proceedings

MDPI

Proceedings

Application of Recommender System for Spending Habits Based Campaign Management

Tuğçe Süheyla Kaya [1,2,*], Murat Gezer [1] and Sevinç Gülseçen [1]

[1] Informatics Department, Istanbul University, Istanbul 34452, Turkey; murat.gezer@istanbul.edu.tr (M.G.); gulsecen@istanbul.edu.tr (S.G.)
[2] Softtech, ITU Ayazaga Campus Teknokent Arı-3, Istanbul 34467, Turkey
* Correspondence: tugcesuheyla.cetin@ogr.iu.edu.tr
† Presented at the 7th International Management Information Systems Conference, 9–11 of December 2020. Available online: http://2020.imisc.net.

Abstract: Nowadays, banks are working on finding a suitable campaign for every customer profile. With this study, we aimed to develop a recommendation system that will direct the customer to the appropriate campaign. With the data received from a private bank, credit card transactions of the users were analyzed, and spending habits were modeled. We aimed to recommend the most suitable campaign to the users through the created models. Within the scope of the study, 662.088 credit card transactions performed by 4997 customers within three months were analyzed, and three campaigns were proposed for each customer as a result of the study. The ALS (Alternating Least Square) algorithm was used on Spark to establish the recommendation system. The primary purpose of the study is to increase customer satisfaction by finding unique users based on spending habits instead of campaigns that are applied collectively to customers by making a personalized campaign offer.

Keywords: recommendation systems; banking; campaign management

Citation: Kaya, T.S.; Gezer, M.; Gülseçen, S. Application of Recommender System for Spending Habits Based Campaign Management. *Proceedings* **2021**, *74*, 7. https://doi.org/10.3390/proceedings2021074007

Published: 4 March 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).

1. Introduction

The recommender systems are a class of information retrieval domain. The main purpose of the recommendation system is to improve the consumer experience and to provide user-related items. With the growing usage of credit cards, banks are storing an enormous amount of data about customers' profiles, such as spending habits, location, or demographic information. This big data can be used for campaign management using recommender systems. Campaign management is providing a suitable campaign for a suitable customer at the right moment. For the banking sector, campaign management is part of customer relationship management (CRM). CRM is a strategy that allows companies to analyze customer profiles, determine their needs and areas of profitability, and take the necessary actions to achieve both customer satisfaction and profitability [1]. CRM covers many management units, such as campaign management, human resources management, sales management, and service management. Today, there is deep competition among banks, which is an advantage for customers. Customers expect fewer transaction fees, higher interest rates, new products, and appropriate campaigns from banks [2]. Therefore, campaign management is essential to ensure customer satisfaction; thus, suitable campaign recommender systems can be used. One example of recent studies on the subject is Reference [3]. In this study, recommender systems were used for CRM to increase customer satisfaction. Recommendation systems are algorithms that provide the most meaningful and accurate products for the user by filtering useful content from big data. The data for recommendation systems can be collected not only by delivering the opinions of the users directly like ratings but also indirectly, such as purchase history, time spent on web pages, email content, etc. [4]. Generally, recommendation systems are classified as

collaborative filtering, content-based filtering, and hybrid systems, as shown in Figure 1. Content-based filtering systems produce recommendations based on item specifications/features, which could be used web pages or news recommendations. In collaborative filtering approaches, we accept that similar users like similar items. These systems have affected the way the online world of e-commerce and social media function, with some popular examples being Netflix movie recommendations, Amazon product recommendations, or friend recommendations on Facebook [5].

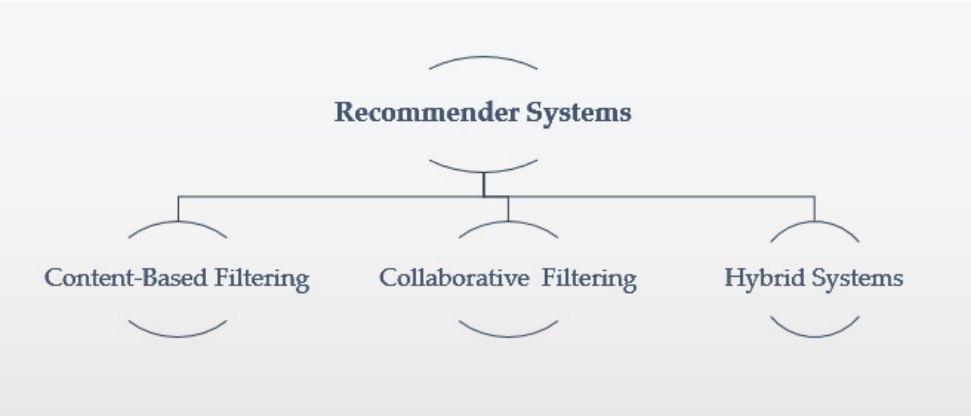


Figure 1. Structure of the recommendation systems.

Lesage et al. (2020) built a recommendation engine for car insurance to increase selling performance. This recommendation system combines two different algorithms to find appropriate cover for the appropriate customer. Stratigi et al. (2019) [6] studied Amazon movies data to build a recommendation system and compare results for a content-based approach with review counts, collaborative approaches for rating values, and hybrid recommendations for combining them. Another example of hybrid systems by Srikanth and Nagalakshmi (2020) [7] built a song recommendation system using the SVD (Singular Value Decomposition) machine learning algorithm. Kulkarni (2017) [8] developed a book recommendation system using Apache Spark. In that study, the solution was proposed to one of the hardest problems of the recommendation system; the cold start problem being a lack of evaluation value for new items or new users, by recommending popular books in the absence of evaluation value. Another approach of solving the cold start problem by Aggarval and Bahuguna (2017) [9], built a recommendation engine for a MovieLens data set where suggestions for new users are produced from the demographic characteristics of the users. Dutta and Bandyopadhyay (2020) [10] used recommender systems to investigate customer behavior on term deposit subscriptions using featured data, which includes customer's age, job profile, marital status, etc. The proposed recommender system has an accuracy of 88.32%. Another example of a recommendation system for banking applications is integrating a recommendation system to the process of delivery of personalized customer services. Nieves et al. (2019) [11] developed a hybrid recommendation system for the banking products such as mortgages, loans for improving aspects of customer support services, and reducing entity management costs. In this study, a spending habit-based recommender system is proposed for campaign management. For this purpose, 4997 customer's spending habits are analyzed and modeled from 662.088 credit card transaction data obtained from a private bank. The developed engine recommends to customers the three most suitable campaigns among sixteen proposed campaigns. The ALS (Alternating Least Square) algorithm was used on Spark to establish the recommendation system. By recommending a campaign according to customer's spending habits, we aimed to increase the satisfaction of the customer.

2. Method

The purpose of this study was to build a recommendation system based on spending habits using collaborative filtering algorithms. These algorithms aim to fill in the missing values of a user-item association matrix. In this study, the ALS (Alternating Least Square) method was used on Apache Spark to establish the recommendation system for a Matrix Factorization Model (MF). R, a rating matrix of size U X M can be decomposed into two low rank matrices, P and Q, of size U X K and M X K, respectively, where K is called the rank of the matrix [12]. The purpose of matrix factorization model, filling empty cells in the original matrix R using low rank matrices P and Q, is given by the following equation:

$$\hat{r}_{ij} = p_i^T = \sum_{k=1}^{k} p_{ik} q_{kj} \quad (1)$$

To make strong recommendations, predicted values are as close as the original values. The error between the original and predicted value given as:

$$\min_{q^*,p^*} \sum_{(u,i)\in K} (r_{ui} - q_i^T p_u)^2 + \lambda(\|q_i\|^2 + \|p_{ui}\|^2) \quad (2)$$

In order to optimize the preceding equation, the Stochastic Gradient Descent (SGD) and Alternating Least Squares algorithms are commonly used. In this study, the ALS algorithm was used. The ALS is an iterative algorithm that involves computing one feature vector term using the least-squares function by fixing the other feature vector term constant until solving the equation optimally [13].

In collaborative filtering recommendation systems with implicit feedback that only have positive feedback, if a user has no feedback for an item in the dataset, it does not mean the user dislikes it [14]. Moreover, for implicit feedback-based recommendation systems, user reactions could not be tracked so precision-based metrics are not very appropriate. In this study, a recall-based evaluation metric [15] known as Mean Percentage Ranking (MPR) was used:

$$\overline{rank} = \frac{\sum_{u,i} r_{ui}^t rank_{ui}}{\sum_{u,i} r_{ui}^t} \quad (3)$$

2.1. Dataset and Processing

This work used credit card transaction data obtained from a private bank. The dataset had 4997 customer's with 662.088 credit card transactions data that included encrypted the customer number, merchant category code (MCC), age, marital status, education level, transaction amount, transaction date. MCC is a four-digit number that is assigned by a bank or card organization such as Visa, Master Card, etc. to determine credit card transaction's market segment [16].

First of all, all MCC codes were merged into sixteen merchant category groups (MCG) according to their fields, and these are also campaign groups to be used in the study. Then the transaction data sets were grouped by user ID and MCG to find users' transaction counts for each MCG. After data processing, the final version of the data set, used in this study, had 79,952 rows including user ID, MCG, and transaction count for each MCG.

A wide range of 16 MCGs from education to insurance were used in the study from the data obtained from the private bank, as shown in Table 1.

Table 1. Definition of MCG.

MCG Code	Definition
MCG1	Kids
MCG2	Other Payments (Dealers)
MCG3	Education
MCG4	Home
MCG5	Bill Payments
MCG6	Clothing and accessory
MCG7	Hobby and entertainment
MCG8	Supermarket
MCG9	Car and transportation
MCG10	Health and personal care
MCG11	Unclassified expenses
MCG12	Insurance
MCG13	Vacation and travel
MCG14	Tax and legal fees
MCG15	Restaurant payments
MCG16	Investment and savings

2.2. Research Model

The model developed for the campaign recommendation system is given in Figure 2.

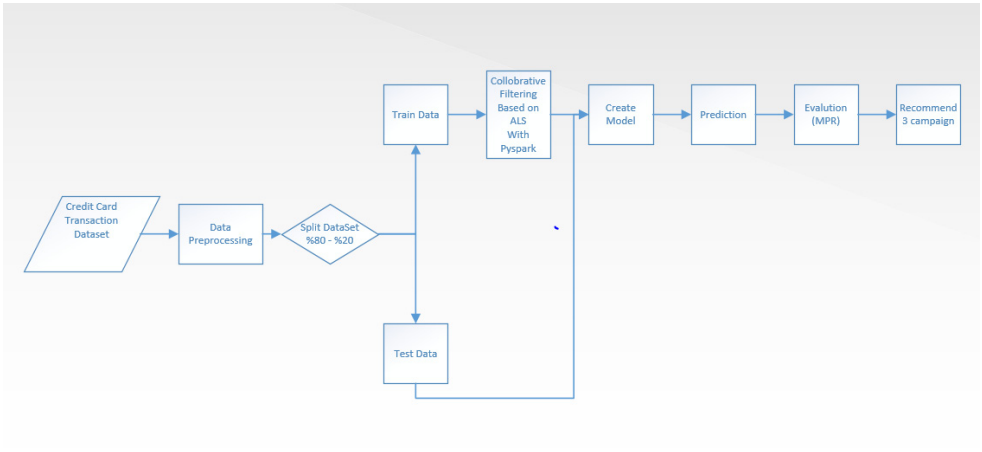


Figure 2. Architecture of the proposed model.

As shown in Figure 1, the first step of the study is data preprocessing. All transaction and customer data imported as Microsoft SQL Server tables included customers and transactions. Then MCC codes were grouped into MCG codes, and the credit card transaction data were organized according to MCG codes. In its final form, the data set consisted of user ID, MCG, and count of transactions.

In this study, we built a recommendation system with implicit feedback using Apache Spark 3.0. Apache Spark is an open-source project for big data and machine learning. For building a recommendation system in this study, the ALS algorithm was used with PySpark. Data were split 60% to train, 20% for validation, and 20% for testing, and the models were evaluated using MPR. The most successful model was selected, and three campaigns were recommended for each user.

3. Findings

This study was aimed at developing a recommendation system with implicit feedback using the ALS algorithm for Matrix factorization. Matrix factorization uses latent

factors that are the features in the lower dimension latent space projected from the user-item interaction matrix for representing user preferences in a much smaller dimension space. ALS is an optimization algorithm for minimizing the loss function. Hyperparameter tuning gives a tuple of hyperparameters that provides an optimal mode [13].

The Spark ALS model has an infrastructure for model tuning with some hyperparameters, such as regularization, rank, etc [17].

- **regParam** is regularization that reduces overfitting,
- **rank** is the number of latent factors in the model,
- **maxIter** is the maximum number of iterations to run,
- **alpha**, is a parameter for implicit feedback that governs the baseline confidence in preference observations values.

The parameter values used in this study; regParam = {0.05, 0.01, 0.02}, Alpha = {10, 20}, rank = {8, 10, 12, 16}, and maxIter = {10, 20}.

For the first model of a recommendation system for ten iterations, the best model had 16 latent factors, 0.01 regularizations, and an MPR value of 0.263.

The second model was created using the same parameters, but with 20 iterations. The results of the two models are given in Table 2.

Table 2. Results of Model.

Number of Latent Factors	Regularization	Alpha	MPR—10 Iterations	MPR—20 Iterations
8	0.05	10	0.504	0.485
8	0.05	20	0.519	0.489
8	0.01	10	0.525	0.522
8	0.01	20	0.540	0.513
8	0.02	10	0.556	0.521
8	0.02	20	0.536	0.580
10	0.05	10	0.390	0.424
10	0.05	20	0.390	0.403
10	0.01	10	0.378	0.418
10	0.01	20	0.380	0.413
10	0.02	10	0.357	0.400
10	0.02	20	0.365	0.376
12	0.05	10	0.325	0.370
12	0.05	20	0.325	0.339
12	0.01	10	0.330	0.374
12	0.01	20	0.319	0.346
12	0.02	10	0.361	0.403
12	0.02	20	0.356	0.386
16	0.05	10	0.271	0.227
16	0.05	20	0.308	0.257
16	0.01	10	0.263	0.213
16	0.01	20	0.312	0.248
16	0.02	10	0.265	0.217
16	0.02	20	0.318	0.241

For the second model of recommendation systems with twenty iterations, the best model had 16 latent factors, 0.01 regularization, and an MPR value of 0.213. This was also the most successful result between the two models, such that recommendations are produced according to it.

When the recommendations created are examined, it was seen that the most recommended campaign was vacation and travel. Other recommended offers for the first recommendation are shown below in Figure 3.

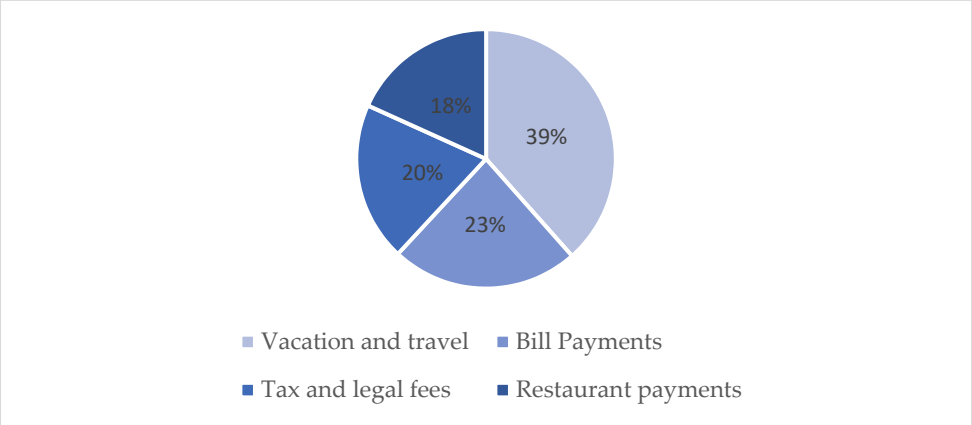


Figure 3. Firstly recommended campaigns.

Within the scope of the study, three campaigns were offered to all users, and the distribution of the suggested campaigns is given in Figure 4. MCG 8 (Supermarket) and MCG 15 (Restaurants payments) were the most recommended campaigns, and both were recommended to around 2000 customers, followed by MCG 13 (Vacation and Travel) and MCG 5 (Bill Payments) which were recommended to around 1800 and 1500 customers, respectively. MCG 1 (Kids), MCG 11 (Unclassified expenses), and MCG 12 (Insurance) were the least recommended campaigns and were recommended to below 200 customers.

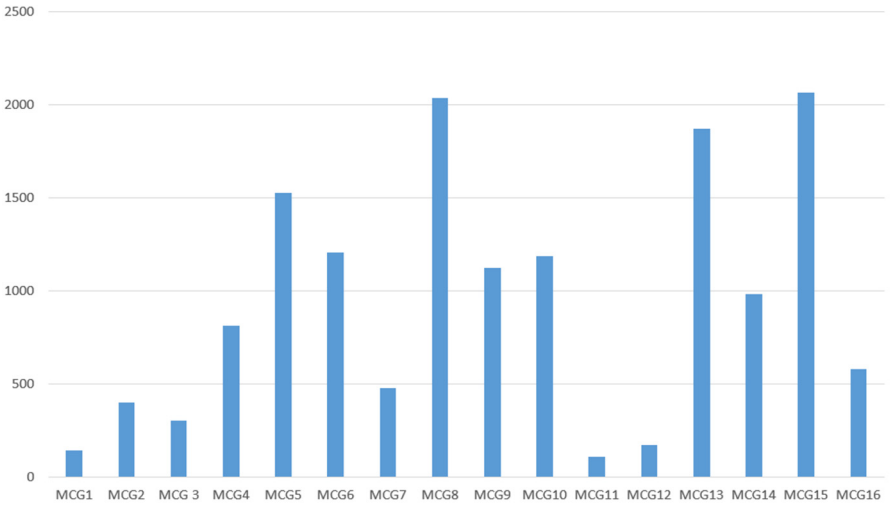


Figure 4. Distribution of recommended merchant category groups (MCGs).

4. Discussion and Conclusions

In this study, for building a recommendation system based on spending habits, the ALS algorithm was used. 662.088 credit card transactions performed by 4997 customers within three months were analyzed, and three campaigns were proposed for each customer as a result of the study. As a result of the evaluations, the best model has 16 latent factors and an MPR value of 0.213. The most recommended campaigns are supermarket and restaurant payments while the least recommended are kids, unclassified expenses, and insurance.

For future work, we aim to develop a hybrid recommender system that includes collaborative filtering, and content-based filtering by combining user evaluation values and user demographic features to increase the performance of the recommendation system.

Author Contributions: S.G. and M.G. supervised the study and T.S.K. analized data and interpreted results. S.G., M.G. and T.S.K. wrote paper and agreed to the published version of the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Al Karim, R.; Habiba, W. Effects of CRM Components on Firm's Competitive Advantage: A Case on Bangladesh Banking Industry. *Manag. Res.* **2020**, *10*, 1–7.
2. Pokharel, B. Customer Relationship Management: Related Theories, Challenges and Application in Banking Sector. *Bank. J.* **1970**, *1*, 19–28, doi:10.3126/bj.v1i1.5140.
3. Xu, C. Personal recommendation using a novel collaborative filtering algorithm in customer relationship management. *Discret. Dyn. Nat. Soc.* **2013**, *2013*, doi:10.1155/2013/739460.
4. Isinkaye, F.O. Recommendation systems : Principles, methods and evaluation. **2015**, 261–273, doi:10.1016/j.eij.2015.06.005.
5. Gupta, S.; Dave, M. Improvised Collaborative Filtering for Recommendation System. *Int. J. Innov. Technol. Explor. Eng.* **2020**, *9*, 361–364, doi:10.35940/ijitee.f3513.049620.
6. Stratigi, M.; Li, X.; Stefanidis, K.; Zhang, Z. *Ratings vs. Reviews in Recommender Systems: A Case Study on the Amazon Movies Dataset*; Springer: Cham, Switzerland, 2019; Volume 3; ISBN 9783030302788.
7. Srikanth, B.; Nagalakshmi, V. Songs Recommender System using Machine Learning Algorithm: SVD Algorithm. . *Int. J. Innov. Sci. & Res. Tech.* **2020**, *5*, 390–392.
8. Kulkarni, I.; Gandhi, P.; Karlekar, P. Book Recommendation System Using Apache Spark. *Int. J. Innov. Res. Comp. & Comm. Eng* **2017**, 7982–7987, doi:10.15680/IJIRCCE.2017.
9. Agarwal, G.; Bahuguna, H.; Agarwal, A. Solving Cold-Start Problem in Recommender System Using User. *Int. J. Emergig Tech.* **2017**, *8*, 55–61.
10. Shawni Dutta; Kumar, B.S. Recommender System for Term Deposit Likelihood Prediction Using Cross;validated Neural Network. Preprints **2020**, *19*, 1–37, doi:10.20944/preprints202004.0550.v1.
11. Hernández-Nieves, E.; Hernández, G.; Gil-González, A.B.; Rodríguez-González, S.; Corchado, J.M. Fog computing architecture for personalized recommendation of banking products. *Expert Syst. Appl.* **2020**, *140*, doi:10.1016/j.eswa.2019.112900.
12. Charu, C. Aggarwal *Recommender Systems*; 2017; ISBN 9783319296579.
13. Gorakala, S.K. *Building Recommendation Engines*; Packt Publishing: Birmingham, UK, 2016; ISBN 978-1-78588-485-6.
14. Johnson, C. Logistic matrix factorization for implicit feedback data. *Adv. Neural Inf. Process. Syst.* **2014**, *27*, 1–9.
15. Hu, Y.; Park, F.; Koren, Y.; Volinsky, C.; Park, F. Collaborative Filtering for Implicit Feedback Datasets. In Proceedings of the 2008 Eighth IEEE International Conference on Data Mining, Pisa, Italy, 15-19 December 2008.
16. Tasoulis, D.K.; Weston, D.J.; Adams, N.M.; Hand, D.J. Mining Information from Plastic Card Transaction Streams. In Proceedings of the in Computational Statistics: 18th symposium (COMPSTAT 2008),Porto 24-29 August 2008; pp. 315–322.
17. Apache Spark. Available online: https://spark.apache.org/docs/2.2.0/ml-collaborative-filtering.html (accessed on 13 August 2020).