A recommender system for service providers' campaigns

Recommendation Systems; Machine Learning; Service Provider; Analytics; Advertising Campaigns

Whitepaper

Version 1.0, September 2020

Introduction

Recommendation systems are information tools tailored to deal with information overload by suggesting items that are likely to match customers' needs and preferences. They have become a trend in the analysis and creation of customised profiles that are customer-oriented. Finding viable products and information and understand what is useful and relevant to a specific customer are key aspects to increase companies trustworthiness and revenues, decrease churn rate and ultimately increase the customer quality-of-experience.

Hence, recommendation systems take an essential role in nowadays companies and are vital to improve customer loyalty [1]. Furthermore, they help to increase the companies' product sales and create personalised advertising, inferring what product to advertise to the customer, according to their preferences and needs.

In this article, whose work is part of the scientific research article presented in the 19th EPIA Conference on Artificial Intelligence [2], we describe a recommender system applied to service provider's advertisement campaigns. This system uses historical data of customers reflecting their previous subscriptions and also data concerning customers' characteristics in terms of their behaviour and personal information. For customers that already had past subscriptions to campaigns, we applied collaborative filtering algorithms to determine what are the most suitable campaigns for them. To the customers that had not joined any campaign yet, we cross the customers' characteristics data with historical data of other customers to obtain the recommendations.

Our experiments show that the system can accurately infer recommendations for customers that purchased products or services in the past and also for the customers that don't. The results obtained show the feasibility of using recommendation algorithms to do personalised advertising. Furthermore, this study reinforces the possibility of having customer characterisation even without explicit feedback concerning the products proposed to them.

The following sections describe the methods we applied to the problem we tackle in this article and present and evaluate the meaning of the results, the strengths of this approach and its limitations. In the final section, we also describe future directions.

Proposed methodology

Service providers' advertising campaigns

A service provider relies on advertising campaigns to increase its revenues and loyalty of its customers. The company sends notifications via SMS, interactive voice response (IVR) or e-mail to the customers advertising its products and services. Having received these notifications, the customers choose to join the campaign or not. Depending on the nature of the campaign, a customer can apply to the same campaign several times within a time frame, if the events that trigger the notification of the campaign occur. This process consists of the information we use to build the recommendations for our system.

In this study, we used two types of customer data, their history of subscriptions to campaigns and their characteristics as customers of the company.

In the context of a service provider's campaigns, customers do not express their preferences in the form of ratings or likes, such as in Amazon [3]. Thus, providing campaigns' recommendations without this type of explicit feedback can be a difficult task. Regarding the historical data of customers, we transform the implicit feedback, specifically, how many times a customer joined a campaign, into explicit feedback. For this, we computed the ratio of the number of subscriptions to the number of notifications received by each customer. This operation allows us to obtain a numerical value that expresses the customer's interest in the service, i.e., a rating value for every campaign the customer received. We do not consider a binary value reflecting that the customer had joined the campaign or not because we want to distinguish the overall acceptance of each customer to a specific campaign. For instance, a customer that was notified three times and joined once is different from a customer who was notified fifteen times and also joined only once. With this ratio value, we have a more fine-grained idea of how interested a customer might be in a campaign.

This data set is very sparse since the majority of customers join very few campaigns, which is also a reason why a recommender system, able to provide more adequate campaigns for each customer, can be very advantageous for a service provider.

Recommendations

Here, we describe the process of obtaining the campaign recommendations for the two types of customers the system deals with: customers that joined campaigns in the past and customers that didn't. The collaborative filtering approach applies to the first type, while for the second type, we use customers characteristics and historical data.

Collaborative filtering

Recommendation systems use collaborative filtering algorithms for providing product recommendations based, solely, on the customers' history of purchases, searches or subscriptions in this case. Typically, these algorithms recommend to the customer products that similar customers had shown interest. The similarity is given by the common products that customers liked or bought. That is shown in **Figure 1**.

We use the data set of ratings mentioned before in this approach, dividing it into a training set, for training the collaborative filtering algorithms, and a test set, to evaluate the output obtained. This output consists of a list of campaigns, ordered by the predicted rating the algorithms calculated for the customer.

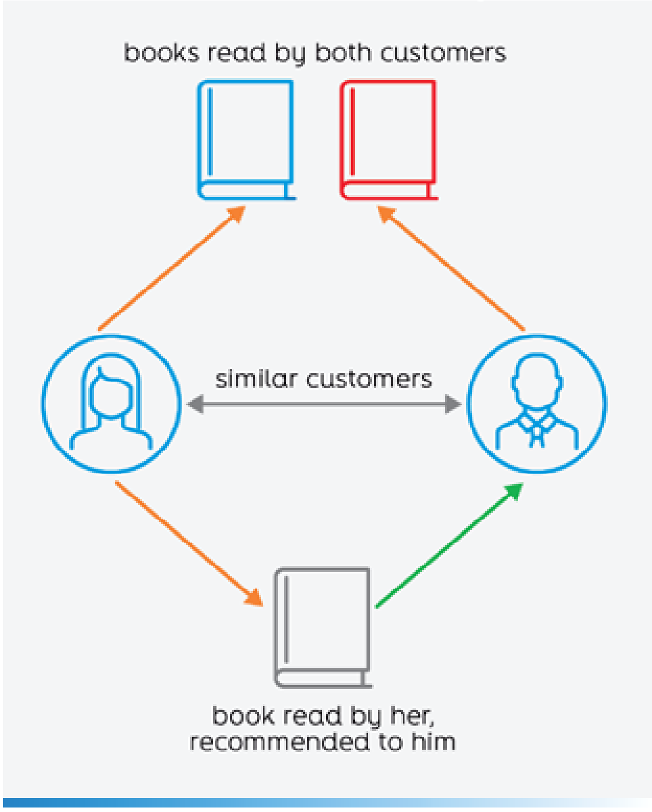

Figure 1 – Collaborative filtering

To evaluate this approach, and thus to understand the trustworthiness of the recommendations algorithms on predicting a rating value that a customer would have given to a particular campaign, we resort to several metrics. These are the mean absolute error (MAE), and the root mean squared error (RMSE), that measure how close the system predicted ratings were from the true ratings given by the customers. We also use classification accuracy metrics like precision, recall, F1-Score [4] and specificity to evaluate the correctness of the predictions. These metrics have in consideration the number of true positives, false positives and false negatives present in the list of recommendations. For this kind of denomination, we defined a threshold corresponding to the value above which a rating is considered to be positive or negative, i.e., if the rating reflects the subscriptions of the campaign. The classification in true positives, false positives, false negatives and true negatives was done according to the table and respective threshold present in **Table 1**.

	predicted rating >= threshold	predicted rating < threshold
true rating >= threshold	true positive	false negative
true rating < threshold	false positive	true negative

Table 1 – Classification of predicted ratings and true ratings for a specific threshold

We also used the mean reciprocal rank (MRR) because we want to analyse the rank of the highest-rated campaign in each users' list of recommendations. The average ranking position in the recommendation lists of every campaign is also calculated to see in which place a campaign is ranked on average, considering several customers recommendation lists.

The algorithms used to test our data set are algorithms of collaborative filtering, namely matrix factorisation and baseline estimators, among others [5].

Customers characteristics

For customers that do not have expressed their preferences on any product or service, collaborative filtering recommendation systems cannot provide recommendations. The reason is that the system does not have enough information on that customer's past to determine what products he is interested in and what are his most similar customers. In the context of collaborative filtering algorithms, this is called the cold-start problem.

To solve this problem, we use customers characteristics to get recommendations for customers that do not have historical data. We do this by applying clustering algorithms to the data set of customers characteristics to obtain groups of customers. The clustering algorithm assigns customers with similar characteristics to the same cluster. The rationale behind this approach is that similar customers will have similar preferences in terms of the products and services they like. Thus, we can recommend them the same campaigns that similar customers have joined.

- A customer receives a list of recommended campaigns, depending on the cluster assigned to him, and the campaigns he was notified.
- The list of campaigns is sorted by a numerical score which reflects the likeability of the customer to join that campaign.
- This score has in account the campaign's popularity in the cluster, as well as it's representativeness.
- The campaign's popularity in a cluster is calculated by the average rating given by the customers in that cluster.
- The campaign's representativeness, which must reflect how well each campaign is represented in each cluster, is obtained by the subscriptions' ratio times the notifications' ratio.
- The final score of a campaign, which decides the position that they are going to be recommended, is obtained by multiplying the popularity and the representativeness values.

The goal of this second approach is to be able to generate recommendations for the customers that don't have historical data. However, this approach can also give recommendations to customers that already joined campaigns in the past. For these customers, the list of recommended campaigns they receive consists of the most popular and representative campaigns of their cluster, except those that they were notified of but did not join. The reason is that we do not want to bother customers with campaigns that they already know and chose not to join. Therefore, this second approach can be complementary to the first one, or it can be used as the only way to generate recommendations to the customers.

Preliminary results

In this section, we present the results obtained for the two approaches. The results are preliminary because the second approach is yet to be evaluated by a proof-of-concept test with a service provider.

Approach with collaborative filtering algorithms

The analysed collaborative filtering algorithms were singular value decomposition (SVD), SVD++, NormalPredictor, BaselineOnly, SlopeOne and CoClustering, which are implemented in Surprise recommender library, available in the prediction_algorithms package [6].

The algorithms were evaluated with the metrics indicated previously. The threshold mentioned in the Collaborative filtering section was tested for several values since this threshold can vary according to what rating value the service provider considers to reflect the interest in a campaign.

The values for some of the mentioned metrics, computed with a threshold value of 7.5, are shown in **Table 2**. We can observe that in terms of the error metrics RMSE and MAE, the BaselineOnly and SVD algorithms show slightly better results than the other algorithms, with predicted ratings deviating on average from true ratings approx. 2.5 and approx. 1.85, for RMSE and MAE, respectively, on a scale of 0 to 10.

algorithm	RMSE	MAE	Precision	Recall	F1-Score	Specificity	MRR
BaselineOnly	2.518	1.888	83.9%	81.7%	82.7%	63.2%	0.0286
SVD	2.558	1.908	82.8%	82.5%	82.7%	59.9%	0.0287
SVD++	2.762	2.312	73.4%	94.1%	82.5%	20.6%	0.2176
CoClustering	3.143	2.537	72.6%	89.9%	80.3%	20.7%	0.0132
SlopeOne	2.959	2.433	73.6%	91.1%	81.4%	23.4%	0.0261
NormalPredictor	3.769	2.864	70.1%	58.8%	63.9%	41.3%	0.0299

Table 2 – Metric values of the different algorithms

Regarding the F1-Score metric, which combines precision and recall, it also shows good results, meaning that the predicted ratings correctly reflect the behaviour expressed in the corresponding true ratings, whether that behaviour means the customer is interested in the campaign or not. The specificity metric of the BaselineOnly and SVD algorithms show the best results, which tells us that they are more capable of identifying customers that should not get notifications about specific campaigns. Values for the MRR metric are all very similar, which implies that the customers' top-rated campaign is occupying roughly the same position in the recommendation lists of each customer. The exception is the SVD++ algorithm that places this best-rated campaign in a much higher position on the list.

As the customers receive a ranked list of campaigns, we analysed the dispersion of that campaign ranking for all the algorithms and presented the results in the heatmap graphic shown in **Figure 2**. The graphic's X-axis represents all the 177 campaigns this system deals with. The score bar represents the campaign ranking, with more saturated colours corresponding to the lower values for the ranking, i.e., the best-positioned campaigns. The graphic's Y-axis contains the six algorithms mentioned before. There are very saturated vertical lines in the graphic, which means that, for different algorithms, the same campaigns are recommended in top customers' recommendations lists. For example, around campaign 150, there are several vertical lines, meaning that some campaigns are highly recommended.

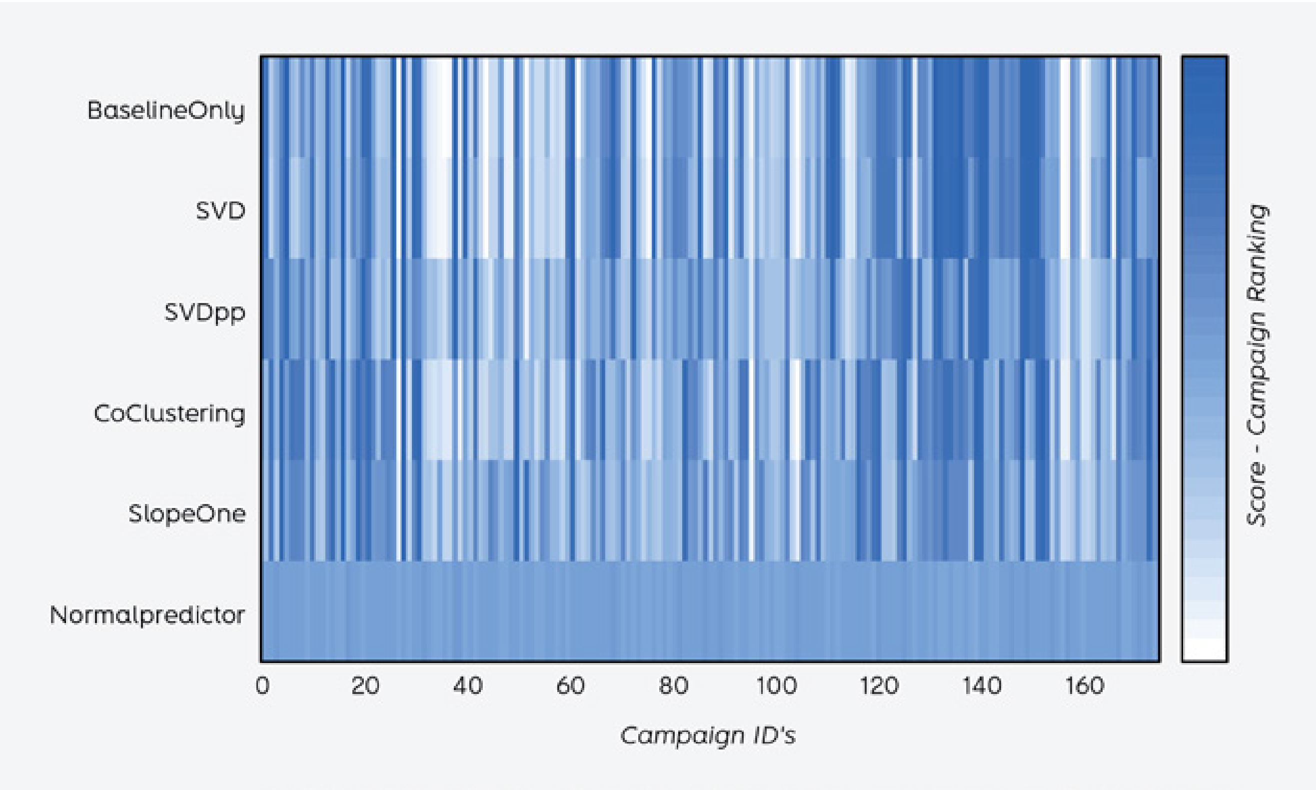

Figure 2 – Heatmap for all the campaign rankings and different algorithms

The box plot of **Figure 3** complements the analysis of the heatmap of **Figure 2**. These plots give us information about the range and distribution of the score values of the adapted ranking metric. We can see a similar behaviour between the algorithms, with the exception of the NormalPredictor algorithm, which is in compliance with the heatmap. The minimum value for the boxplot is always very low, regardless of the algorithm (exception made to the NormalPredictor) demonstrating that some campaigns are always appearing in the top recommendations.

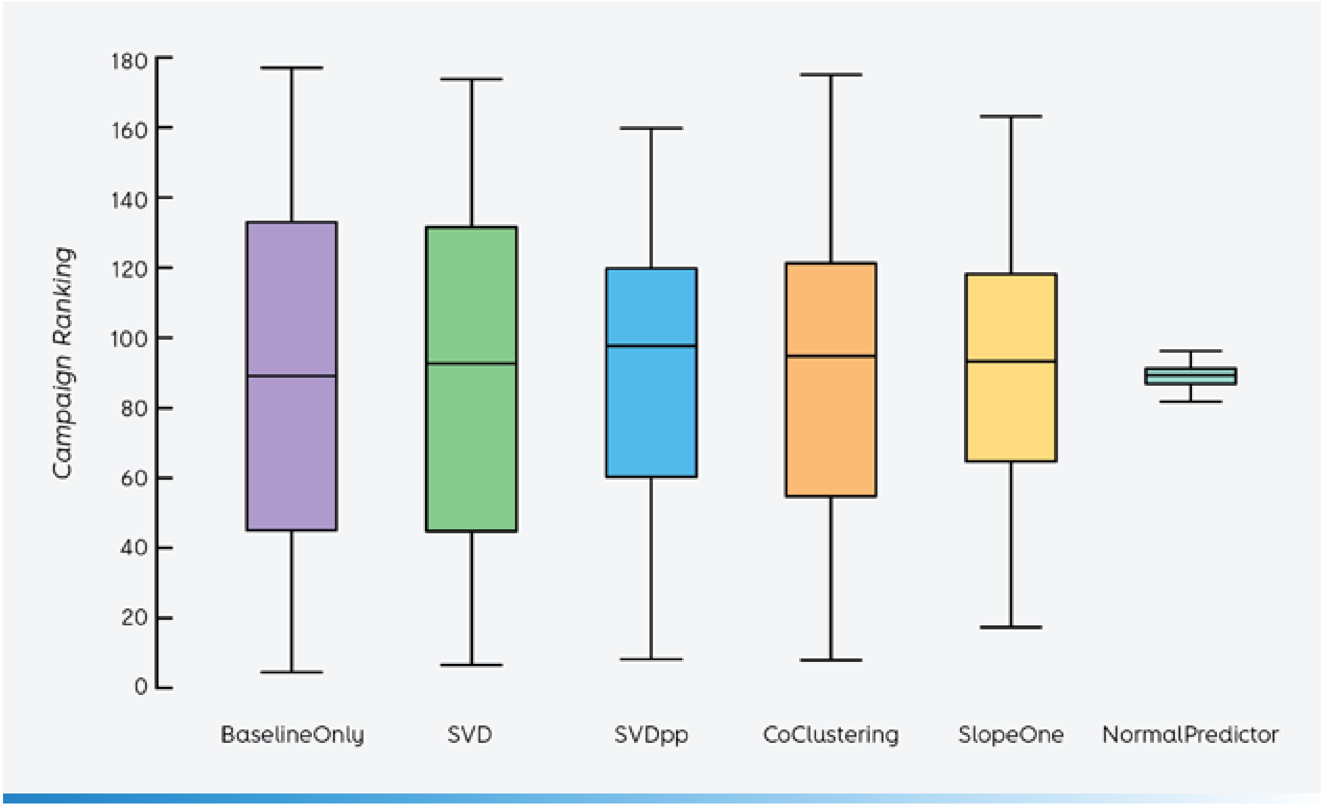

Figure 3 – Box plot for campaign ranking of different algorithms

We also measured the training and testing performance of the system. The NormalPredictor algorithm has the fastest training phase, but, as shown by the heatmap in **Figure 2**, this algorithm does not generate good recommendations. As shown in **Figure 2** and **Table 2**, the best algorithms are BaselineOnly and SVD. They present a good model training time performance and also have good results concerning our evaluation metrics.

Approach with customer characteristics

Before the recommendation phase, we evaluated the clustering model using some popular methods to execute this task, such as the elbow method [7] and the Davies-Bouldin metric [8]. The elbow method measures the intra-cluster variation, i.e., the distance of every sample in the data to their corresponding cluster centroid. The Davies-Bouldin metric measures the similarity between clusters, considering the intra-cluster and inter-cluster distances. With these two methods, we can infer the optimal number of clusters to choose for our clustering model, specifically for the data we used.

For evaluating the recommendations obtained with this approach, we intend to conduct a proof-of-concept with the service provider. This consists on notifying the company's customers with the recommended campaigns, and measure its performance, i.e., if the customers are joining the campaigns they are being notified of, and therefore, generating profit to the company.

As an example, **Table 3**, on next page, shows the campaigns considered for the recommendations of customers in one of the clusters. In this example, we only used a subset of six campaigns.

campaign	number of notifications	number of subscriptions	represent. - notifications	represent. - subscriptions	mean rating	score
campaign 1	412	280	0.12022	0.84337	7.28870	0.99129
campaign 2	1658	14	0.48380	0.04216	0.10035	0.00274
campaign 3	988	15	0.28829	0.04518	0.1567	0.00272
campaign 4	150	13	0.04377	0.03915	0.89655	0.00206
campaign 5	35	10	0.01021	0.03012	2.85714	0.00117
campaign 6	184	0	0.05369	0	0	0

Table 3 – Example of popular and representative campaigns in one of the clusters

The columns represent. - notifications and represent. - subscriptions indicate the representativeness of the campaigns, in relation to notifications and subscriptions, respectively. These columns derive directly from the number of notifications and number of subscriptions columns, consisting of a ratio of the respective class (notifications or subscriptions). The mean rating column indicates the mean rating values given by the customers in that cluster. This value represents the popularity of the campaign.

As mentioned before, we obtained the score by considering the representativeness of the campaign and its popularity. The score indicated in the table uses a scale from 0 to 1. So, the top one campaign, the one with the highest score, represents the best recommendation for the customer in that cluster. If that customer already received notifications for that campaign and did not subscribe it, it is recommended with the next one on the list, and so on.

Conclusion and future work

In a world with a plethora of products and services, it is a herculean task for a customer to identify good offers for his needs. Hence, recommendation systems have an essential role in ensuring the best experience and quality-of-service.

With this study, we explore the possibility of recommending advertising campaigns to the customers of a service provider. These campaigns will advertise products or services more suited to the customer's needs and interests.

The system bases its recommendations on collaborative filtering and clustering algorithms, using customer-related data, such as subscriptions history to campaigns and personal characteristics. We considered these two approaches because collaborative filtering may suffer from the cold-start problem, and the customer characteristics approach can address that.

We analysed and evaluated distinct state-of-the-art algorithms. Our results allow us to infer the best one to use for recommendations for customers that have historical data. We based this decision in metrics evaluating the quality of the recommendations and the performance of the algorithm. The recommendations obtained from the clustering approach, aimed at the customers that did not join any campaign in the past, will be evaluated in a real-world scenario, with real customers being notified with their recommended campaigns. This evaluation step consists of a proof-of-concept being executed alongside with the service provider that provided the data.

Although the second approach of this study is yet to be completed, we believe in the methodology's feasibility for our recommender system. Overall, we acknowledge the advantages of recommender algorithms applied to a service provider's advertising campaigns but also recognise the challenges required to obtain good recommendations with the data we have available.

References

[1] F. Ricci, L. Rokach, and B. Shapira, "Introduction to Recommender Systems Handbook," in ***Recommender Systems Handbook, Springer***, 2010, pp. 1–35.

[2] D. Alves, B. Valente, R. Filipe, M. Castro, and L. Macedo, "A Recommender System for Telecommunication Operators' Campaigns," in ***Progress in Artificial Intelligence***, 2019, pp. 83–95.

[3] G. Linden, B. Smith, and J. York, "Amazon.com recommendations: item-to-item collaborative filtering," ***IEEE Internet Comput.***, vol. 7, no. 1, pp. 76–80, 2003.

[4] K. P. Shung, "Accuracy, Precision, Recall or F1?," Towards Data Science, 2018. [Online]. Available: https://towardsdatascience.com/accuracy-precision-recall-or-f1-331fb37c5cb9.

[5] "Surprise prediction_algorithms package," 2019. [Online]. Available: https://surprise.readthedocs.io/en/stable/prediction_algorithms_package.html.

[6] "Surprise, a Python scikit for recommender systems," 2019. [Online]. Available: http://surpriselib.com/.

[7] R. Gove, "Using the elbow method to determine the optimal number of clusters for k-means clustering," 2017. [Online]. Available: https://bl.ocks.org/rpgove/0060ff3b656618e9136b.

[8] D. L. Davies and D. W. Bouldin, "A Cluster Separation Measure," ***IEEE Trans. Pattern Anal. Mach. Intell.***, vol. PAMI-1, no. 2, pp. 224–227, 1979.

Authors

Ricardo Ângelo Filipe

Data Scientist

Altice Labs, Portugal

ricardo-a-filipe@alticelabs.com

https://www.linkedin.com/in/ricardo-filipe-0a589787/

Diogo André Cadima Alves

Data Scientist

Altice Labs, Portugal

diogo-c-alves@alticelabs.com

https://www.linkedin.com/in/diogo-alves-910563144/

Maria Manuel Castro

Recommendations Platform and BI Analytics Technical Manager

Altice Labs, Portugal

maria-m-castro@alticelabs.com

Contacts

Address

Rua Eng. José Ferreira Pinto Basto
3810 - 106 Aveiro (PORTUGAL)

Phone

+351 234 403 200

Media

contact@alticelabs.com
www.alticelabs.com