologies* 5 (Dec. 2015). DOI: 10.13165/ST-15-5-1-06.
- [3] Rania Nemat. “Taking a look at different types of e-commerce - TI Journals”. In: *World Applied Programming* (2012).
- [4] Dominique Roux and Denis Guiot. “Measuring Second-Hand Shopping Motives, Antecedents and Consequences”. In: *Recherche et Applications en Marketing* 23 (Dec. 2008). DOI: 10.1177/205157070802300404.
- [5] Haiyan Fan and Marshall Poole. “What Is Personalization? Perspectives on the Design and Implementation of Personalization in Information Systems”. In: *Journal of Organizational Computing and Electronic Commerce - J ORGAN COMPUT ELECTRON COMME* 16 (Jan. 2006), pp. 179–202. DOI: 10.1207/s15327744joce1603&4_2.
- [6] Yifan Hu, Yehuda Koren, and Chris Volinsky. “Collaborative Filtering for Implicit Feedback Datasets”. In: Dec. 2008, pp. 263–272. DOI: 10.1109/ICDM.2008.22.
- [7] Tian Wang, Yuri M. Brovman, and Sriganesh Madhvanath. *Personalized Embedding-based e-Commerce Recommendations at eBay*. 2021. arXiv: 2102.06156 [cs.IR].
- [8] Badrul Sarwar et al. “Item-based Collaborative Filtering Recommendation Algorithms”. In: *Proceedings of ACM World Wide Web Conference* 1 (Aug. 2001). DOI: 10.1145/371920.372071.
- [9] Charu C. Aggarwal et al. “Hortling Hatches an Egg: A New Graph-Theoretic Approach to Collaborative Filtering”. In: *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '99. San

- Diego, California, USA: Association for Computing Machinery, 1999, pp. 201–212. ISBN: 1581131437. DOI: 10.1145/312129.312230. URL: <https://doi.org/10.1145/312129.312230>.
- [10] Paul Covington, Jay Adams, and Emre Sargin. “Deep Neural Networks for YouTube Recommendations”. In: *Proceedings of the 10th ACM Conference on Recommender Systems*. RecSys ’16. Boston, Massachusetts, USA: Association for Computing Machinery, 2016, pp. 191–198. ISBN: 9781450340359. DOI: 10.1145/2959100.2959190. URL: <https://doi.org/10.1145/2959100.2959190>.
 - [11] Tomas Mikolov et al. “Efficient Estimation of Word Representations in Vector Space”. In: *Proceedings of Workshop at ICLR 2013* (Jan. 2013).
 - [12] Andreas Pfadler et al. “Billion-scale Recommendation with Heterogeneous Side Information at Taobao”. In: *2020 IEEE 36th International Conference on Data Engineering (ICDE)*. 2020, pp. 1667–1676. DOI: 10.1109/ICDE48307.2020.00148.
 - [13] Qiwei Chen et al. “Behavior Sequence Transformer for E-commerce Recommendation in Alibaba”. In: *CoRR* abs/1905.06874 (2019). arXiv: 1905.06874. URL: <http://arxiv.org/abs/1905.06874>.
 - [14] Greg Linden, Brent Smith, and Jean-Luc bullet. “Recommendations Item-to-Item Collaborative Filtering”. In: 2001.
 - [15] Prem Melville, Raymond Mooney, and Ramadass Nagarajan. “Content-Boosted Collaborative Filtering for Improved Recommendations”. In: *Proceedings of the National Conference on Artificial Intelligence* (May 2002).
 - [16] Oren Barkan and Noam Koenigstein. “Item2Vec: Neural Item Embedding for Collaborative Filtering”. In: *CoRR* abs/1603.04259 (2016). arXiv: 1603.04259. URL: <http://arxiv.org/abs/1603.04259>.
 - [17] Mihajlo Grbovic et al. “E-commerce in Your Inbox: Product Recommendations at Scale”. In: (June 2016). DOI: 10.1145/2783258.2788627..
 - [18] Mihajlo Grbovic and Haibin Cheng. “Real-Time Personalization Using Embeddings for Search Ranking at Airbnb”. In: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD ’18. London, United Kingdom: Association for Computing Machinery, 2018, pp. 311–320. ISBN: 9781450355520. DOI: 10.1145/3219819.3219885. URL: <https://doi.org/10.1145/3219819.3219885>.

- [19] Greg Linden, B. Smith, and J. York. “Linden G, Smith B and York J: ‘Amazon.com recommendations: item-to-item collaborative filtering’, Internet Comput. IEEE, , 7”. In: *Internet Computing, IEEE* 7 (Feb. 2003), pp. 76–80. DOI: 10.1109/MIC.2003.1167344.
- [20] P. Rao et al. “Matrix Factorization Based Recommendation System using Hybrid Optimization Technique”. In: *EAI Endorsed Transactions on Energy Web* 8 (Feb. 2021), p. 168725. DOI: 10.4108/eai.19-2-2021.168725.
- [21] István Pilászy et al. “Recommender systems and methods using modified alternating least squares algorithm”. Mar. 2014.
- [22] Subhasish Ghosh et al. “Recommendation System for E-commerce Using Alternating Least Squares (ALS) on Apache Spark”. In: Feb. 2021, pp. 880–893. ISBN: 978-3-030-68153-1. DOI: 10.1007/978-3-030-68154-8_75.