COLLABORATIVE FILTERING BASED RECOMMENDER SYSTEM DESIGN FOR E-COMMERCE

(E- TİCARET İÇİN İŞBİRLİKÇİ FİLTRELEME TABANLI ÜRÜN ÖNERİ SİSTEM TASARIMI)

by

Merve ARTUKARSLAN

Thesis

Submitted in Partial Fulfilment

of the Requirements

for the Degree of

MASTER OF SCIENCE

in

INDUSTRIAL ENGINEERING

in the

GRADUATE SCHOOL OF SCIENCE AND ENGINEERING

of

GALATASARAY UNIVERSITY

Supervisor: Assoc. Prof. Emre ALPTEKİN

July 2019

This is to certify that the thesis entitled

COLLABORATIVE FILTERING BASED RECOMMENDER SYSTEM DESIGN FOR E-COMMERCE

prepared by **Merve ARTUKARSLAN** in partial fulfilment of the requirements for the degree of **Master of Science in Industrial Engineering** at the **Galatasaray University** is approved by the

Examining Committee:

Assoc. Prof. Emre ALPTEKİN (Supervisor)
Department of Industrial Engineering
Galatasaray University -------------------------

Assoc. Prof. Ebru ANGÜN
Department of Industrial Engineering
Galatasaray University -------------------------

Assoc. Prof. Başar ÖZTAYŞİ
Department of Industrial Engineering
Istanbul Technical University -------------------------

Date: -------------------------

ACKNOWLEDGEMENTS

This study would not have been accomplished without mentorship and support of various people. First and foremost, I would like to express my gratitude to Doç. Dr. Emre Alptekin, my advisor, for his academic and motivational support throughout the work in progress. His approach, guidance and valuable ideas enlightened and enhanced this M.S. application. I feel privileged by completing this study per his professional proficiency.

I would like to thank management of Hepsiburada for providing the privacy-aware environment to build this framework. My thanks extend to my friend Yağız Demirsoy for his valuable technical support.

I am very grateful to my family, friends and my associates from Mordor, particularly Alper Hankendi, for their regular support during my M.S. study and my thesis.

July 2019
Merve ARTUKARSLAN

TABLE OF CONTENT

LIST OF SYMBOLS

ALS	**:** Alternating Least Squares
CF	**:** Collaborative Filtering
FMCG	**:** Fast Moving Consumer Goods
MAE	**:** Mean Absolute Error
MSE	**:** Mean Squared Error
PCA	**:** Principal Component Analysis
RDD	**:** Resilient Distributed Data Set
RMSE	**:** Root Mean Square Error
ROC	**:** Receiver Operating Characteristics
RSE	**:** Relative Squared Error
SDA	**:** Small Domestic Appliances
SKU	**:** Stock Keeping Unit
SVD	**:** Singular Value Decomposition

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Recommender systems are one of the core engagement functions for e-commerce industry. In a typical recommender system, customer and product data is analysed and a prediction model is generated which evaluates products for prospective customers. In terms of business value, it helps individuals identify their interest among overwhelming variety of products. In this paper, a collaborative filtering based recommender system framework is proposed for Turkey's leading e-commerce platform hepsiburada. First of all, implicit feedback and customer-product prediction pairs are prepared from collected data. Second, a regularized singular value decomposition (SVD) based matrix factorization model is established for collaborative filtering (CF). Customers and products are represented with latent factor vectors. This model is trained with implicit feedback, as the SVD problem is solved with Alternating Least Squares (ALS). Third, predictions are gathered from CF model. Then, predictions are limited to ten-product recommendation sets. At last, recommendations are evaluated by behavioural data generated by prospective customers.

ÖZET

Öneri sistemleri, e-ticaret endüstrisi için temel etkileşim işlevlerinden biridir. Tipik bir öneri sisteminde, müşteri ve ürün verileri analiz edilir ve olası müşteriler için ürünleri değerlendiren bir tahmin modeli oluşturulur. İş değeri açısından, bireylerin çeşitli ürünler arasından kendi ilgilerini çekecek olanları belirlemelerine yardımcı olur. Bu çalışmada, Türkiye'nin önde gelen e-ticaret platformu hepsiburada için işbirlikçi filtreleme temelli bir öneri sistemi tasarlanmıştır. İlk olarak, müşteri-ürün tahmin çiftleri halinde örtük geri bildirim verisi hazırlanmıştır. İkinci olarak, işbirlikçi filtreleme için düzenlenmiş tekil değer ayrıştırma esaslı matris faktörizasyon modeli oluşturulmuştur. Müşteriler ve ürünler örtük faktör vektörleri ile temsil edilir. Problem Alternatif En Küçük Kareler yöntemi ile optimize edilerek bir tahminleme modeli elde edilir. Üçüncüsü, potansiyel müşterilerin ürün puan tahinleri modelden alınır. Ardından, tahminler on ürrüne indirgenir. Son olarak, öneriler potansiyel müşteriler tarafından üretilen davranışsal verilerle değerlendirilir.

1 INTRODUCTION

People have made their purchase decisions by considering recommendations and mentions of their peers. They look for the advice of experts, a trustworthy friend or magazines. Considering the expansion of online services, digitized and user-aware recommendations are emerged for enhanced user experience. Recommender system is an intelligent software which interprets digital footprint of users and predicts their future path or requirements. They count as one of the core engagement functions of modern online retail businesses. Broadly defining, recommender systems collect data and transform into user feedback for items as user-item pairs then they generate a model which predicts scores for missing possible user-item pairs. In terms of business value, it helps individuals identify their interest among overwhelming variety of products.

Offering high relevant and hopefully personalized content to users is at the heart of modern marketing in e-commerce. Product recommendation is the first solution that people have in their mind. Regarding the sales funnel of a classic e-commerce business, engaging customers on discovery step by assisting them for proper decision will return as commercial conversion. Therefore, recommendation system should be tailored for choosing accurate products based on their needs.

Basic recommendation systems can be empowered by data mining techniques such as classification and cluster analysis. Content based recommendation models are based on what customers liked before and offer them similar products. Neighbourhood based recommendation methods seeks the similarity of users based on their behaviour or scaled feedback.

Collaborative filtering is also another approach focused on customer behaviour. A customer's interest to a product can be extracted from behavioural data. Collaborative filtering methods generate score prediction for items by characterizing both users and items within the same model. Context aware models are also emerging for online services, some recommendations may make sense only within the context such as time, location or relationships for specific industries.

This study aims to build a recommender system for an e-commerce platform which is empowered by customer behaviour, up to date, relevant yet unforeseen and personalized. Considering the accessibility of behavioural data and sophistication of customer taste, latent factors based collaborative filtering is proposed. An implicit feedback model is designed and retrieved from data. Customer and products are represented with latent factors. A prediction model is generated by optimizing a dynamically regularized singular value decomposition problem with alternating least squares. Model training parameters are fine-tuned and predefined predictions are delivered with the model. Then prediction pairs are filtered by total number of 10 and personalized recommendations are generated. For evaluation, prospective customers are tracked for a period and precision and recall parameters are calculated. Future work and improvements are also discussed.

2 LITERATURE REVIEW

Recommendation systems consist of three components in such customer data, an algorithm for data processing and performance evaluation. Considering the variety of digitalized services, big data and scalability, personalized recommendation generation become an engineering problem in terms of source optimization and efficiency. Therefore, recommendation accuracy and scaling have found a place in digital service design literature.

This literature review covers the basis of recommender system types, error optimization, and usage evaluation. Recommender systems are grouped in content based and collaborative filtering based models. Different algorithms and applications are introduced.

Since we propose a learning recommendation framework, machine learning concepts are also mentioned. Several applications which are widely used in digital services are introduced. Learning types and problems are disclosed with various examples from academic studies. Also issues related to big data are briefly acknowledged.

2.1 Recommender Systems

Recommender systems are techniques which predicts users' future or potential interests by computing and filtering user feedback of items. An item is basically what a system can offer to its users. Regarding the modern paradigms, it can be a restaurant, house, movie, video, product, joke, image, friend, even a partner.

Recommendation systems which employ a social strategy made their first appearance in literature in the mid 90's. To leave new users alone in a digital system is defined as a pain point. A general history-of-use method is offered for advanced customer orientation. Assuming a video recommendation system, active viewer's personal rating is compared to the video viewer community and unseen videos are recommended to the viewer. Also a 'virtual community' is defined as people with shared characteristics and interact. Hypothetically, if this community would have been interacted they would influence each other. Recommendation systems are making an arrangement for people to share their personal information without the associated cost of communication. (Hill, Stead, Rosenstein & Furnas, 1995)

A recommendation problem can be reduced to rating prediction for items which is not seen by users. Recommendation systems can be classified into 3 categories:

- Content based recommendations: user will be recommended items similar to the items they preferred in the past.
- Collaborative recommendations: user will be recommended items which are preferred by people with similar taste.
- Hybrid approaches: content and collaborative recommendations (Adomavicius & Tuzhilin, 2005)

Recommendation systems depend on customer data. Eventually recommender systems can suffer from data privacy violations and even negative reaction from users. It can be perceived by the customers as their purchase history is used against their will. Several studies address the privacy protection. Customers' identity, demographic profile, purchase history, rating history, browsing behaviour and search history are encrypted and anonymised by a third party model called the Alambic agent. It is based on division of trust principle. (Aïmeur, Brassard, & Onana, 2007)

Algorithm can be regarded as core feature of a recommendation system. Nevertheless, a recommendation system is related to user experience, data collection, experimenting and other issues. In fact, most of the algorithms are already functioning well, it is the smallest problem to solve. Recommendation systems should be reshaped by evaluating user needs. An evaluation strategy should be proposed by definition and to measure the effectiveness to understand user goals and context. We can build a properly serving system if and only if we understand the results. (Ekstrand, Riedl, & Konstan, 2010)

Rating data used in recommendation system can be obtained in several methods. User's opinion for a product can be in an explicit rating form which is scalar or binary. In this case, user selects a numeric value. On the other hand, opinions of user can be extracted by the designer. This is called implicit feedback which is based on user's behaviour. Explicit feedback is defined as the categorical assessment of the customer for a product regarding their interest. For example, Netflix collects star ratings for movies and Youtube has thumbs up-thumbs down options. Implicit feedback is the customer behaviour inferring the user's preferences. Implicit feedback represents the customer's opinion indirectly. Purchase history, browsing history, search patterns or page view period are examples for implicit feedback.

Explicit feedback is usually preferred thanks to the ease of using pure classified information. However, implicit feedback is less limited in terms of data collection effort. Implicit feedback depends on the goal of the recommender system. It can be time spent on a page, a clicked product or abandoned transaction. It should be converted into numeric values with an appropriate transform process. There is no negative feedback in implicit feedback models. It is hard to presume a user disliked an item. Considering the tracking all user behaviour, implicit feedback can be noisy. Numerical value of explicit feedback is the customer preference, while implicit feedback indicates the confidence. A customer's real preferences and true motivations are not more than guess. Therefore, implicit feedback based recommendations should be generated and evaluated with approximate metrics. (Hu, Koren, & Volinsky , 2008)

2.1.1 Content Based Recommender Systems

Content based recommendations attempt to discover the products which are similar to those the given user showed an intention before. Basically, the approach is to match the interests and features of a user profile to the content of the items.

Content based information filtering is performed in three steps. First step is to retrieve structured information as item content. Items are represented with feature spaces. This representation is an input for the second phase which is to learn the profiles. The user preferences are collected for the attempt of generalized construction of user profile. Then items are filtered by matching the items with user profiles by computing similarity metrics. Filtering component predicts the incline of a user for an item and ranks the potential interesting items.

Content based recommender systems are advantageous sides compared to the collaborative models. Content based recommendations are based on the profile of active user, while collaborative filtering methods needs ratings from other users. Besides to that, content based recommendations are more transparent than collaborative filtering, because content features are explicitly described. Also, content based recommendations suffer less from new items with no rating. There are also shortcomings of content based recommendations. First of all, content analysis should be mature enough to distinguish what a user likes and dislikes. Feature representations capture the content influencing the users. Other shortcoming is over analysing the features so there is no chance for unexpected suggestion. This is called as serendipity problem which highlights the limitation of novelty. Another drawback is understanding new users with few ratings. (Lops, Gemmis, & Semeraro, 2011)

Cantador, Bellogin, & Vallet presented and compared several content based recommendation models used in social tagging systems. Users create the content and annotate it with tags. The whole set of tags is an unstructured classification which is called folksonomy. These tags are considered as content features describing both users and items. Features are weighted in terms of importance in several distributions. Then, several similarity functions are evaluated for each tag weighting scheme. They emphasised on penalising popular tags which are commonly used and therefore does not describe the item content precisely. (Cantador, Bellogin, & Vallet, 2010)

Despite the shortcomings of content based recommendations, content based recommendations are included in new model proposals. Along with the high variety of knowledge sources, such as user generated content, hybrid systems are designed for better predictions. In some applications, explanation of recommendation and content awareness can be inevitable. User generated content, visual and multimedia features and heterogeneous information networks are new data related trends on content based recommendations. Deep learning is also emerging as an algorithm trend for its ability to rapidly model seasonal and sequential recommendations. (Lops, Jannach, Musto, Bogers, & Koolen, 2019)

Regarding the cold starter problem of content based recommender systems, a hybrid model combining collaborative filtering with content is proposed. In order to reduce the dimensionality of features, singular value decomposition is applied for TV view preferences of users. Item based collaborative filtering is applied for new users. Collaborative filtering technique works well for utilizing high number of ratings and users. Similarity of user vectors and program vectors is calculated with cosine similarity. Then the model is evaluated with mean absolute error and a custom confusion matrix model. (Barragáns-Martínez, Costa-Montenegro, Burguillo, Rey-López, Mikic-Fonte, & Peleteiro, 2010)

Another hybrid model is proposed for predicting how an active user rates an item. The relationship among users, items and features are represented by using Bayesian networks. Bayesian networks define the graphical interdependencies and strength of these relationships by means of probability distributions. Hybrid model is used to construct the knowledge how an active user rates the items and the relationships related to target item. Hybrid component is node combining both collaborative and content based relevance of an item for a user. The accuracy of model is improved by this combination. (De Campos & Fernandez-Luna, Huete, & Rueda-Morales, 2010)

2.1.2 Collaborative Filtering Based Recommender Systems

Content based recommendations are based on generating user and product profiles to characterize its nature. Users and items are associated by the external information and content strategy. Collaborative filtering is an alternative to content based recommendations by utilizing user behaviour without requiring content profiling. It is based on users’ behavioural history. It resolves the affinity within users and interdependencies among items to identify possible valuable user-product associations. Briefly, content based recommendations attempt to discover the products similar to the ones users preferred before, while the collaborative recommendations identify the users whose preferences are alike, therefore their insight can be shared. It is first articulated by Goldberg in 1992 as collaboration of people to aid one another execute document filtering by interpreting readers' reaction to documents they read. In collaborative filtering systems, users' analytical judgements for the products they use are shared for better decisions. (Goldberg, Nichols, Oki, & Terry, 1992) Reasonable prediction of an active user's preferences are generated in light of others opinions. In other words, if users agree about relevance of certain items, they will likely agree about other items. Therefore, if people like similar items on a user-item group, it is predictable that one person likes a certain item which hasn't seen by him thanks to the fact that it is already liked by someone in the group. In most general sense, collaborative filtering model incorporates three components, similarity computation, neighbour assignments, and prediction of preferences. (Batmaz & Polat, 2016) Personalized recommendation is the ultimate deliverable of a collaborative filtering model.

Comparing the content based filtering, collaborative filtering has three major advantages:

- First of all, it is possible to filter products by not analyzing the content, as the relevance, quality and interest is determined by the human. The only information needed is a user showed intention to the product.
- Second, collaborative filtering engages how quiet a product satisfies a user's wishes, it is beyond the content analysis. Quality or taste is not possibly analyzed by computers, it is woven within human decisions. Focusing on content may diminish the flexibility on user part.
- Third one is the serendipity of recommendations. People may make desirable decisions by accident. Collaborative filtering technique can generate recommendations which are valuable to the user but it is not expected according the content of the product or user. (Herlocker, Konstan, Terveen & Riedl, 2004)

Despite the advantages, there are also disadvantages of collaborative filtering. One of them is sparsity of user preference. Regarding the variety, It is not easy to find users with similar intentions. Recommendations may include much similar products. Similar products should be treated as same as the model improves better. Gray sheep or distinct users may bias the model.

There are two methods for collaborative filtering, first one is neighbourhood methods discovering the relationships within the users or alternatively within the products. The item oriented method assesses a user's inclination for a product by the user's rankings provided for the neighbouring items. It assumes the user has the similar desire for bystander items. Same approach also applies to the user neighbourhood based collaborative filtering. The second one is latent factors models. A simple recommender system models the similarities between people or products, added to that latent factor model tackles the problem with a more sophisticated approach by converting data into a theme space. Then the similarities in this theme space is discovered. Latent factor models are suitable considering the problem of explaining the observed customer feedback. Latent space explains the ratings by characterizing both users and products as factors inferred from implicit feedback. A user's predicted rating for an item is equal to the dot product of user's and item's latent factor values.

User neighbourhood collaborative filtering is based on predicting one's preferences based on his similarity of other users. If a user has high agreement with some other user in terms of common rating to same items, that user will have similar preferences of unseen items to other user.

$$p_{u,i} = \bar{r}_u + \sigma_u \frac{\sum_{u' \epsilon N} s(u,u')(r_{u',i} - \bar{r}_{u'})/\sigma_{u'}}{\sum_{u' \epsilon N} |s(u,u')|} \tag{1}$$

Eq.1 computes prediction of user u's preference for item i which is represented as $p_{u,i}$. Similarity of users u and u' is represented as (u,u') . Mean rating of user u is $\bar{r}_u$. Subtracting user's mean rating will help compensating the differences in users rating scale. Some users may tend to give lower ratings in general. Standard deviation of user's rating, which is represented as σ_u, will also compensate user's mean rating and user specific rating spread interval. (Ekstrand et al., 2010)

There are several methods for calculating user similarity. Pearson correlation is one of them. This method finds the statistical correlation between two user's common ratings to determine similarity. One pitfall of this method is, it calculates high similarity between users with few ratings in common. Constrained Pearson correlation scales the ratings in a like-dislike range by subtracting the neutral value from scales ratings. (Polatidis & Georgiadis, 2017) Spearman rank correlation coefficient is another method. It is derived from Pearson, except the ratings are replaced by ranks. Highest rated item by user is ranked as 1 and so on. Cosine similarity is a vector-space approach to similarity calculation. Users are represented by item rating vectors and similarity is measured by cosine distance between item rating vectors. Cosine distance is calculated by dividing dot product of two vectors by the product of their Euclidean norms. (Patra, Launonen, Ollikainen & Nandi, 2015)

The data set behind the recommendation systems may contain high variety of features. Multi-dimensional data may come out with complications in model training. To overcome these issues, a recommender system approach is proposed. It is based on high dimensional model representation and user to user collaborative filtering. In other words, this framework contains more than one model for handling feature variety. Representation constructs a purchase history vector for each customers by producing one output from a model for a set of inputs for other model at a time. Prospective customers are targeted and similar customers are identified with cosine similarity of purchase history vectors of customers. (Kasap & Tunga, 2017)

Item neighbourhood collaborative filtering uses the similarities between rating patterns of items. Eq.2 computes prediction of user u's preference for item i which is represented as $p_{u,i}$. Baseline predictor of rating from user u for item i is set as $b_{u,i}$. It is required to eliminate the possibility of negative similarity scores and therefore negative predictions. It will bias the predicted values which does not map the user-rating scale.

$$p_{u,i} = \frac{\sum_{j \epsilon S} s(i,j)(r_{u,j} - b_{u,i})}{\sum_{j \epsilon S} |s(i,j)|} + b_{u,i} \quad (2)$$

Similarity of items i and j can be calculated by the methods mentioned for user similarity. Another method for finding the final prediction is to use weights instead of similarity.

User similarity should be evaluated comprehensively as it is the core of collaborative filtering. In order to deal with the data sparsity and common rated item dependency, Wang proposed an extend Proximity–Significance–Singularity model combined with item similarity. This collaborative filtering based user similarity measure is tested in various sparse data sets and results prove that this similarity measure is plenty flexible and breaks the constraint of common rated items. (Wang, Deng, Gao, & Zhang, 2017) Polatidis and Georgiadis argued the quality of user similarity based recommendations. They proposed a constrained Pearson correlation coefficient model. If number of co-rated items between two users is greater than a predefined threshold, and Pearson correlation coefficient is greater than a predefined threshold then their similarity should be positively adjusted. Otherwise similarity should be penalized. Proposed model is tested on three different data sets. Mean Absolute Error and Root Mean Square Error, precision and recall metrics are calculated to evaluate the results. By adding the constraints, the accuracy of the recommendations are improved (Polatidis & Georgiadis, 2017).

Latent factor models are fairly popular for collaborative filtering, thanks to their promising level of accuracy and scalability. Single value decomposition is well established for identifying latent semantic factors. Latent space characterizes products and users on factors which are form of user feedback and demonstrates ratings. Matrix factorization model is a joint latent factor space of dimensionality f, and user-item interactions are products of factors. As expressed in Eq.3, a predicted rating of user for item ($\hat{r}_{u,i}$) consists of baseline predictors for ratings (b_i , b_u and μ) and the interaction between user and item ($q_i^T p_u$). q_i is an f dimensional vector of factors which item belongs to and p_u is the vector of factors which user belongs to.

$$\hat{r}_{u,i} = \mu + b_i + b_u + q_i^T p_u \tag{3}$$

Baseline predictors and vectors can be revealed by minimizing regularized squared error. λ controls the extent of regularization. This parameter should be optimized by cross validations.

$$\min_{b*,q*,p*} \sum_{(u,i)\epsilon K} (r_{ui} - \mu - b_i - b_u - q_i^T p_u)^2 + \lambda(b_i^2 + b_u^2 + \|q_i\|^2 + \|p_u\|^2) \tag{4}$$

Alternating least squares technique fixes p_u and optimizes the equation shown in Eq.4 for q_i and rotates the problem by vice versa. Stochastic gradient descent optimization is an alternative method for minimization problem. (Ricci, Shapira, Rokach, & Kantor, 2011) Alternating least squares is implemented for an implicit feedback based recommender system by Takacs and Tikk in 2012. Implicit feedback based model is optimized by minimizing a ranking objective problem instead of conventional mean square error. A ranking based method is proposed, which optimizes the original objective function. Key components of this model is a matrix factorization model, a ranking based objective function and an alternating least squares optimizer. (Takacs & Tikk, 2012)

Hu & Koren et al. also mentioned singular value decomposition for implicit feedback dataset based collaborative filtering applications. Since cost function of an SVD contains vast amount of user-item pairs, this minimization problem cannot be solved by a conventional technique such as stochastic gradient descent, which is preferred for explicit feedback datasets. ALS is based on fixing user factors or item factors, the cost function becomes quadratic. This model is able to scale linearly with the data size. In order to overcome the data sparsity and dense cost function, new ratings are generated by using existing ratings with confidence level. It is considered as a novel way to explain recommendations. (Hu, Koren et al., 2008)

Billsus & Pazzani proposed a framework for latent factors based recommendations which also tackles with data sparseness. They applied singular value decomposition to an initial user-rating matrix and evaluated the proposed model with large data. In order to reduce dimensionality null values are replaced with global values. They find out that SVD approach is significantly performed better. The sparse matrix of user ratings is diverged into a like/dislike matrix and training data is generated. Number of dimensions is defined and singular vectors are eliminated. Then a neural network is trained. Predictions consist of the item's user ratings as feature vector and k-dimensional space of feature vector or item's representation in terms of features. Results are evaluated by F-Measure which is a weighted combination of precision and recall. Evaluation results provide an evidence that learning algorithms can lead improved performance on collaborative filtering. Besides to that, dimensionality reduction results in additional performance increase. (Billsus & Pazzani, 1998)

Since dimensionality reduction is emerged as a method for handling sparse data, principal component analysis based model which results in subset of the rating matrix named as Eigentaste is proposed by Goldberg. It is a collaborative filtering algorithm which obtains real user ratings under a common set of items. Different algorithms are tested by using a large sparse data set and calculation of Normalized Mean Absolute Error and it is shown that Eigentaste model performs faster and without any shortage on accuracy. (Goldberg & Roeder, Gupta, & Perkins, 2001)

Another method for dimensionality reduction is combining the items under topics. Therefore, user's preference for an item is represented by the combination of the user's interest in the topic and an item's relevance to the topic. This conversion yields two results. First, this approach reduces the dimensionality. Second, by dropping the singular values, noise of data is eliminated and strong trends shapes the model. Decreasing the noise, eventually, results in high quality recommendations. (Billsus & Pazzani, 1998)

One of the well-known problems of a collaborative filtering design is how to make recommendations for new users. This is a major challenge and called as cold start problem. Shahraki and Bahadorpour proposed ask to rate technique. The most direct and adaptive way to cope with such problem is asking for explicit feedback from new users right away. Another suggestion is a random strategy, which is based on the prediction of which items are rated more frequently by the users. Entropy can also be applied to the items. The entropy on a target item is dispersion of item ratings in the matrix. Another strategy is popularity index. Since entropy and random strategy are short in accuracy, they can be combined with popularity. HELF (Harmonic mean of Entropy and Logarithm of Frequency) combines popularity end entropy score of items. Logarithm frequency represents the popularity. Adaptive methods mostly relies on initial rating of users. Interpreting the initial item, new items similar to the selected one or cluster of selected on can be predicted and offered to the user. (Nadimi-Shahraki & Bahadorpour, 2014)

2.1.3 Model Validation and Evaluation

A recommender environment consists of users, data and application consuming the recommender system. Users and their ultimate goal should be defined. Data should be collected and transformed into the characteristics of algorithm. Application is the platform customer interacts to. (Ricci et al., 2011) In order to verify productivity of the recommender environment, designer should define success criteria and propose an evaluation approach.

Evaluating the collaborative filtering algorithms can be challenging. First of all, each algorithm interprets data in a unique way. Second reason is success criteria of recommendation engines. Overall, success is user satisfaction, however the goals for which an evaluation is generated may differ. Another reason is user consistency. There is a ground rule for recommendation engines: an algorithm can only reach the accuracy level of users' ratings variety for same items. There are different dimensions of a best practise recommender system. User and item space coverage is one of them. Also a recommendation system should emphasise on novel recommendations, it means offering users items they don't know about. Another concept is serendipity, it is how surprising the recommendations are. Obviously, serendipity and novelty should be balanced to accuracy. (Herlocker et al., 2004)

A recommendation system which constantly promotes popular products or easy-to-predict items is not considered as valuable. It should be both satisfying and effective in terms of relevance and commercial conversion. An accurate recommender system is the one with new product offers while it depends on a collaborative act.

Making progress in learning models is only visible when evaluation is possible. There is high variety of methods and refined approaches for interpreting data. However to find the one which suits the problem the best, systematic evaluation is inevitable. Considering the real life problems mostly include big data, relying on human expertise is not scalable. Different machine learning algorithms produces apparently different results and it should be assessed by statistical tests.

Learning models are instantly evaluated for progression. Error rate is the proportion of errors made over the set of instances and it is the metric for overall performance. The machine usually learns from the data which is also used as test data. Therefore, the error rate of training data does not necessarily reflect how they will perform independent test data. The error rate on training data is called the re-substitution error.

It is essential to test the algorithm with a test set which is not used for training, knowing that it is not an estimation of future error rate. There are three sets of data. The training data is utilized for model generation. The validation data is used to optimize model parameters. Then the test data is used to calculate the error rate of final optimized method. (Witten, Frank, Hal, & Pal, 2016)

Several error rate measures can be used to evaluate the success of numeric prediction. Predicted values and actual values are interpreted in several formulas. Mean-squared error is the most commonly used measure. The difference between predicted rating (p_{ui}) and actual rating (a_{ui}) is considered as error. MSE formulation is shown in Eq. 5. Assuming there are total number of n tuples, mean of sum squared error calculated as a measurement. Other than easily manipulated mathematically, there is no particular advantage to use MSE as a measure (Witten et al., 2016). Mean absolute error is an alternative, comparing the mean squared error metric, this one treats outlier related errors evenly regarding the magnitude. MAE formulation is shown in Eq. 6. They are different in terms of the degree to which errors are penalized. (Roy, Banerjee, Sarkar, Darwish, Elhoseny, & Hassanien, 2018)

$$MSE = \frac{\sum|(p_{ui} - a_{ui})^2|}{n} \tag{5}$$

$$MAE = \sqrt{\frac{\sum|p_{ui} - a_{ui}|}{n}} \tag{6}$$

Root mean squared error (RMSE) is the most popular metric. The measurement relies on how close are the predictions to the user ratings. RMSE equation is shown in Eq. 7. The system generates predicted ratings p_{ui} for a test set consists of n user item pairs (u, i) , for which true ratings a_{ui} are known, they may be obtained from an offline experiment. (Roy et al., 2018)

$$RMSE = \sqrt{\frac{\sum(p_{ui} - a_{ui})^2}{n}} \tag{7}$$

Relative squared error (RSE) is also alternative measurements. As shown in Eq.8, they can be calculated by normalizing the actual and predicted values by subtracting the average actual values.

$$RSE = \frac{\sum(p_{ui} - a_{ui})^2}{\sum(a_{ui} - \bar{a})^2}, where\ \bar{a} = \frac{1}{n}\sum\nolimits_{u,i} a_{ui} \tag{8}$$

Correlation coefficient measures the statistical correlation between actual and predicted values. Correlation coefficient, as displayed in Eq. 9, is a function of sample covariance (S_{PA}) and standard deviation of actual and predicted ratings (S_P, S_A). It is different from other metrics because it scales independent. Also, good performance results in larger value of coefficient. (Witten et al., 2016)

$$\frac{S_{PA}}{\sqrt{S_P S_A}}, where\ S_{PA} = \frac{\sum_{u,i}(p_{ui} - \bar{p})(a_{ui} - \bar{a})}{n-1} \tag{9}$$

A learning recommendation model is not simply defined as the one with lowest error rate. Performance of a machine learning algorithm is determined by two factors; small training error and small gap between training and test error. These two factors correspond to two challenges: underfitting and overfitting. Underfitting is the inability of training a model with sufficiently low error value. Overfitting is high gap between test and training error. (Goodfellow, Bengio, & Courville, 2016) An underfitting model shows high bias while the overfitting shows high variance. Bias and variance of a model should be balanced by performing a trade-off. Bias is an error caused from the incorrect assumptions of learning algorithm. Variance is the error of high sensitivity to insignificant patterns in training data. High bias results in a model missing the relevant relationship between features and expected output. High variance results in a model shaped by random noise of data. (Fortmann-Roe, 2012)

Usage of a recommendation system also should be measured to evaluate the outcome. A single prediction machine has four different possible outcomes, the true positive and true negative are correct predictions which are consistent with real fallout, while false positive and false negative label incorrect predictions which are opposite of real output.

- A false positive occurs when the outcome is predicted as positive, when it is realized as negative. Total number of false positive predictions is represented as fp.
- A false negative occurs when the outcome is predicted as negative, when it is realized as positive. Total number of false negative predictions is represented as fn.
- A true positive occurs when predicted outcome and real fallout is consistent and positive. Total number of true positive predictions is represented as tp.
- A true negative occurs when predicted outcome and real fallout is consistent and negative. Total number of true negative predictions is represented as tn.

The overall success rate and error rate are given in Eq.10 and Eq.11.

$$Success\ Rate = \frac{(tp + tn)}{(tp + tn + fp + fn)} \tag{10}$$

$$Error\ Rate = 1 - Success\ Rate \tag{11}$$

In a broad explanation, usage prediction metric is the probability that recommended item is selected. There are four possible results of a recommended item. It is also called as retrieval confusion matrix.

Table 2.1: Retrieval Confusion Matrix

	Selected	Not Selected
Recommended	True Positive	False Positive
Not Recommended	False Negative	True Negative

Precision, Eq.12, is the probability that selected item for a user is relevant.

$$Precision = \frac{tp}{tp + fp} \tag{12}$$

Recall, Eq. 13, is the probability that relevant item will be selected. It is also called as true positive rate.

$$Recall = \frac{tp}{tp + fn} \tag{13}$$

Precision and recall curves represents the relative cost of false positives. Precision and recall makes sense for relevance problems. Information retrieval and ranking problems are evaluated by these metrics. Regarding a search engine problem, precision is the number of relevant selections over total number of selections. Recall is number of relevant selections over total relevant selections.

ROC or Receiver Operating Characteristic curves emphasize the volume of recommended but not preferred items. (Ricci et al., 2011) ROC curves visualize false positive rates. ROC stands for receiver operating characteristics which is used in signal detection to outline the trade-off between hit rate and false alarms over a channel. In other words, ROC curve visualizes the true positives and false positives as data pairs.

F-measure, as known as F_1 is also used in information retrieval. As shown in Eq. 14, It is an expression of precision and recall.

$$F_1 = \frac{2 * Precision * Recall}{Precision + Recall} \tag{14}$$

Sensitivity and specificity are commonly used in medical experiments. Sensitivity is the proportion of people with disease having positive test results. It is also known as true positive rate, recall or hit rate. Specificity is the proportion of people without disease having negative test results. It is also known as true negative rate or selectivity. The product of sensitivity and specificity is also a measure. (Witten et al., 2016)

The application may offer to the customer a list of item. Ranking the recommendation properly can be a core feature for some businesses such as Youtube and Netflix. There are two metrics to evaluate the ranking. One for sorting the items as customer would expect as it is. One for the utility of system's sorting to the user. Normalized Distance based Performance Measure compares an ideal predefined ranking to the systems ranking. Normalized Discounted Cumulative Gain measure assumes that top position is the crucial and users wouldn’t care the items at the end of the list. (Xiong, Wang, Zhang, & Ma, 2018)

2.2 Machine Learning

A recommendation system is regarded as successful when it provides relevant content to active users. Rapidity of customer activity urges the service providers to process the behavioural data as fast as possible. This demand brings in machine learning and deep learning concepts. A learning recommendation service deals with big data and optimizes the error for relevance.

In this section several machine learning concepts and applications are acknowledged. Several big data matters are mentioned. To precede with machine learning concepts, learning types and algorithms are defined and expanded with several academic studies.

2.2.1 Concepts and Applications

Machine learning is a concept of artificial intelligence which capacitates a system or a business model for learning from data. Traditional business models built with explicit programming can be empowered with machine learning outputs. It uses a variety of algorithms which learns from data to improve, identify or predict possible conclusion. Considering the fact that a learning machine computes and interprets a training data, it is eventually expected that prediction models produce more precise outcome.

Roughly, a business model with machine learning has two core components. First one is a machine learning model trained by a learning algorithm and data. For example a predictive algorithm will yield a predictive model. Second one is the predictions generated by the predictive model based on the data which trained the model. Given that digitalization is a part of almost every business model, Machine learning is unavoidable and essential for creating analytics model. (Hurwitz & Kirsch, 2018)

Scientists and companies needed learning and improving machines on the foundation of their experience, one of them is complexity and the other one is agility. Tasks performed by human and beyond human capacities are too complex to program. Examples like driving, speech recognition or image processing programs achieves better results by learning from sufficiently enough training sets. Machine learning is also employed to detect meaningful patterns from digital data which are too large for human to interpret. Learning tools are advantageous for adapting their behaviour despite to the rigid unchanged programs. (Shalev-Shwartz & Ben-David, 2014)

It is very noticeable that, data is generated by everyone at every moment from almost every human machine interaction. Big data is quite prominent regarding the booming innovations in science and engineering. As a result, there is a potential to learn from this vast amount of data. Intelligent solutions are solely obtainable with discovering the underlying structure of data sources and learning from the data. Big data is characterized with volume, variety and value. The ambition of big data processing is to build a decision support system which provides objective judgement from existing events. (Fathi, Abghour, & Ouzzif, 2017)

Over the past years, machine learning applications have been extensively accepted in complex scientific and commercial fields. As the collection of data is so large and complex, traditional machine learning applications are not able to meet the need of mining the information hidden in the data. Notwithstanding the achievements in machine learning with the rise of big data, there are major challenges to be addressed.

The first critical issue is the large scale of data. Machine learning algorithms are supposed to be trained with distributed sources in terms of processors and storage. There is a variety of frameworks to accomplish parallel and distributed large scale data processing. Alternating direction method of multipliers is able to split a large scale global problem into smaller sub problems. Map-reduce is another powerful framework for tackling large data sets.

Second critical issue is to learn from different types of data. In general, data comes from different sources in different structures, sometimes even entirely unstructured. Learning from such data is a complex problem and a great challenge. Data integration is proposed the user a unified view of different data shapes. Dimensionality reduction is proposed to deal with high-dimensional data. Principal component analysis, linear discriminant analysis, locally linear embedding and Laplacian Eigenmaps are typical machine learning algorithms for data dimensionality reduction.

Third critical issue is learning velocity. If the business model requires time sensitive insight, data value mostly depends on freshness and real time processing. Sequential learning models are proposed as machines cannot hold entire dataset in memory. It is named extreme learning machine.

Fourth issue is learning from uncertain and incomplete data. Data uncertainty can be handled by simple statistical methods to data samples. For data incompleteness, an advanced deep learning method is proposed to handle data noise.
Fifth issue is learning from data with low value density. Knowledge discovery in databases and data mining technologies are pointed out for finding value in diverse set of big data. (Qiu, Wu, Ding, Xu & Feng, 2016)

One of the most popular application of machine learning is ranking which we confront almost every content based application. Web page ranking is the process of finding the content relevant to the search query and ranking them in order of relevance. A search engine should learn and adapt to know which pages are relevant.

Collaborative filtering is another application. Collaborative filtering is very much alike to ranking problem. The distinction is the explicit query. We can interpret the past purchases and viewing decisions of the users to predict future interactions. The key point in collaborative filtering is the information provided from similar users. The same principles are indifferent for most social networking applications in terms of ranking the content for their users' intentions and seeks. Machine learning replaces the guesswork and assumptions with automated relevance design.

Automatic language translation is also a major field of machine learning. Face recognition and access control systems are considered are classification problems. Speech recognition, handwriting recognition, failure detection are several other applications empowered with machine learning. Another application supported by learning is named entity recognition. This is crucial for understanding the texts in a meaningful context by identifying places, titles, names etc. (Smola & Vishwanathan, 2008).

2.2.2 Learning Types and Algorithms

Learning algorithms are adequately implemented in variety of applications including spam detection, natural language processing, speech recognition, computer vision, fraud detection, recommendation systems and so forth.

Some major learning problems are mentioned below:

- *Classification:* Each item is assigned to a category or label. The number of categories are usually small and it can be large or unbounded depending on the application.
- *Regression:* Real value of an item is predicted. Incorrect predictions are encountered with penalty depending on the significance of difference.
- *Ranking:* Items are ordered by a rule set.
- *Clustering:* Items into insignificant regions are partitioned. Clustering is applied for analysing large data. For example, social media networks are analysed by clustering in an attempt to identify communities.
- *Dimensionality reduction:* Items with high number of features are transformed into a lower dimensional representation. Initial data can be compressed as a pre-processing step for following operations. Alternatively, a smaller and more useful set of features can be generated. (Mohri, Rostamizadeh, & Talwalkar, 2012) It is also called as manifold learning.

Learning Types are detailed below:

- *Supervised Learning:* In supervised learning, the training data contains explicit examples of expected correct output for a given input. In other words, it is already provided to the algorithm what to predict. (Harrington, P., 2012) Considering a handwriting recognition example, training data already includes images and actual digits. The learning is already supervised in terms of paired image and digit data. (Abu-Moustafa, Magdon-Ismail, & Lin., 2012)

- *Active Learning:* Active learning is based on the hypothesis that if a learning algorithm is allowed to choose data from the learning source, then it will perform better. Considering the sophisticated supervised learning data with thousands of labels and instances like speech recognition, learning systems may consume unexpected sources and time. Active learning systems ask a supervisor to label unlabelled queries. This systems can be resourceful when the labels are sparse or very expensive to obtain. (Settles, 2009)

- *Unsupervised Learning:* The data prepared for an unsupervised setting does not contain any output information. Considering a clustering problem, a model trained with unsupervised learning may generate the same clusters compared to the output of supervised model. However, unsupervised learning will yield results without labels and number of clusters may be more unclear. In other words, unsupervised learning is finding the patterns and structure in input data instinctively. (Abu-Moustafa et al., 2012)

- *Reinforcement Learning:* It is a behavioural learning model. The algorithm receives feedback from the environment. The model is provided with a feedback after every action. It is different from supervised learning in terms of training data. This system learns from the trial. Successful decisions will be reinforced the process. This learning algorithm discovers the relationship between successful results and the sequence of events that leads the successful outcome. (Hurwitz & Kirsch, 2018)

- *Online Learning:* In online learning, prediction and training phases are not separated. Each time an action is happened, it is considered a test example. A result for every decision is predicted. When the result of decision is known, the decision is labelled. Eventually every decision is used as a training example, which improves the prediction mechanism for the future. (Shalev-Shwartz & Ben-David, 2014)

- *Deep Learning:* Real world machine learning problems confronts the difficulty as variety of factors leverages the data which we are able to observe. Most applications requires those factors to be untied and some factors should be eliminated. Obviously this task is not an effortless one. It is almost impossible to obtain a sophisticated representation of learning model by human understanding. Deep learning solves this problem by building a representation consists of several simpler learning representations. This difficulty is addressed by breaking the sophisticated data-result mapping into a series of nested simple mappings. Each element of this mapping is called as layers. The data is provided into the visible layer of the model. Then the data is passed to hidden layers in terms of abstract features. Deep learning is a classification of machine learning which is empowered by flexibility to learn to represent the problem as a nested hierarchy. (Goodfellow et al., 2016)

Learning Algorithms proposed for Classification are acknowledged below:

- *k-Nearest Neighbourhood:* The idea of nearest neighbour algorithms is to predict the label of new instance on the basis of the labels of its closest neighbours in training set. The assumption of such method is clusters are described by the features which are relevant to the labels of centre points. Therefore, neighbouring points are expected to have same labels. (Shalev-Shwartz & Ben-David, 2014)

- *Multi-class classification:* Real world classification problems consist of more than one class rather than binary. Decision trees address the problem of multi-class classification. Carrying out large number of classes can be computationally problematic and increasing the time complexity. Unbalanced classes can become a challenge in the matter of learning. Classifier can be trained with a balanced training data which implies a smaller sample. Otherwise, when a large portion of training data belong to one class, the model tend to return the same class for all instances. Another learning related issue is the hierarchical relationship between classes. Hierarchical relationships should be handled as a rich and complex problem.(Mohri et al., 2012)

- *Bayesian Decision Theory:* Bayesian probability theory is a mathematical reasoning and inference framework. In Bayesian probability theory judges the relative truth of the hypothesis given the data. The likelihood function assesses the probability of the observed data originating from the hypothesis. Prior function is prior knowledge before the data. Posterior function is the probability of hypothesis after the knowledge of data. The transformation of prior to posterior represents learning. The posterior reflects what is learned about the validity of hypothesis from examination of data. (Smola & Vishwanathan, 2008) Naive Bayesian networks are composed of directed acyclic graphs with only one parent and couple independent child nodes. Bayes classification algorithm simply computes the probability of item having a certain label. This network has limitation that each feature can be related to only one other. (Kotsiantis, Zaharakis, & Pintelas, 2007)

- *Logistic Regression:* Logistic regression is a widely used binary classification algorithm. The probability that the label of an item is interpreted. The hypothesis class associated with logistic regression is the composition of a sigmoid function. It is called logistic function. Given the logistic hypothesis is not sure about the value of the label, a loss function is specified. This loss function is placed to define how bad the label estimation is. Therefore loss function is aimed to be minimized within the training. (Shalev-Shwartz & Ben-David, 2014)

- *Support Vector Machines:* Support vector machines are one of the most effective classification algorithms in modern machine learning. Considering the items are separated enough to exist as subgroups, the line used as a decision boundary is called hyper-plane. Every item at one side of a hyper-plane belongs together as one class. The closest point the separating hyper-plane defines the margin. The greater the margin is, the tolerable the mistakes are. The closest points are known as support vectors, The goal function of a support vector problem is to maximize the margins regarding the items are separable.(Harrington, 2012) A good decision boundary achieves an optimized balance when total margin based loss for samples within the margin is minimized and maximum separation between training samples are achieved. (Cherkassky & Mulier, 2007)

- *Boosting and AdaBoost:* Boosting is a learning method based on the idea that combination of simple classifiers can perform better than a simple classifier alone. Simple classifiers produce results with probability error slightly less than random guessing. They are called weak learners. The strategy of boosting is briefly building an ensemble of weak learner classifiers instead of building a single strong classifier. Typically, a weak learning algorithm is trained repeatedly with different weighted versions of a training data and several weak classifiers are generated. Weighting of ach training data set depends on the accuracy of previous classifier. Therefore algorithm are allowed to focus on incorrectly classified samples. AdaBoost or adaptive boosting is the best-known algorithm. AdaBoost uses the differently weighted versions of same training data. (Zhang & Ma, 2012)

Learning Algorithms proposed for Regression are acknowledged below:

- *Linear Regression:* The learning problem of regression consists of using data to predict the correct values of items as closely as possible. As regression is a supervised learning problem, labeled samples are provided as training data, in terms of real numbers. Yet, we cannot expect the precise prediction of correct label. Learner should provide predictions as close as the correct ones. Regression is different from classification in terms of the measure of error. It is based on the difference between predicted label and correct one. Linear regression consist of seeking a hypothesis which minimizes the mean square error. (Mohri et al., 2012) Multi response linear regression is also used in several ensemble learning models. Combined with association rules, multi response linear regression is implemented for calculating the product based purchase probability of a customer. A Meta learning model is designed and the output of learning model is propagated to a linear regression model. Thereby, individual poor classifiers are associated with a multi response linear regression model to improve the model precision. (Kasap, Ekmekci, & Ketenci, 2016)

- *Tree Based Regression:* A classification tree is a special form of classifier where each disjoint piece is a union of sets recursively partitioning the item space. This allows the classifier to be represented as a decision tree. A regression tree is similarly a tree structured solution. A regression tree algorithm has three major tasks. first one, to partition the data at each step, second one, when to stop partitioning, third one to predict labels for items at each partition.(Loh, 2008) Regression trees are insensitive to outliers and able to handle missing predictors. Classification and regression tree contains a regression method which builds a decision tree based on a partitioning algorithm which repeatedly splits data until the groups are homogenous. In order to avoid overfitting, the number of leaves of decision tree is balanced by cutting the less important nodes. This is called pruning. (Naghibi, Pourghasemi, & Dixon , 2015)

Learning Algorithms proposed for Clustering are acknowledged below:

- *k-means clustering:* K-means is an algorithm which locates k clusters within a given data set. The number of clusters is pre-defined. Each cluster is represented by a data point called centroid. Centroid is the centre of the cluster. Centroids are randomly assigned at the beginning of the algorithm. Then data points are assigned to the clusters. This assignment is distance based. Then centroids are updated as centre of clusters. The quality of cluster assignments are measured by sum squared error of clusters. (Harrington, P., 2012)

- *Linkage-Based Clustering:* This algorithm starts from a trivial clustering in which every data point is a single point cluster. Then the algorithm merges the closest clusters as one cluster. There are two success metrics to terminate the algorithm before generating one large cluster. The distance between clusters are determined. Also when to stop merging is predefined. There are two options of stopping criteria. One of them is fixed number of clusters and the other one is an upper bound for between-clusters distance. (Shalev-Shwartz & Ben-David, 2014)

- *Principal Component Analysis:* Principal component analysis is a technique which is used for applications such as dimensionality reduction, feature extraction, data compression or visualization. PCA seeks a space of lower dimensionality. It is also called as principal subspace. It is the orthogonal projection of data points. The variation of projected points is maximized on this subspace. Basically, principal component analysis technique is based on minimizing the sum of squares of the projection errors. This is the average cost of projection. (Bishop, 2006) A kernel PCA computes the principal components in a high dimensional feature set, which is related to the input space. There are several manifold learning techniques as nonlinear methods for dimensionality reduction. There algorithms assume that high dimensional data consists a low dimensional nonlinear manifold. It is aimed to learn this manifold design by finding a low dimensional space. Isomap algorithm aims to preserve the distance along the manifold between all pairs of data points. Laplacian eigenmaps are preserving local neighbourhood relationships in high dimensional space. (Mohri et al., 2012)

3 METHODOLOGY

3.1 Proposed Framework Overview

Nowadays, e-commerce platforms accommodates high diversity of choice for high number of visitors. Under these circumstances, there are two anticipated challenges. One of them is to expedite the decision making process for the prospective customers and the other one is scaling the technological solutions for high demand.

Recommender systems are cut-out for sorting. There are several methodologies which are applicable for recommendations. One of them is content based which requires human work for characterising product and user content. The other one is collaborative filtering. CF produces recommendations based on previous customer preferences. Since customer preferences are essential for recommender systems and customer taste is not an objective phenomena, collaborative filtering is proposed in this study.

One component of a recommendation system is the data source. Customer feedback data is an essential information. In collaborative filtering, decisions of customers are considered as analytical judgement and they are shared to others. It is not possible to design a model of customer taste. It is hidden in customer behaviour. Therefore the impressions and behaviours of customers are converted into implicit feedback data.

Other component is the recommendation prediction model. Implicit feedback does not reflect the preference of a customer, it is the strength of confidence of customer for a product. Therefore, latent factor method is preferred over user or item neighbourhood based collaborative filtering. Users and items are represented as vectors of latent factors.

Since implicit feedback is a sum product of a product and user, latent factors are established to uncover the user-item association. The collaborative filtering model is formed as a singular value decomposition problem and trained with implicit feedback data. Weighted λ Regularization method is applied to avoid overfitting without having a scalability problem. Also, a confidence level is added to feedback data and the ratings retrieved from implicit feedback is converted to a value including the confidence parameter alpha and rating value. The SVD problem is solved by alternating least squares method which routes and solves the optimization function for both users and users for the given number of iterations.

Model parameters and number of factors are fine-tuned until observing the minimum mean squared error. Then the model is generated. Predictions are generated by the model for the predefined customer-product pairs. These pairs are filtered in accordance with business objectives and customer intentions. Following the predictions, recommendations are also tailored and limited as total number of 10.

The last step is, eventually, evaluation of recommendations. At last, prediction and recall metrics are calculated. These steps are visualized in following flow, Figure 3.1.

Ultimately, this study aims to build a recommender system for an e-commerce platform which is targeting the prospective customers, empowered by customer behaviour, up to date, reasonable yet unforeseen and personalized.

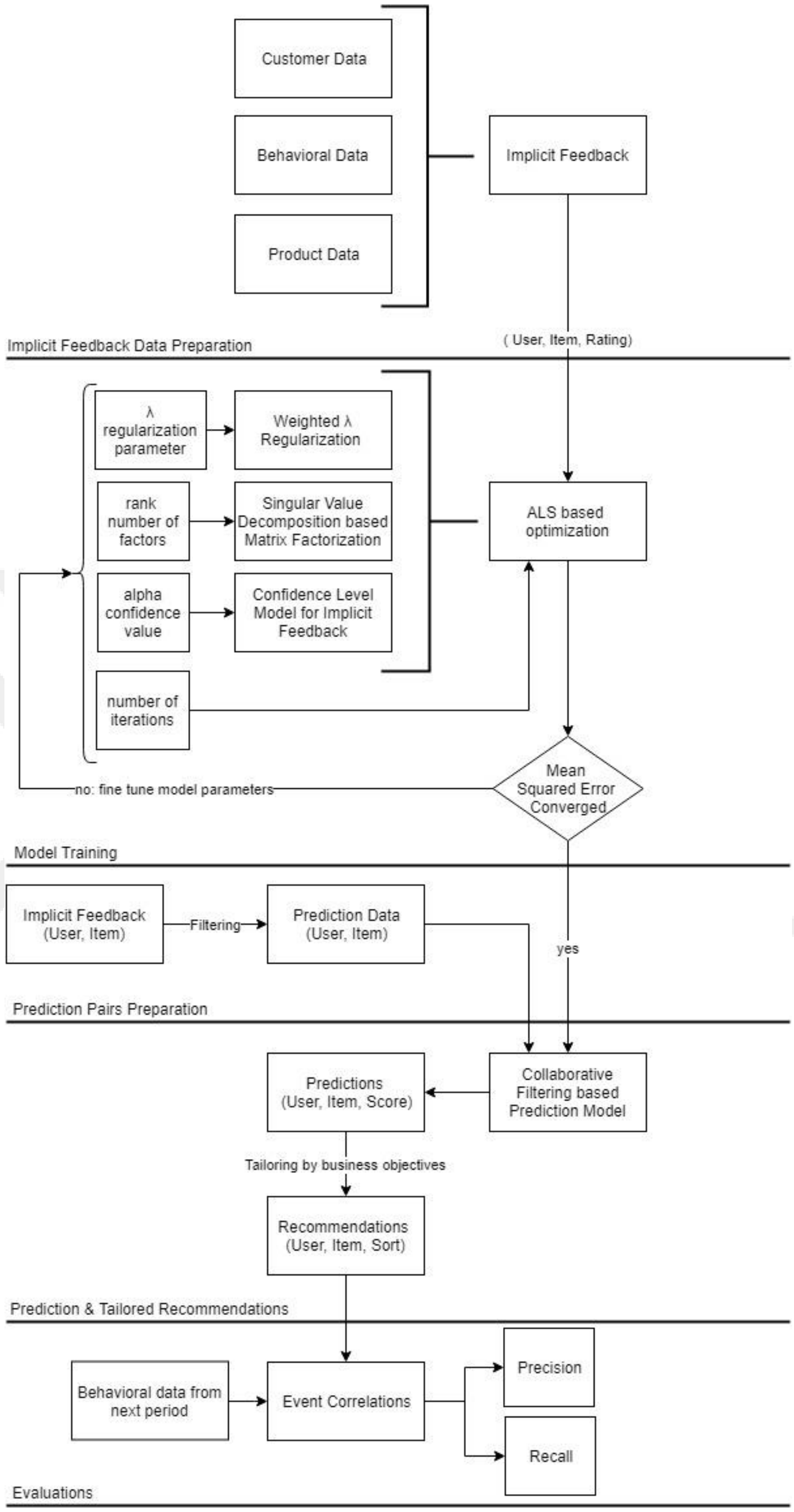

Figure 3.1: Proposed Recommendation Framework

3.2 Alternating Least Squares for Optimizing Singular Value Decomposition Problem

Matrix factorization is a method used for latent factor models. It characterizes users and items as vectors of factors inferred from item rating patterns. A valuable recommendation carries high correspondence within user and item factors. This method is preferred for two reasons, it is scalable and accurate in predictions. Matrix factorization models fit users and items into a latent factor space of dimensionality. User-Item interactions are considered as inner products in this space. Singular value decomposition is a technique for identifying latent semantic factors in information retrieval. In collaborative filtering, it is applied to the user-item rating matrix. To learn the factor vectors, the model minimizes the regularized squared error on the set of known ratings.

The learning model is generated by fitting the previous quantitative implicit feedback data in terms of ratings. The overall goal of a model is to reuse the model for unknown rating predictions. The minimization problem has two unknowns of the optimization goal, therefore one of the unknowns is fixed to solve the problem. Alternating Least Squares rotates the problem by fixing feature vector of users and items sequentially. The least squares computation problem is solved and regularized squared error is decreased until convergence. (Koren, Bell, & Volinsky, 2009)

3.3 Tackling Overfitting with Weighted λ Regularization

Overfitting is considered as overtraining a model by feeding with noisy and inaccurate data. Some machine learning algorithms have more freedom to build a model based on the given training data, therefore we may end up with an unrealistic model. Regularization is implemented to reduce the variance of model without increasing the bias. Bias is the error of the model. Variance is the change in predictions observed with different training models. Therefore a tuning parameter λ is added to model. As the tuning parameter increases, it reduces the value of coefficients and variance. The regularization parameter is selected carefully for finding the balance. It is usually selected by cross-validation.

An ALS with weighted λ regularization model is proposed for large scale collaborative filtering by Zhou et al. in 2008. The main purpose of weighted regularization is to ensure the model never overfits by increased number of features (latent factors) or iterations.

$$F(U,M) = \sum_{(u,i)\in\mathcal{K}} (r_{ui} - q_i^T p_u)^2 + \lambda\left(\sum_i n_{q_i}\|q_i\|^2 + \sum_u n_{p_u}\|p_u\|^2\right) \qquad (15)$$

Notations are listed below:

$u: user$

$i: item$

$\mathcal{K}: user - item\ pairs\ in\ training\ data$

$r_{ui}: rating\ of\ user\ for\ an\ item$

$\hat{r}_{ui}: estimated\ rating\ of\ user\ for\ an\ item, q_i^T p_u$

$q_i: latent\ factor\ vector\ of\ item\ i, q_i \in R^f$

$p_u: latent\ factor\ vector\ of\ user\ u, p_u \in R^f$

$\lambda: regularization\ factor$

$n_i: number\ of\ items$

$n_u: number\ of\ users$

$n_{q_i}: number\ of\ ratings\ of\ item\ i$

$n_{p_u}: number\ of\ ratings\ of\ user\ u$

$I_u: Set\ of\ items\ that\ user\ u\ rated$

$I_i: Set\ of\ users\ that\ rated\ item\ i$

$Q: item\ feature\ matrix$

$P: user\ feature\ matrix$

$R: user - item\ matrix, \{r_{ui}\}_{n_u \times n_i}$

$Q_{I_u}: Submatrix\ of\ Q,\ i \in I_u$

$R(u, I_u): ratings\ row\ vector\ of\ items\ that\ user\ u\ rated$

$P_{I_U}: Submatrix\ of\ P,\ u \in I_i$

$R(I_i, i): ratings\ column\ vector\ of\ users\ that\ rated\ item\ i$

n_f: $feature\ dimension\ space$

E: $n_f \times n_f$ $identity\ matrix$

In order to minimize the regularized squared error, ALS is applied. The minimization problem has two sets of decision variables as part of the optimization goal, therefore as one of the decision variables set is fixed to solve the problem for the remaining set the problem is solved. As mentioned in the previous section, ALS rotates the problem by fixing item latent factors and user latent factors sequentially. The least squares computation problem is solved and regularized squared error is decreased until convergence. ALS is preferred over gradient descent as it can use parallelization.

Objective function of the model is stated in Eq. 15. Matrix $Q = [q_i]$ is initialized by assigning the average rating for an item as the first row, and small random numbers for the remaining entries. Then Q is fixed and $P = [p_u]$ is solved by minimizing the sum of squared error in the objective function. Then P is fixed and Q is solved similarly. This rotation is repeated until the mean squared error convergence.

A given column of P, which latent factor vector of user u denoted as p_u, is determined by solving a regularized linear least squares problem involving the known ratings of user u and feature vectors q_i of the items that user u rated. p_u becomes an expression of Eq. 17 and Eq.18, which is given in Eq. 16.

$$\frac{1}{2}\frac{\partial f}{\partial i_{kj}} = 0, \qquad \forall u, k$$

$$\Rightarrow \sum_{i \in I_u} (p_u^T q_i - r_{ui}) q_{ki} + \lambda n_{p_u} p_{ku} = 0, \qquad \forall u, k$$

$$\Rightarrow \sum_{i \in I_u} q_{ki} p_u^T q_i + \lambda n_{p_u} p_{ku} = \sum_{i \in I_u} r_{ui} q_{ki}\,, \qquad \forall u, k$$

$$\Rightarrow \left(Q_{I_u} Q_{I_u}^T + \lambda n_{p_u} E\right) p_u = Q_{I_u} R^T(u, I_u), \qquad \forall u$$

$$p_u = A_u^{-1} V_u, \qquad \forall u \tag{16}$$

$$A_u = Q_{I_u} Q_{I_u}^T + \lambda n_{p_u} E \tag{17}$$

$$V_u = Q_{I_u} R^T(u, I_u) \tag{18}$$

Q_{I_u} denotes the sub-matrix of Q (item feature matrix) consisting of columns $i \in I_u$ (set of items rated by user u). $R(I_i, i)$ denotes the row vector retrieved from $u_{th}\ row\ of\ R$ (user-item matrix) for $i \in I_u$ (set of items rated by user u). Similarly, when Q is updated, individual q_i can be computed via regularized linear least squares solution including the feature vectors of users who rated item i. q_i becomes an expression of Eq.19 and Eq.20, which is given in Eq. 21.

$$q_i = A_i^{-1} V_i, \qquad \forall i \tag{19}$$

$$A_i = P_{I_i} P_{I_i}^T + \lambda n_{q_i} E \tag{20}$$

$$V_i = P_{I_i} R^T(I_i, i) \tag{21}$$

P_{I_U} denotes the sub-matrix of P (user feature matrix) consisting of columns $u \in I_i$ (set of users rated item i). $R(I_i, i)$ denotes the column vector retrieved from $i_{th}\ column\ of\ R$ (user-item matrix) for $u \in I_i$ (set of users rated item i). (Zhou, Wilkinson, Schreiber, & Pan, 2008)

3.4 Confidence of Implicit Feedback

As suggested by Hu et al., in this stage we tried to identify the unique properties of implicit feedback data. t_{ui} is a binary set which indicates the preference of user u for item i. In other words, if user u is interacted to item i, t_{ui} is equal to 1. On the other hand, if user u never confronted item i, then active user's preference is set equal to 0. Preference values are poor in confidence, as having no preference may have variety of reasons other than not liking an item. Thus a confidence level model representing the user's preference is required. Consequently, as r_{ui} grows, the strength of preference should be increased. c_{ui} is measurement for the confidence in t_{ui} equals ($1+ \propto r_{ui}$).
The squared error part of goal function $(r_{ui} - p_u^T q_i)^2$ is extended as $c_{ui}(t_{ui} - p_u^T q_i)^2$.

$$A_u = Q_{I_u} C^u Q_{I_u}^T + \lambda n_{p_u} E \tag{22}$$

$$V_u = Q_{I_u} C^u R^T(u, I_u) \tag{23}$$

$$A_i = P_{I_i} C^i P_{I_i}^T + \lambda n_{q_i} E \tag{24}$$

$$V_i = P_{I_i} C^i R^T(I_i, i) \tag{25}$$

Replacing the ratings with confidence values, A_u, V_u, A_i and V_i are updated as seen in Eq.22, Eq.23, Eq.24 and Eq.25. (Hu et al., 2008)

3.5 Evaluation of Recommendations

Confusion matrix scheme is widely used in recommendation system evaluations. It is the intersection of recommendations and customer preferences. Aforementioned, four dimensions are represented in a typical confusion matrix. The elements of confusion matrix is redefined in this context.

Table 3.1: Proposed Retrieval Confusion Matrix

		Customers	
		Interacted	Not Interacted
Products	Predicted & Recommended	True Positive	False Positive
	Predicted & Trimmed	False Negative	True Negative

In this study, recommendations are generated for certain customers, as mentioned as prospective customers. These customers are tracked within the next period.

- Customers who interacted a recommended product are considered as true positive.
- Customers who interacted a predicted but trimmed product are considered as false negative.
- Customers who didn't interact any recommended products are considered as false positive.
- Customers who didn't interact any predicted but trimmed products are considered as true negative.

Customer interaction status types and product states are crossed and results are visualized in Table 3.1.

Operating these parameters, precision, recall and success ratios are calculated. Precision is the ratio of true positives over predicted positives. This ratio represents the engagement the tailored recommendations. Recall is the ratio of true positives over actual positives. This ratio represents the coverage of tailored recommendations.

3.6 Running the Model as an Application

Apache Spark is a unified analytics engine for large scale data processing. It can run standalone or in the cloud. Projects can be developed in Java, Scala, Python, R or SQL.

The main advantage of Spark is being a resilient distributed dataset. It provides parallel operations within the nodes of a cluster. Resource allocation is also efficiently handled by variable broadcasting.

Spark contains a machine learning library. There is also a collaborative filtering solution since it is commonly used for recommender systems. A model training function is constructed. This function is tailored for a latent factor model based collaborative filtering. It is trained by implicit feedback data. Regularization is also scaled for solving the least squares problem for large data sets, as the training is supposedly minimizing the mean squared error.

The trainimplicit function is used for model generation. This function trains a matrix factorization model by using the implicit feedback in form of (User ID, Product ID, Ranking) pairs. Aforementioned, this ranking is considered as product of two low-rank matrices of given rank.

Docker is preferred for containerization. Docker is a tool allowing us to package this application as a whole. In other words, this application becomes platform independent which runs on any machine and operating system.

Three parameters are fine tuned in this study. Since scaled regularization is used, lambda is optimized by model training function. Rank parameter represents the number of features. Number of iterations is the number of epochs for ALS routing. Alpha is used as a baseline confidence of preference strength. It is applicable for implicit feedback.

3.7 Data Requirements

Implicit feedback demonstrates the preference of a customer for a product. It is generated by interpreting customer behaviour. Eventually, a customer has one rating for a product. This rating is a numeric value representing the strength of the confidence for ratings. The model requires numeric ratings to be increased with the strength of the confidence. Also customers and products are represented with numeric unique ID. Overall, implicit feedback tuples are converted into (User ID, Product ID, Rating) form. In this study, a purchase is perceived as a feedback with the strongest confidence. Likewise, when a product is displayed in a category listing page yet not even clicked, this action is perceived as a feedback with weakest confidence. This model is detailed in case study. Model training is completed with implicit feedback data.

Prediction data is also required as customer product tuples in form of (User ID, Product ID). Prediction pairs are filtered for the sake of business requirements. We stick with two principles, buyers should be introduced to different categories and assortment among categories should be represented. While the training is performed with 4 week period of data, predictions should be generated for the customers who visited the website within 24 hours.

In order to evaluate the recommendations, future behaviour of customers who are targeted for recommendations is tracked. Customers are classified over the confusion matrix. Therefore, required events for customers associated to predicted and recommended products are collected.

The data used in this work is provided by hepsiburada, one of Turkey’s leading e-commerce platforms. Regarding the EU general data protection regulation and Turkish personal data protection law, customer data and behavioural data is completely anonymised and processed with internal resources.

4 CASE STUDY

4.1 Preliminaries

In this application, collaborative filtering based recommendation engine is proposed to utilize for an online shopping platform. Behavioural raw data is provided from a real online store in an encrypted form in terms of customer privacy protection. Aforesaid platform hosts almost ten million unique visitors in daily basis and provides high variety of products on the scale of millions.

Forasmuch as the high number of visitors and products, recommendation engine is utilized for specific customers and product groups. Small domestic appliances (SDA) and fast moving consumer goods (FMCG) are taken into account. SDA consist of the kitchen appliances. Also, FMCG consist of beverages such as tea and coffee.

Collaborative filtering is a recommendation algorithm based on selecting, aggregating and sharing users' behaviour and ratings. In this context, it is based on analysing and interpreting the behaviour of people who already interacted with the products we plan to recommend. Customer's type of interaction with a product is an implicit feedback and it is interpreted as that customer's ranking for that specific product. A model is trained by this ranking set and future rankings are predicted.

People are represented and mentioned as users, products are called as items and customer behaviour is illustrated as implicit feedback of a user for an item.

There are four main stages of this study:

1. Implicit feedback data is prepared as numeric ranking provided by a user for an item.
2. Recommendation engine is trained.
3. Potential rankings for user-item pairs are predicted.
4. Ten items are selected for each user.

As an evaluation of predictions, users' interaction with selected items are observed. Precision and recall is calculated.

4.2 Business Model and Principals

The online store mentioned above beholds diverse array of customer behaviour in form of click view and impression data. Each traceable behaviour corresponds to an event in data hub. This detailed data breeds plenty of opportunities. Having this amount of behavioural data, both retrospective reporting and future predictions are achievable. In this study we will build a learning recommendation model trained by existing data. A recommendation engine predicts users' future or potential interests by computing and filtering user feedback of items.

An e-commerce sales funnel is described as the journey which leads potential visitors from discovery to purchase. Other than having high rate of e-commerce conversion as to make sales, retention is also a key success metric. Retention or customer loyalty results in long-established competitive advantage.

A conventional sales funnel consist of four steps. First, customer visits the online store. The internal discovery starts with the first step in. Second, product pages are visited, which is considered as a solid hint of interest. Third, customers starts checkout. Fourth, purchase is completed.

Overall goal of such sales cycle is to make a casual window-shopper actually pay for something. However each step of discovery is a phase of decision making which provides feedback about what the customer left behind to end-up with a certain product. Obviously the most trustworthy feedback from customer is to purchase an item and the most uncertain one is to overlook one.

Given the fact that the online store subject to this study provides different features for the customers. Therefore a custom sales funnel is designed to visualize the behavioural flow. There is corresponding traceable event for each step of the sales funnel. Also each type of customer-product interaction is portrayed by the step which they dropped off the funnel.

Users are anonymously identified for consistent tracking with respect to customer privacy regulations. Anonymous User ID's are unified for logged in customers. Products are represented by a unique stock keeping unit as known as SKU.

4.2.1 Behavioural Events

1. *ProductList*: This is a view event collected every time a user visits a product listing page. Listing pages can be a search result page, category page or campaign page.
2. *ProductView*: This is a view event collected every time a user visits a product detail page. Product discovery can start from product listing pages or product view pages.
3. *AddtoCart*: This is a click event collected every time a user clicks add to cart button. In other words, this event is generated when a product is added to shopping bag. Customers can add a product to shopping cart from listing pages directly or from product detail page.
4. *SaveforLater*: This is a click event collected when a product in shopping cart is moved to save for later list.
5. *CarrytoCart*: This is a click event. A user is able to carry a product to cart from save for later list.

6. *OrderSummary*: This is a view event. When a customer completes a purchase, order summary page is displayed. This event is collected when an order is completed.

All these events contain anonymised user and product information.

4.2.2 Customer-Product Interaction Portraits

A user usually confronts more than one product throughout the sales funnel. Therefore users have distinct portraits for each product. There is only one final interaction of customer and product. These portraits will be considered as implicit feedback.

Interaction Portraits:

1. *Not Interested:* The customer passed over the product on product listing page. ProductList is the only event we can observe for this customer-product pair.
2. *Curious*: The customer viewed the product detail page. ProductView is the most engaging event observed.
3. *Interested*: This customer added the product to the cart. AddtoCart is the most engaging event observed.
4. *Interested with second thoughts*: This customer is moved the item to save for later list. SaveforLater is the most engaging event observed.
5. *Doubtful lover:* This customer is moved the item from save for later to shopping cart. CarrytoCart is the most engaging event observed.
6. *Buyer*: This customer is reached to the end of the funnel. This customer is purchased the product.

A typical journey of a customer is to confront a product, to view the product, to add the product to the basket and to purchase the product. However different paths are also possible. All possible paths among the events and interaction portraits are visualized in Figure 4.1.

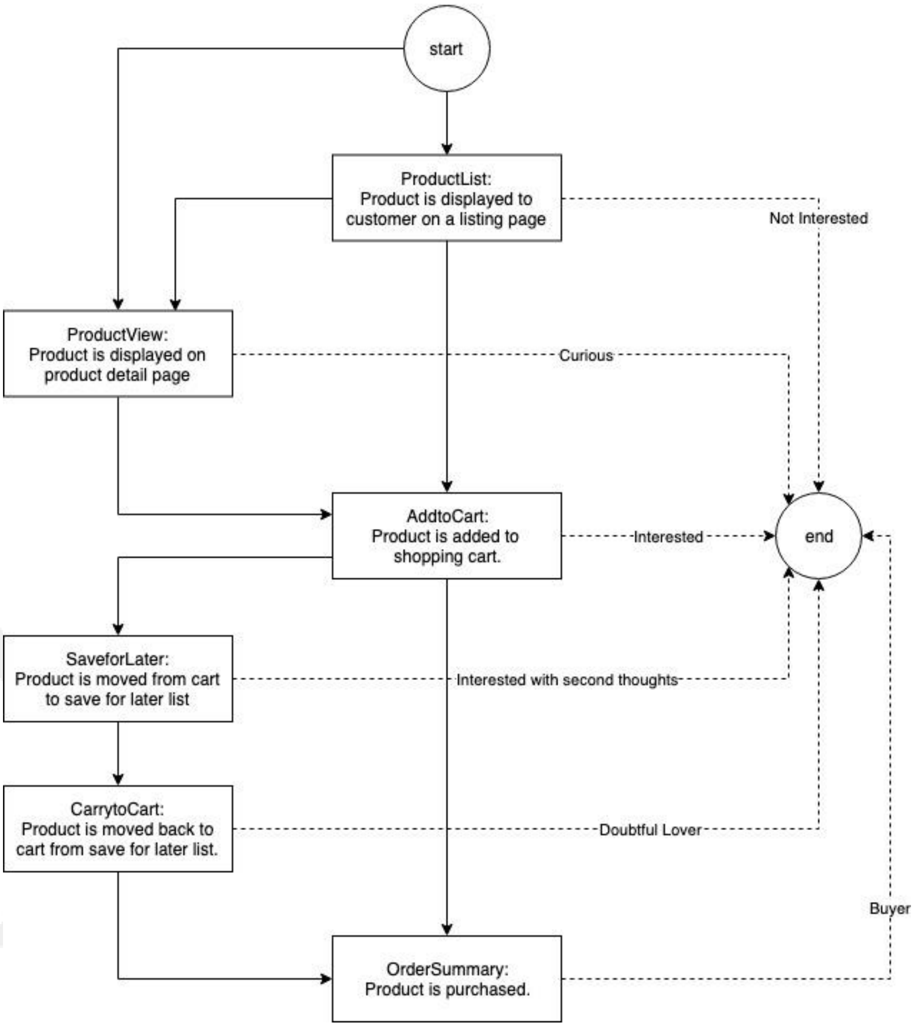

Figure 4.1: Events and Interactions Portraits

4.2.3 Product Spectrum

Two main departments are taken into account. One of them is small domestic appliances. Kitchen appliances are selected as electronic category. The other one is fast moving consumer goods department. Tea and coffee categories are selected as non-electronic category.

Products belong to one main category by definition. Besides, main categories are represented to customers in a nested category tree structure. This is a common navigation structure in e-commerce platforms.

Kitchen appliances department includes teapots, kettles, coffee makers, blenders, toasters, juicers, deep fryers, mini/midi ovens, steamers, food processors, toaster ovens, egg cookers. These categories are combination of more specific main categories. For example coffee makers category includes Turkish Coffee Makers, Filtered Coffee Makers, Espresso and Cappuccino Makers, Capsule Coffee Makers and Coffee Grinders.

FMCG department includes teas and coffees. These categories are also combination of more specific main categories. Products can be identified by their SKU, product category meta-data is found in different sources.

4.2.4 Prospective Customers

Since this online platform hosts multitude of visitors, users are filtered by their level of interactions. One should at least show an intention to an SDA product in a given time interval. Any type of interaction with the product spectrum generated by these users are interpreted for both model generation and prediction data.

4.2.5 Sales Funnel

Proposed sales funnel is visualized in Figure 4.2. It has three phases. First one is discovery phase. Customers enters the funnel from a discovery phase. Discovery phase includes product list and product view events. Customers who abandoned the funnel from discovery phase are also represented. These customers have loose interaction with the products as they didn't move the products to next stage. Customers are expected to drop out most of the products they interact in this phase.

Second phase is intention. Some customers decide to add the product to the basket. Although they add the product to shopping cart, it is not 100% positive that they will purchase those products. Yet, they are relatively more interested in the products they proceed to shopping cart. The users who didn't succeed to next phase are also represented. Third phase is purchase, these customers are successfully dropped from the funnel as they purchased something from our product spectrum.

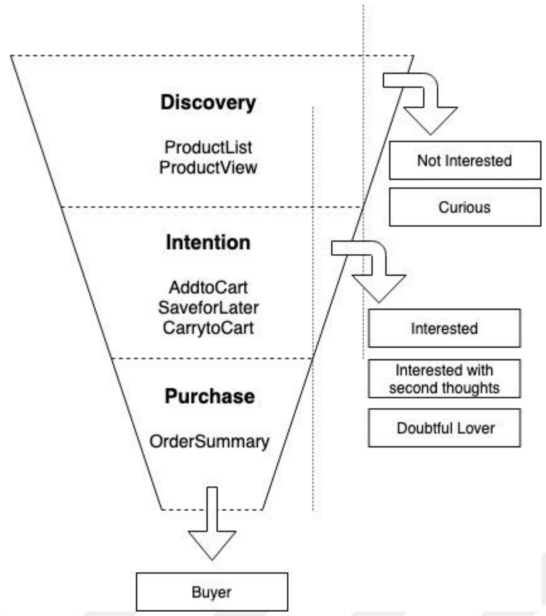

Figure 4.2: Proposed Sales Funnel

4.3 Implicit Feedback Model

Every customer who visited the online store is engaged in the sales funnel described above. Yet, how they dropped from the funnel is an implicit feedback source for our recommendation engine. An implicit feedback is generated from the data of actions of a customer. In this context, when a customer interacts with a product from our spectrum in given time interval, an implicit feedback from that certain customer to certain product is constructed. The more the customer carries the product to the end of the funnel, the more implicated the customer is. Nevertheless, abandoned products are also utilized as feedback sources. Every stage of sales funnel is named as an interaction portrait. The interaction portrait represents the feedback. Products are called as items and customers are called as users. Implicit feedback data is used for model generation.

Ground rules for implicit feedback data:

1. A product spectrum is generated for event filtering. Product spectrum includes products belonging to kitchen appliances and beverages.
2. Collected events are related to products in the spectrum.

3. Collected events are filtered with a time interval. 4 weeks of data is taken into account. Start date is selected as January 1st 2019 and end date is January 31st 2019.
4. ProductList, ProductView, AddtoCart, CarrytoCart, SaveforLater and OrderSummary events are listed in form of user and item.
5. Users are determined by the SDA intention rule.
6. Events are correlated by user and item. Therefore, interaction portraits are determined for every user-item interaction.
7. Interaction portraits are converted to numeric values from 1 to 6. Not interested is 1, curious is 2, interested is 3, interested with second thoughts is 4, doubtful lover is 5 and buyer is 6. Numeric rating values is directly proportional to the incline degree of the interaction portrait.
8. One user can have only one type of interaction portrait for an item and it is the larger one. Therefore user item pairs are unified. For example, a customer casually views the product and adds the same product to the shopping cart. The implicit feedback of customer for this item is the maximum value which is 3 in this case.
9. User IDs are normally in GUID form and items are alphanumeric values. They are converted into numeric IDs for model generation.

Obviously, a customer may interact different products and show different amount of interest for each product. An example is presented in Table 4.1. This customer purchased a filter coffee machine and interacted to different filter coffee machines and beverages.

Table 4.1: Implicit Feedback Example

Customer ID: 89e47c84-9804-4ced-8466-71b29bdd84d8			
Product Name	Main Category	Root Category	Interaction Portrait
Jacobs Monarch Aroma Filtre Kahve 2 x 500 gr + French Press	Filter Coffee	Beverages	Curious
Kiwi KCM 7540 Filtre Kahve Makinesi	Filter Coffee Machine	Kitchen Appliances	Curious
Kurukahveci Mehmet Efendi Colombian Filtre Kahve 500 Gram	Filter Coffee	Beverages	Curious
Sinbo Scm-2938 Kahve Makinesi	Filter Coffee	Kitchen Appliances	Curious
Arzum AR3046 Brewtime Filtre Kahve Makinesi - Siyah	Filter Coffee Machine	Kitchen Appliances	Buyer

A total of 174626 feedbacks are retrieved from 39926 users for 1403 different items. These users showed intention to or purchased at least one item from kitchen appliances category. 170937 of feedback is given for kitchen appliance products. 1103 of items belong to kitchen appliances category, which means the remaining 300 products belong to beverages category.

57930 of feedbacks are found as Not Interested and 67332 of them are Curious. Total of 125262 user-item pairs are dropped from the sales funnel at discovery stage. 34491 feedbacks are found as Interested, also 162 of them are Doubtful Lover and 464 of them are interested with second thoughts. Total of 35117 user-item pairs are dropped from funnel at Intention stage. 14247 of the feedbacks are found as Buyers.

Average number of impression generated by a user for an item or average number of feedback given by a user is 42. Total number of 1439 users are provided feedback for both kitchen appliances and beverages.

4.4 Recommendation Model Generation

A collaborative filtering based recommendation model is generated. Model structure is visualized in Figure 4.3. This model is prepared for evaluating user-item pairs by generating a prediction score.

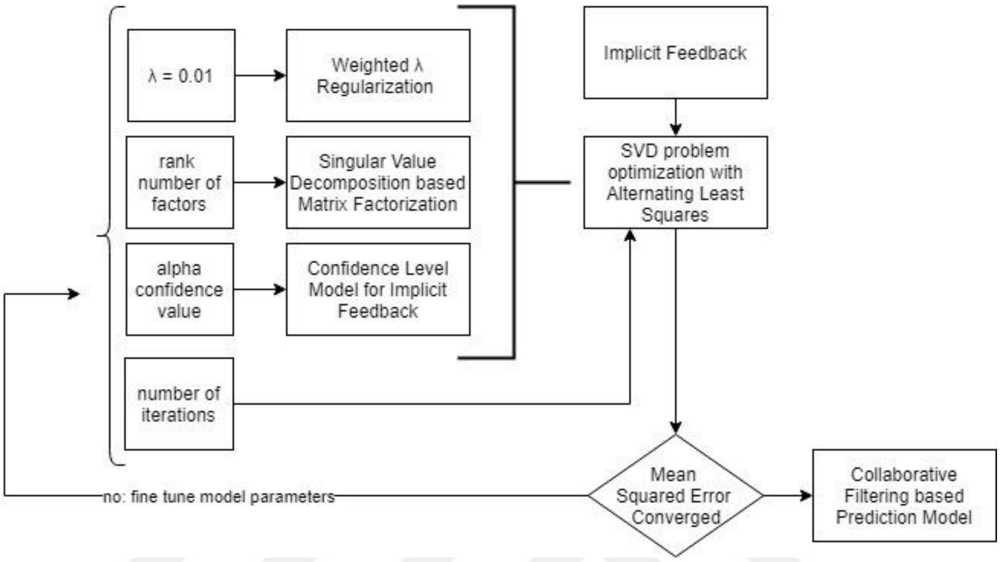

Figure 4.3: Recommendation Model Generation

- Weighted λ regularization is implemented to the model. Due to the resource limitations, λ is set as 0.01 and not fine-tuned. As it is unfolded above, regularization is required for preventing overfitting.
- Singular Value Decomposition is used for solving the matrix factorization problem. Matrix factorization is utilized for modelling users and items with latent factors. Since ratings are considered as the strength of confidence for a user-item relationship, the user-item association is uncovered. Rank parameter defines the number of latent factors.
- Confidence level model is adopted to implicit feedback data. It is a model converts the implicit feedback rating into a rebalanced value. Alpha is used as multiplier for rating data.
- Since singular value decomposition problem has two unknowns which are user latent factors vector and item latent factors vector, Alternating Least Squares is selected for optimization. As it is detailed above, it solves the SVD problem by fixing the unknowns one by one and rotating the mean squared error minimization

problem. Number of iterations or epoch number represents the number of rotating for ALS.

- Mean Squared Error is used as test error. When MSE converges, the problem is optimized with given parameters.

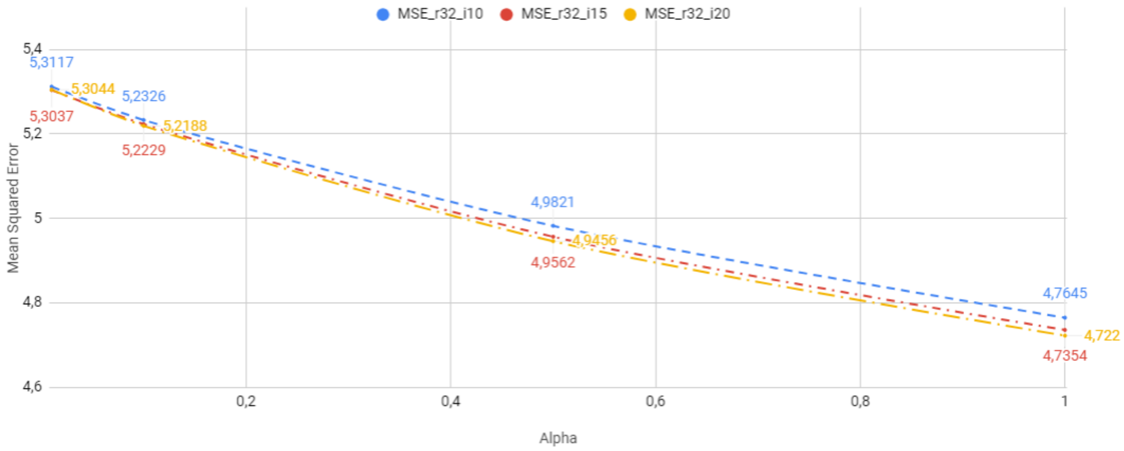

Figure 4.4: MSE versus Number of Iterations and Alpha

Figure 4.4 represents the how number of iterations and alpha parameters interact mean squared error. Given the rank is fixed as 32 and number of iteration is set as 10, 15 and 20, MSE is inversely proportional with alpha. Spark sets alpha as 1.0 by default. Therefore alpha is iterated between 0.01 and 1.0.

Comparing the convergence under fixed rank for different number of iterations, increased alpha decreases the error rate, however the convergence rate arguably changes with increased number of iterations. Therefore alpha is fine-tuned with rank instead of number of iterations.

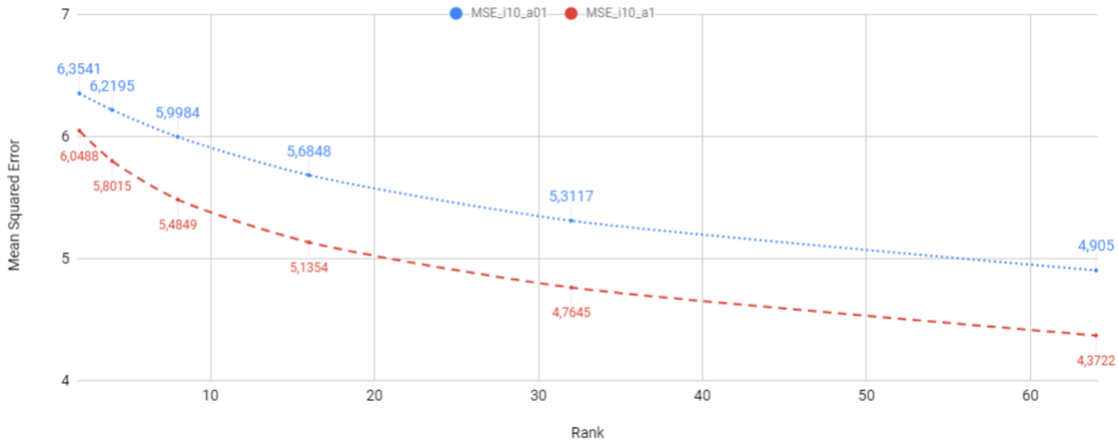

Figure 4.5: MSE versus Rank and Alpha

Figure 4.5 represents the how rank and alpha parameters leverage mean squared error. Given the number of iterations is fixed as 10 and alpha is set as 0.01 and 1, MSE is inversely proportional with rank. MSE converges at a lower limit with the increased alpha.

Comparing the convergence under fixed number of iterations for different alpha, increased rank decreases the error rate. In addition to that, convergence rate is higher with increased alpha. Therefore, alpha is selected as 1.

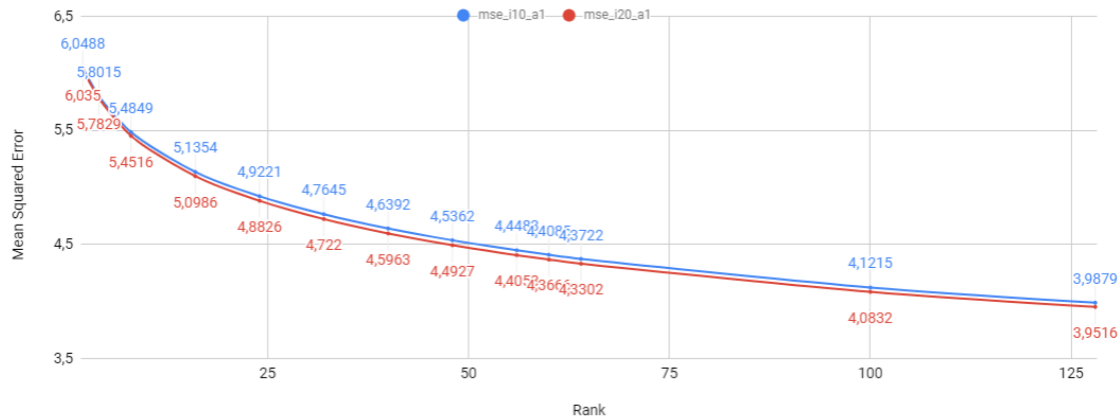

Figure 4.6: MSE versus Rank and Number of Iterations

Figure 4.5 represents the how rank and number of iterations parameters leverage mean squared error. Given the alpha is fixed as 1 and number of iterations is set as 10 and 20, MSE is inversely proportional with number of iterations.

Comparing the convergence under fixed alpha for different number of iterations, increased number of iterations decreases the error rate, however the convergence rate arguably changes with increased number of iterations. Regarding the memory limitations of our sources, epoch number is selected as 10. Mean squared is retrieved for different rank values. The model is trained with number of iterations as 10 and alpha as 1.

Slope, shown as Eq.26, represents how mean squared error decreased per increase in rank. Since mean squared error has nonlinear decrease pattern within the increased rank, slope is calculated individually.

$$Slope = \frac{MSE_{x-1} - MSE_x}{Rank_x - Rank_{x-1}}, where\ x: Rank\ Range \tag{26}$$

Table 4.2: MSE Decrease Slope

x	Rank Range	Slope
1	2-4	0.1231
2	4-8	0.0792
3	8-16	0.0437
4	16-24	0.0267
5	24-32	0.0197
6	32-40	0.0157
7	40-48	0.0129
8	48-56	0.0110
9	56-60	0.0099
10	60-64	0.0091
11	64-100	0.0070
12	100-128	0.0048

As shown in Table 4.2, closer slopes are filtered and ranks between 48 and 64 are considered as convergent. In order to demonstrate the acceleration and find the convergence, slope decrease rate is calculated.

Slope decrease rate, as shown in Eq.27, represents proportional improvement of slope by comparing the current range with previous range. A rank range with high slope means the error is still decreasing. Slope is expected to be decreased with the convergence or mean squared error. Since slope decrease rate compares the previous and current slopes, it is a relative value.

$$Slope\ Decrease\ Rate = \frac{(Slope_{x-1} - Slope_x)}{Slope_x}, where\ x: Range\ Shift \tag{26}$$

Table 4.3: Slope Decrease Rate

x	Range Shift	Slope Decrease Rate
1	40-48 > 48-56	0.1718
2	48-56 > 56-60	0.1043
3	56-60>60-64	0.0964

As shown in Table 4.3, from 40 to 56, MSE decreased more than 48 to 60. Also MSE decreased more at 48- 60 range comparing to 56-60 range. Comparing the slope decrease rate at 56-60 range (0.1043) and 60-64 range (0.0964), the successor range is relatively close to previous range. 48-56 is decreasing relatively faster than 56-60 and eventually 60-64. Since 56-60 is a rank range with relatively steady MSE decrease, number of iterations is selected as 60.

4.5 User-Item Pair Selection for Ranking Prediction

The recommendation engine is trained with implicit feedback collected in 4 weeks from specific customers for specific group of products. Predictions are made for the users who showed intention or purchased a kitchen appliance product on January 31st 2019.

In short predictions are generated within one day of interaction by utilizing 4 weeks of feedback data. The last day of training data is preferred, therefore future actions can be interpreted for further analysis. Prediction pair generation flow is visualized in Figure 4.7.

Ground rules for prediction data generation:

1. Predictions are made for the users who showed intention or purchased a kitchen appliance product in a specific day.
2. In order to increase the relevance, products and main categories with poor impression are eliminated.
    - Item based total number of users with feedback is calculated.
    - Main category based average number of users with feedback is calculated with using item based total values.
    - Total number of feedback for an item is larger than the main category average.
    - Main category average is larger than 169 for SDA categories and is larger than 9 for FMCG categories.
3. Users are characterized as inclined users and purchased users. Inclined users are those who showed an intention however didn't purchase a kitchen appliance. Purchased users are those who ordered one or more SDA product.
4. For purchased users, all products other than the ones belonging to the main category they already shopped from are predicted.
5. For inclined users, all FMCG products and the SDA products belonging to the categories they showed intention are predicted.

A total of 324617 user-item pairs are predicted for 1209 users. The number of items is 565, 522 of them are kitchen appliance products, the remaining amount of 43 are beverages. 339 users purchased an SDA product from our prediction product set. The remaining 870 users showed intention to kitchen appliance products. Average amount of predicted products for inclined customers is 185, while the same value is 517 for purchased customers.

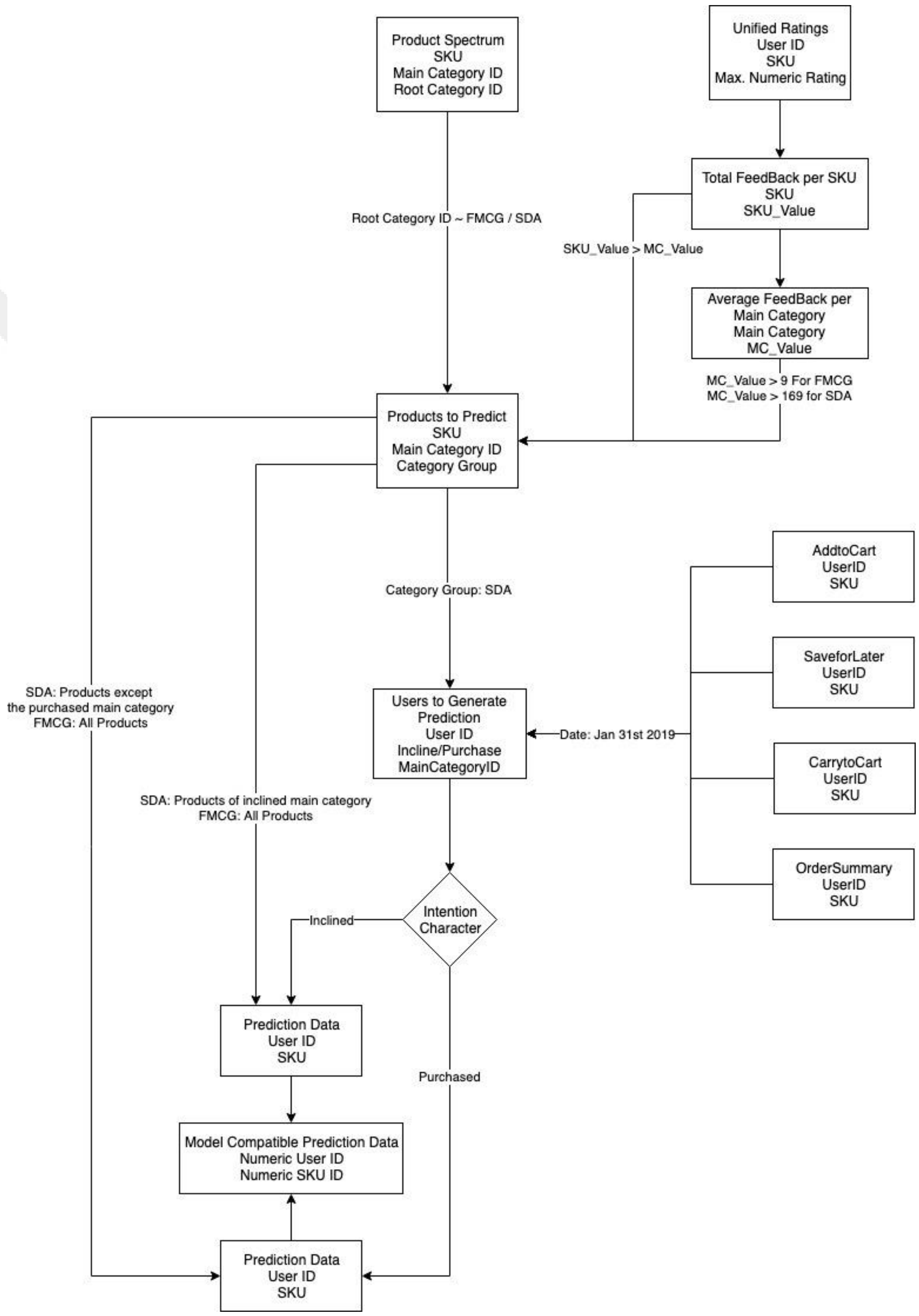

Figure 4.7: Prediction Pair (User, Item) Generation

4.6 Predictions and Tailored Recommendations

The collaborative filtering model generates numeric predictions for each user-item pair. This approach treats the numeric implicit feedback data as a representation of strength in observations of user behaviour. The model is trying to find the latent factors that can be used for prediction of expected preference of a user for a given item. Likewise, the prediction values represent the strength of intention of a user for an item. As prediction pairs are already introduced, the scale of items for each user is in hundreds. It is quite noticeable that even 50 item recommendation set is excessive for a multichannel online platform. Therefore recommendations are narrowed as 10 items for each user.

As mentioned above, the users in predictions and model training are limited as the ones who showed an intention to kitchen appliances. We already know from the training data, most of these users didn’t show any intention to the FMCG products. Regarding the business objectives such as showing the assortment of products and converting more of the cross-sale opportunities, these 10 items are divided into two subgroups. One group is the products with highest predicted ranking. The other group is tailored in accordance with business objectives. Several tailored recommendation examples are presented in Table 4.4.

Ground Rules for Tailored recommendations:

1. Number of recommendations is limited by 10 items for each user.
2. Total number of 6 items are the ones with highest predictions.
3. 4 items are selected from the other predicted items for the user.
4. There must be at least 4 at most 6 FMCG products in a recommendation set.
5. The deficit number of item for SDA and FMCG product for each user is calculated.
6. The deficits are covered by finding the highest ranked products from the corresponding category.
7. First 6 items and tailored 4 items are combined resorted by prediction value for each user.

Table 4.4: Tailored Recommendation Examples

Intention Character: Inclined Elektrikli Çay Makineleri	Intention Character: Purchased Elektrikli Çay Makineleri	Intention Character: Purchased Blender
Korkmaz A353-03 Mia Çay Kahve Makinesi Lila	Arzum Okka OK005-B Grandio Türk Kahvesi Makinesi Eu 2 Pin Beyaz	Altus AL 797 M Mor Ehli Kahve Türk Kahve Makinesi
Lipton Signature Series Royal Ceylon Siyah Dökme Çay 130 gram	Kurukahveci Mehmet Efendi Türk Kahvesi Teneke 250 gr	Stilevs Sadem Türk Kahve Makinesi Pembe&Gri
King P-315 MP Lea Çay Makinesi Turkuaz	Beko Bkk 2185 Tg Tost Makinesi	Korkmaz A309-02 Vertex 800 W Waffle ve Tost Makinesi
Korkmaz A 369-03 Demtez Elektrikli Çaydanlık Lila	Delonghi Kg89 İnox Kahve Öğütücü	Arzum Okka OK005-B Grandio Türk Kahvesi Makinesi Eu 2 Pin Beyaz
Karali Organik Siyah Çay 500 Gr	Fakir Grace 2000W Izgara ve Tost Makinesi – Siyah	Schafer Kaffefan Elektrikli Cezve Turkuaz
Karali Karadeniz Filiz Siyah Çay 1 Kg	Premier Pwm 215 Waffle Yapma Makinesi	Korkmaz A 365-22 Smart Elektrikli Cezve Mavi/Gri
Korkmaz A 369-04 Demtez Elektrikli Çaydanlık Turkuaz	Arzum AR131 Ironmix 550W Çubuk Blender Siyah	Çaykur Tomurcuk 125 gram
Julius Meinl Auslese Filtre Kahve 1Kg	Lipton Demlik Poşet Çay Earl grey 100'Lü	Nescafe Classic 200 gr Ekopaket
Bluehouse Bh268 Çay Makinesi	Lipton Yellow Label 750 li Demlik Poşet Çay + Termos	Çaykur Organik Siyah Hemşin Çayı 400 gr (Karton Kutu)
Stilevs Çays Cm 16 Çay Makinesi	Kurukahveci Mehmet Efendi Colombian Filtre Kahve 500 Gram	Jacobs Monarch Filtre Kahve 500 Gr Alana 2.Si %50 İndirimli

4.7 Evaluation and Results

Recommendations are generally exhibited to the customers on online platforms to increase the engagement. To predict the next best action of display is a whole art of recommendation strategy execution. Every location of an online platform has different strength of preference for each customer. It is related to the customer intention, the nature of product and the other environmental factors leading customers to come and visit. Therefore, the reasoning behind the personalized recommendation should fit the location. In conclusion, a customer bouncing from one product page to other to find a product requires a help on sorting the options, while a customer on checkout page should not be distracted with other products. In this study, we prepared the recommendations however they are not displayed to the customers. In order to evaluate our predictions, the product view and add to cart events are used.

A subtle recommender system should be empowered by customer behaviour, up to date, relevant yet unforeseen and personalized. These goals are addresses with following steps.

- In this study, total of 174626 events are retrieved from 39926 customers for 1403 different products. These events are converted into numeric feedback. An implicit feedback data is a (Customer, Product, Rating) tuple. Predictions are generated for 1209 customers and 565 products. A prediction data is a (Customer, Product) pair.
- Customer behaviour is converted into implicit feedback. Implicit feedback represents the strength of confidence between customers and products. With respect to the riddle of customer taste, ratings are decomposed with latent factors.
- This model is empowered with machine learning for the sake of recommendations freshness and serendipity. This learning machine is optimized for predicting prospective customer-product association strength. It is regularized for preventing overfitting. Therefore, predictions are considered as adaptive and confident.
- Prediction pairs are prepared. Customers are characterized as purchasers and incliners. Also recommendations are filtered by 10. Two categories are included in product spectrum. Therefore, four of 10 products are selected in accordance

with business objectives which is to represent the assortment and to respect the customers incline/purchase preferences.

Recommendations are prepared for the customers who at least added a kitchen appliance product to the basket on 31st of January 2019. Product view and add to cart events generated in February 2019 are collected for these customers. 1209 prospective customers are tracked, 234 customers viewed the at least one of ten recommended products within February 2019. 79 customers added at least one of ten recommended products to basket within February 2019. Comprehensively, %19 of prospective customers ended up discovering what we predict for them, %7 of them showed a purchase intention to what we predict for them.

975 prospective customers are not interacted to any of recommended products. 73 of them viewed a product we predict however didn't recommend. 29 of them is added to basket a product we predict however didn't recommend. Comparing to the existing e-commerce metrics, these ratios are promising for a visual implementation. Confusion matrix values are displayed in Table 4.5.

Confusion matrix is redefined for evaluation. Precision is the ratio of true positives over predicted positives. True positive is the total number of customers who viewed or added to basket what we recommended. In this study, it is 234. False positive is total number of customers who didn't view or add to basket what we recommended. It is 975. False negative is the total number of customers who viewed or added to basket what we predict but didn't recommend. These customers didn't interact any of recommended products. In this study, it is 73. Predicted positive is total customers we made recommendations. It is the total of true positive and false positive. It is 1209. Actual positive is total customers who visited or added to cart at least one predicted product. It is the total of true positive and false negative. It is 307.

Table 4.5: Confusion Matrix Values

Prediction Status	Definition	Number of Customers
True Positive	Interacted to Recommendation	234
False Positive	Not Interacted to Recommendation	975
False Negative	Interacted to Trimmed	73
True Negative	Not Interacted to Trimmed	1136

Tailored recommendations contain only 10 products while average amount of predicted products for inclined customers is 185 and the same value is 517 for purchased customers. Precision fraction answers that: how likely a customer who added a kitchen appliance product in a given day would be interested in these ten products? It represents the engagement the tailored recommendations. In this study, precision is calculated as 0.19.

Recall fraction answers that: how many of predicted interactions are covered by recommendations. It represents the coverage of tailored recommendations over all product interactions. In this study it is 0.76.

By definition, precision and recall depends on the number of predictions and number of recommendations. Therefore, precision and recall should be interpreted relatively to these values. Success rate is 0.57 and error rate is 0.43.

Regarding we are not able to present these recommendations to the customers. Overall, ten product recommendation set is interacted by 19% of prospective customers and ten product recommendations are enough to cover %76 of customers who interacts a predicted product in next period.

5 CONCLUSION AND PERSPECTIVES

This study aims to propose a comprehensive recommendation system framework for Turkey’s leading e-commerce platform. In this context, a directed research is accomplished. This research is consolidated as a literature review referring to the motivation, types of recommender systems and recommendation model accuracy evaluation. It is followed by collaborative filtering referring to types of collaborative filtering, dimensionality reduction, matrix factorization and cold start problem. Machine learning concepts are also broadly explained.

Next in order, the methodology behind the proposed recommendation system is analysed. Applying alternating least squares for solving a regularized singular value decomposition problem is explained.

Following to that, the case study is acknowledged. Implicit feedback model, product spectrum, customer targeting and sales funnel are introduced. Model training step is elaborated. Then prediction data preparation tasks are visually and verbally explained. The motivation behind tailored and limited predictions and how it is accomplished is demonstrated. Recommendations are evaluated and results are shared.

Considering the accessibility of behavioural data and sophistication of customer taste, latent factors based collaborative filtering is proposed. Implicit product feedback from customers are retrieved from data. Customer and products are represented with latent factors.

A prediction model is generated by solving a dynamically regularized singular value decomposition problem with alternating least squares. Model training parameters are fine-tuned and predefined predictions are delivered.

This framework can be enhanced with further implementations. One of them is to explain the strategy behind the recommendations. The examples are 'You are seeing this because people like you purchased this product' or 'You are seeing this because you purchased that product'. To inform the customer about the reason behind the recommendations is more trustworthy and the customer can know the coherence.

The other improvement opportunity is to enrich the implicit feedback model with after sales data such as review context, return status, replenishment status. Considering the visit numbers and high assortments, millions of events are generated by customers visiting different domains. Our case study is limited with 3 categories. However solving the problem with ALS and weighted λ regularization is suitable for big data handling. This framework can be extended for larger datasets.

In this study, recommendations are generated for the customers who at least added an item to basket within given day. Thus, cold start problem is excluded, the model can be trained larger data sets and cold starters can be tested.

REFERENCES

Abu-Moustafa, Y. S., Magdon-Ismail, M., & Lin, H. T. (2012). *Learning from data*. New York, NY, USA: AMLBook.

Adomavicius, G., & Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge & Data Engineering*, *6*, 734-749.

Aïmeur, E., Brassard, G., Fernandez, J. M., & Onana, F. S. M. (2008). A lambic: a privacy-preserving recommender system for electronic commerce. *International Journal of Information Security*, *7*(5), 307-334.

Barragáns-Martínez, A. B., Costa-Montenegro, E., Burguillo, J. C., Rey-López, M., Mikic-Fonte, F. A., & Peleteiro, A. (2010). A hybrid content-based and item-based collaborative filtering approach to recommend TV programs enhanced with singular value decomposition. *Information Sciences*, *180*(22), 4290-4311.

Batmaz, Z., & Polat, H. (2016). Randomization-based Privacy-preserving Frameworks for Collaborative Filtering. *Procedia Computer Science*, *96*, 33-42.

Billsus, D., & Pazzani, M. J. (1998). Learning Collaborative Information Filters. *ICML*, *98*, 46-54.

Bishop, C. M. (2006). *Pattern recognition and machine learning*. New York, NY, USA: Springer.

Cantador, I., Bellogín, A., & Vallet, D. (2010) . Content-based recommendation in social tagging systems. In *Proceedings of the fourth ACM conference on Recommender systems*, 237-240.

Cherkassky, V., & Mulier, F. M. (2007). *Learning from data: concepts, theory, and methods*. John Wiley & Sons.

De Campos, L. M., Fernández-Luna, J. M., Huete, J. F., & Rueda-Morales, M. A. (2010). Combining content-based and collaborative recommendations: A hybrid approach based on Bayesian networks. *International Journal of Approximate Reasoning, 51*(7), 785-799.

Ekstrand, M. D., Riedl, J. T., & Konstan, J. A. (2011). Collaborative filtering recommender systems. *Foundations and Trends in Human–Computer Interaction, 4*(2), 81-173.

Fathi, F., Abghour, N., & Ouzzif, M. (2017, September). From Big data platforms to smarter solution, with intelligent learning:[PAV] 4-Pave the way for Intelligence. In *Proceedings of the 2017 International Conference on Cloud and Big Data Computing*, 11-16.

Fortmann-Roe, S. (2012). Understanding the bias-variance tradeoff. Retrieved from: http://scott. fortmann-roe. com/docs/BiasVariance

Goldberg, D., Nichols, D., Oki, B. M., & Terry, D. (1992). Using Collaborative Filtering to Weave an Information Tapestry. Commun. *ACM*, *35*(12), 61-70.

Goldberg, K., Roeder, T., Gupta, D., & Perkins, C. (2001). Eigentaste: A constant time collaborative filtering algorithm. *Information Retrieval*, *4*(2), 133-151.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press.

Harrington, P. (2012). *Machine learning in action*. Manning Publications Co..

Herlocker, J. L., Konstan, J. A., Terveen, L. G., & Riedl, J. T. (2004). Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems (TOIS)*, *22*(1), 5-53.

Hill, W., Stead, L., Rosenstein, M., & Furnas, G. (1995). Recommending and evaluating choices in a virtual community of use. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 194-201.

Hu, Y., Koren, Y., & Volinsky, C. (2008, December). Collaborative filtering for implicit feedback datasets. *Eighth IEEE International Conference on Data Mining*, 263-272.

Hurwitz, J., & Kirsch, D. (2018). *Machine learning for dummies*. John Wiley & Sons.

Qiu, J., Wu, Q., Ding, G., Xu, Y., & Feng, S. (2016). A survey of machine learning for big data processing. *EURASIP Journal on Advances in Signal Processing*, *2016*(1), 67.

Kasap, Ö. Y., & Tunga, M. A. (2017). A polynomial modeling based algorithm in top-N recommendation. *Expert Systems with Applications*, 79, 313-321.

Kasap, O. Y., Ekmekci, N., & Ketenci, U. G. (2016). Combining logistic regression analysis and association rule mining via MLR algorithm. *ICSEA 2016,* 167.

Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer*, 8, 30-37.

Kotsiantis, S. B., Zaharakis, I., & Pintelas, P. (2007). Supervised machine learning: A review of classification techniques. *Emerging Artificial Intelligence Applications in Computer Engineering*, 160, 3-24.

Loh, W. Y. (2008). *Classification and regression tree methods*. Wiley StatsRef: Statistics Reference Online.

Lops, P., De Gemmis, M., & Semeraro, G. (2011). Content-based recommender systems: State of the art and trends. *Recommender Systems Handbook*. Boston, MA: Springer.

Lops, P., Jannach, D., Musto, C., Bogers, T., & Koolen, M. (2019). Trends in content-based recommendation. *User Modeling and User-Adapted Interaction*, 1-11.

Mohri, M., Rostamizadeh, A., & Talwalkar, A. (2018). *Foundations of Machine Learning*. MIT press.

Nadimi-Shahraki, M. H., & Bahadorpour, M. (2014). Cold-start problem in collaborative recommender systems: efficient methods based on ask-to-rate technique. *Journal of Computing and Information Technology*, *22*(2), 105-113.

Naghibi, S. A., Pourghasemi, H. R., & Dixon, B. (2016). GIS-based groundwater potential mapping using boosted regression tree, classification and regression tree, and random forest machine learning models in Iran. *Environmental Monitoring and Assessment*, *188*(1), 44.

Patra, B. K., Launonen, R., Ollikainen, V., & Nandi, S. (2015). A new similarity measure using Bhattacharyya coefficient for collaborative filtering in sparse data. *Knowledge-Based Systems*, 82, 163-177.

Polatidis, N., & Georgiadis, C. K. (2017). A dynamic multi-level collaborative filtering method for improved recommendations. *Computer Standards & Interfaces*, 51, 14-21.

Ricci, F., Shapira, B., Rokach, L. & Kantor, P. (2011). *Recommender Systems Handbook*. Springer.

Roy, A., Banerjee, S., Sarkar, M., Darwish, A., Elhoseny, M., & Hassanien, A. E. (2018). Exploring New Vista of intelligent collaborative filtering: A restaurant recommendation paradigm. *Journal of computational science*, 27, 168-182.

Settles, B. (2009). *Active learning literature survey*. University of Wisconsin-Madison Department of Computer Sciences.

Shalev-Shwartz, S., & Ben-David, S. (2014). *Understanding machine learning: From theory to algorithms*. Cambridge University Press.

Smola, A., & Vishwanathan, S. V. N. (2008). *Introduction to machine learning.* Cambridge University, 32, 34

Takacs, G., & Tikk, D. (2012). Alternating least squares for personalized ranking. In *Proceedings of the sixth ACM conference on Recommender systems*, 83-90.

Wang, Y., Deng, J., Gao, J., & Zhang, P. (2017). A hybrid user similarity model for collaborative filtering. *Information Sciences*, 418, 102-118.

Witten, I. H., Frank, E., Hall, M. A., & Pal, C. J. (2016). *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann.

Xiong, R., Wang, J., Zhang, N., & Ma, Y. (2018). Deep hybrid collaborative filtering for Web service recommendation. *Expert Systems with Applications*, 110, 191-205.

Zhang, C., & Ma, Y. (Eds.). (2012). *Ensemble machine learning: methods and applications*. Springer Science & Business Media.

Zhou, Y., Wilkinson, D., Schreiber, R., & Pan, R. (2008, June). Large-scale parallel collaborative filtering for the netflix prize. In *International conference on algorithmic applications in management*, 337-348.

BIOGRAPHICAL SKETCH

Merve ARTUKARSLAN was born in İstanbul on January 6, 1990. She obtained her B.S. degree in Industrial Engineering in İstanbul Technical University. Since September 2015, she is a student at Galatasaray University in M.S of Industrial Engineering. Also she worked in Ziraat Teknoloji Customer and CRM team between 2013 and 2017. Since February 2017, she is working at Hepsiburada Search and Navigation team as product lead. Her areas of interest include search relevance optimization and ranking cocktail, machine learning in application, big data, quantitative data analytics and KPI measurements, findability and discoverability in e-commerce.