Development of a Hybrid Real Estate Recommender System

Hakan Tas[*]
Zingat.com,
Istanbul, Turkey

Hilal Erdogan Sumnu
Zingat.com,
Istanbul, Turkey

Bahadir Gokoz
Zingat.com,
Istanbul, Turkey

Tevfik Aytekin
Bahcesehir University
Istanbul, Turkey

Abstract: In the current study, the details of a real estate recommender system developed for Zingat.com are discussed. The system developed is a hybrid of collaborative and content filtering approaches. Scalable methods in both the model building phase and in the recommendation list generation phase were used to work on the data set of the project (as of 2018, 300k listings and 6 million monthly sessions). This study also explained the challenges faced in developing and implementing the system, the recommendation techniques used to overcome these challenges, and the final product used for recommendation. Based on these, future recommendations are discussed.

I. INTRODUCTION

A recommendation system is an extensive class of web applications that involves predicting the user responses to the options. It has been a hot topic for a long time. It is a simple algorithm which aim to provide the most relevant and accurate items to the user by filtering useful stuff from of a huge pool of information base. Recommender system, often termed as recommendation engine, discovers data patterns in the data set by learning consumers choices and produces the outcomes that co-relates to their needs and interests. Recommender system has been used successfully applied to overcome to information overload in various domains such as movies and TV [1], news [2], e-commerce [3], and travel [4]. Compared to other domains, applications and academic work in real estate recommendation is rather limited. The leading real estate web portals in this area, Trulia and Zillow , have developed their own recommender systems. And to the best of our knowledge, there are no in-house developed recommender system at other real estate web sites in Turkey.

There are companies which provide recommendation services using a SaaS approach, however, these services have high costs and in our tests did not perform well since they are not specifically developed for the real estate domain. A solution which is specifically designed for the real estate domain will enable us to provide a better service and increase our competitiveness.

Compared to the other domains there are differences in the nature of the real estate recommender system: for example, in a movie recommender system, the type of the movie, actors, and the director can be a feature of the movie however, in a real estate recommender system, two houses which have the same set of features can be two different products when their locations are different. Here, the location of a property is a special parameter among others. Similarly, not only the features of a property, but also the regional characteristics of its location also comes into play. For example, the demographic structure of the region where the real estate is located and regional characteristics such as the education level have to be taken into consideration. In this sense, the project to be realized differs from the already known recommender system types and shows originality. These considerations show the need for developing recommendation techniques specifically designed for real estate domain.

The long tail phenomenon seen on e-commerce sites also applies to Zingat.com. A small number of ads are displayed frequently, and a large number of ads are displayed infrequently. Although recommender systems

---

[*]Correspondence concerning this article should be addressed to Hakan Tas, Zingat.com, Istanbul, Turkey. E-mail: hakan.tas@zingat.com

are proposed to solve this problem, it is known that recommender systems generally contribute to the formation of a long tail [5, 6]. Improving the diversity of recommendations is one of the active research areas in recommender systems where algorithms are developed to improve the diversity of recommendations without harming the accuracy much [7, 8]. The current project also measured the diversity of recommendations lists in order to take precautions if needed.

The field of large-scale recommender systems is another active research topic [9]. The size of the data set in our project (as of 2018, 300k listings and 6 million monthly sessions) requires the use of scalable methods in both the model building phase and in the recommendation list generation phase.

II.   BACKGROUND

Recommender systems emerge as a solution to the information overloading problem. As the number of products and services increase it is getting more difficult for consumers to find the products among the alternatives. There are basically two different recommendation techniques: content-based filtering [10] and collaborative filtering [11]. Content-based filtering uses the content information of products in order to build user profiles based on users past transactions. For example, in a movie recommender system, the genre, actors, director, and the script of a movie might be used for building users past likes and dislikes. Once such a user profile is built, users can be recommended movies which match their profiles. Collaborative filtering, on the other hand, does not use content information of products but only uses interactions between users and products such as product views and purchases. There are two basic types of collaborative filtering: neighborhood based [12] and matrix factorization [13]. And, there are two types of neighborhood-based algorithms: user and item based. In user-based collaborative filtering a user is recommended products which are liked by similar users to the target user. In item-based collaborative filtering a user is recommended products which are similar to the products like by the user in the past. Matrix factorization-based techniques find latent feature representations of users and items in a low dimensional space and use these representations to make predictions about whether a user will like a product or not. Hybrid methods are also possible [14] which combines the different techniques to overcome their limitations. For example, if there are not enough interactions of a specific product (named the cold start) then that product might be recommended first using its content information and later when enough information is collected the system might

switch to collaborative filtering.

The major aim in recommender systems is to improve the accuracy of recommendations, however, recently it has been recognized that there are other properties of recommender systems which are important for user satisfaction. One of them is the diversity of recommendations [15]. There are two aspects of diversity: individual and aggregate diversity. Individual diversity [16] measures how diverse an individual recommendation list is. On the other and aggregate diversity [17] measures how diverse the recommended items across all recommendation lists. There are two generally used measures of aggregate diversity which are defined below.

$$DiversityInTopN = |\cup_{(u \in U)} L(u)| \qquad (1)$$

$$GiniDiversity = 2 \sum_{i=1}^{n} \left[\left(\frac{n+1-i}{n+1} \right) \times \left(\frac{rec(i)}{total} \right) \right] \quad (2)$$

where U is the set of users, $L(u)$ is the recommendation list of user u, $rec(i)$ is the number of times item i appears in the recommendations list, and total is the total number of recommendations. Regarding to above equations, "DiversityInTopN" simply measures the total number of unique items that appear in the recommendation lists of all users. "GiniDiversity" measures how even the recommended items are distributed. In the following section, results with respect to the measures are presented.

III.   METHODS AND MATERIALS

As for collaborative filtering, two main approaches are tried. One is called the item based collaborative filtering [18]. This is one of the most widely used algorithms in industry. The reasons for its wide adoptance include its ease of implementation, high accuracy, and being interpretable. One drawback of this algorithm is its quadratic time complexity with respect to the number of items since it needs to find all pairwise item similarities. However, this step is done offline and it could be done under 2 hours which is more than enough for our purposes. For every item, the most similar 50 items are stored in a key-value store for fast retrieval. In this way the system can respond to heavy user traffic. Item similarities are calculated using cosine similarity between item vectors where item vectors store information about whether a user is interested in an item (a specific listing).Different implicit feedback types are used to find out whether a user is interested in an ad or not, these include: listing view, phone view, click to whatsapp icon, sending a message to the property owner, and photo views.

Another commonly used recommender algorithm

named Alternating Least Squares (ALS) [19] is used. This algorithm is a matrix factorization based method which can use implicit feedback from users. Similar to item-based method here, implicit feedback data to build the ALS model is used.

For both methods following evaluation protocol are used: For every user we take out one of her views from the dataset and put it into the test set. Remaining view data is used for training purposes. After the model is build using the test set, for every user in the test set a recommendation list of 5 items and calculate F1-score is generated.

Table 1 gives the F1-score values of the item-based neighborhood method for different values of k (number of neighbors). Table 2 gives the F1-score values with the ALS method for different values of f (number of factors). As can be seen from the tables item-based CF give better results compared to ALS algorithm, so it is decided to implement item-based CF in our system.

TABLE 1
ITEM-BASED CF

F1-score	k
0.35	20
0.36	40
0.38	60
0.40	80
0.39	100

TABLE 2
ALTERNATING LEAST SQUARES

F1-score	f
0.32	10
0.33	20
0.34	30
0.36	40
0.36	50

In addition to collaborative filtering a content-based recommendation system is also developed. Content-based system is similar to the collaborative filtering model in that here also pairwise similarities between items are found. But this time instead of implicit feedback from users the features of properties are used. After a through experiment it is found that the three features, namely, price, area, and location are the most important features and they are correlated with the other features with regard to similarity. So, it is decided to use these three features only in similarity computation. A subjective evaluation of the similarity results show that the method finds very similar properties. Variations of the following general formula to calculate the distance between two properties

x and y are used.

$$Similarity(x,y) = \alpha_1 \frac{(min(area_x, area_y))}{(max(area_x, area_y))} + \alpha_2 \frac{(min(price_x, price_y))}{(max(price_x, price_y))} + {}_3Distance_{xy}$$

(3)

Where min() and max() functions which return the minimum and maximum value of their arguments respectively. α_i's are weight values of the corresponding features. $istance_{xy}$ is calculated by taking the ratio of the distance between x and y by the largest distance in the same district.

One problem with the above approach is the fact that for every property there will be a most similar property even though they are not very similar. This leads to poor recommendations, such as recommending a house whose area is very different than the target house. To prevent this, some threshold values such that only houses below this threshold (for every feature) are recommended. These ratios by analyzing the properties a user visits in a single session are found. For every such session the price, area, and location sensitivity of the users is calculated. The results we found are represented in the below tables for two different cities, namely, Ankara and Izmir.

TABLE 3
SENSITIVITY VALUES FOR ANKARA

	Min. Value	Max. Value
Price	-12%	8%
Area	-6%	12%
Location	-7%	7%

Table.4

TABLE 4
SENSITIVITY VALUES FOR IZMIR

	Min. Value	Max. Value
	Min. Value	Max. Value
Price	-14%	9%
Area	-10%	13%
Location	-5%	5%

Min and max values are the minimum and maximum values represented as percentages with respect to the mean of the values over all the sessions. For example, price sensitivity values for Ankara shows that on average users visit houses whose prices are 12% less than (and 8% more than) the mean of the house prices visited in that session. The calculation of the location sensitivity is somewhat different. At first, the average pairwise distance between the houses visited in a single session is found and divide this value by the average pairwise

distance between all the houses in that district. In the end the thresholds are set according to these results and this method let us prevent some dissimilar houses to be recommended.

Both of these models (collaborative filtering and content filtering) are kept in memory and use them for different business purposes in a hybrid way. For example, when there is not enough feedback for a particular listing, resort to the content model or sometimes in order to increase diversity we mix the recommendations from both models to build a recommendation list.

We also measure diversity values of the recommendation lists using the measures defined above. DiveristyIn-TopN, as a fraction of the total number of unique listings, turned out to be 0.82 which is more than we expected. This means that most of the listings are recommended to the users. On the other hand, GiniDiversity turned out to be 0.3 which shows that the items are close to being evenly distributed in the recommendation lists. In the future we plan to use special methods [16, 17] to further improve these results without harming accuracy much.

IV.   SYSTEM ARCHITECTURE

The behavior of Zingat.com users such as listing views, phone views, click on whatsapp icon, messages sent, and photo views are transferred to the Divolte service in the Z-metrics project. Then mapping type is checked for the incoming data from the relevant service. If the data can be validated, the relevant data is transferred to the Kafka service. The data coming to Kafka service is processed in the queue with python stream processing service and the appropriate data is written to Cassandra service with Z-metric id. In addition, user information with the Z-metrics id is processed into the Mysql database.
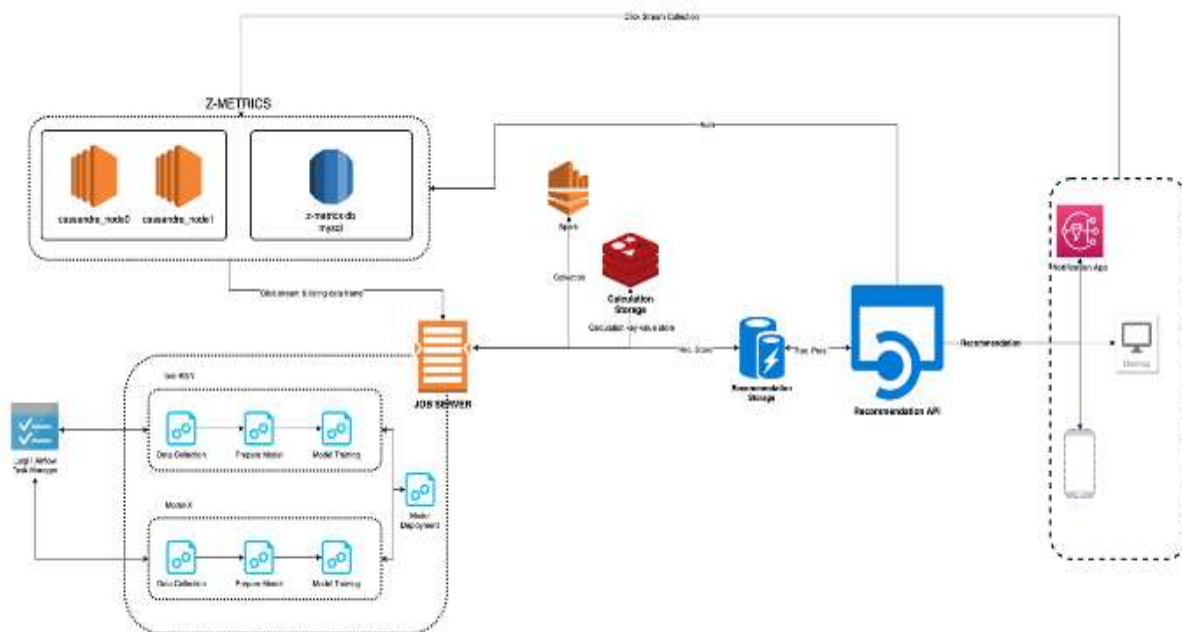

Fig. 1. System architecture

The above operations are performed for each event. At the end of the day, the click stream data collected on Cassandra, and the current algorithm and user behavior data are inserted into the train model and the model is retrained. After the model is trained, recommendations are created for each listing and transmitted to the Recommendation API. Recommendation data collected on the API is returned to users via requests to the API.

V.   CONCLUSION AND RECOMMENDATIONS

Recommendation systems, as discussed, are data filtering tools that make use of algorithms and data to recommend the most relevant items to a particular user. In this work we explained how we implemented a fully functional recommendation system for property listings. Since the number of listings is huge and new listings come and go frequently it is a challenge to build a successful recommender system. We tackled this challenge by building a hybrid system which uses both collaborative filtering and content-based filtering. We also designed a scalable system architecture which can function under heavy load. In the future we plan to further improve this system by using diversification techniques and new hybrid solutions. This study encourages scholars to de-

sign and implement similar systems by taking the current system as a reference.

Declaration of Conflicting Interests

Authors hereby declare that this work has no known conflicting interests.

REFERENCES

[1] M. A. Beam, "Automating the news," *Communication Research*, vol. 41, no. 8, pp. 1019–1041, 2013. doi: https://doi.org/110.1177/0093650213497979

[2] M. S. Pera and Y.-K. Ng, "A group recommender for movies based on content similarity and popularity," *Information Processing & Management*, vol. 49, no. 3, pp. 673–687, 2013. doi: https://doi.org/10.1016/j.ipm.2012.07.007

[3] K. Wei, J. Huang, and S. Fu, "A survey of e-commerce recommender systems," in *International Conference on Service Systems and Service Management,* Chengui, China, 2007.

[4] R. Logesh, V. Subramaniyaswamy, and V. Vijayakumar, "A personalised travel recommender system utilising social network profile and accurate GPS data," *Electronic Government, an International Journal*, vol. 14, no. 1, pp. 90–113, 2018. doi: https://doi.org/10.1504/eg.2018.089538

[5] D. M. Fleder and K. Hosanagar, "Recommender systems and their impact on sales diversity," in *Proceedings of the 8th ACM Conference on Electronic Commerce*. ACM Press, 2007.

[6] S. N. Cubero, S. McLernon, and A. Sharpe, "Overspeeding warning system using wireless communications for road signs and vehicles," *Journal of Advances in Technology and Engineering Studies*, vol. 2, no. 5, pp. 140–155, 2016. doi: https://doi.org/10.20474/jater-2.5.2

[7] M. Kunaver and T. Požrl, "Diversity in recommender systems – a survey," *Knowledge-Based Systems*, vol. 123, pp. 154–162, 2017. doi: https://doi.org/10.1016/j.knosys.2017.02.009

[8] M. Iuliana, G. Nicolae, and C. S. Razvan, "Web application for self-diagnosis and drug recommendation based on user symptoms," *Journal of Advances in Technology and Engineering Research*, vol. 5, no. 2, pp. 62–71, 2019. doi: https://doi.org/10.20474/jater-5.2.1

[9] T. Ye, D. Parra, V. Ostuni, and T. Wang, "Workshop on large-scale recommender systems," in *Proceedings of the 11th ACM Conference on Recommender Systems,* Hong Kong, China, 2017.

[10] M. de Gemmis, P. Lops, C. Musto, F. Narducci, and G. Semeraro, "Semantics-aware content-based recommender systems," in *Recommender Systems Handbook*. Boston, MA: Springer, 2015, pp. 119–159.

[11] M. D. Ekstrand, "Collaborative filtering recommender systems," *Foundations and Trends® in Human–Computer Interaction*, vol. 4, no. 2, pp. 81–173, 2011. doi: https://doi.org/10.1561/1100000009

[12] D. Christian and G. Karypis, "A comprehensive survey of neighborhood-based recommendation methods," in *Recommender Systems Handbook*. Boston, MA: Springer, 2011, pp. 107–144.

[13] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, 2009. doi: https://doi.org/10.1109/mc.2009.263

[14] R. Burke, "Hybrid recommender systems: Survey and experiments," *User Modeling and User-Adapted Interaction*, vol. 12, no. 4, pp. 331–370, 2002. doi: https://doi.org/10.1023/a:1021240730564

[15] P. Castells, N. J. Hurley, and S. Vargas, "Novelty and diversity in recommender systems," in *Recommender Systems Handbook*. Boston, MA: Springer, 2002, pp. 881–918.

[16] T. D. Noia, J. Rosati, P. Tomeo, and E. D. Sciascio, "Adaptive multi-attribute diversity for recommender systems," *Information Sciences*, vol. 382-383, pp. 234–253, 2017. doi: https://doi.org/10.1016/j.ins.2016.11.015

[17] M. O. Karakaya and T. Aytekin, "Effective methods for increasing aggregate diversity in recommender systems," *Knowledge and Information Systems*, vol. 56, no. 2, pp. 355–372, 2017. doi: https://doi.org/10.1007/s10115-017-1135-0

[18] B. Sarwar, G. Karypis, J. Konstan, and J. Reidl, "Item-based collaborative filtering recommendation algorithms," in *Proceedings of the 10th International Conference on World Wide Web - WWW - 01,* Hong Kong, 2001.

[19] Y. Hu, Y. Koren, and C. Volinsky, "Collaborative filtering for implicit feedback datasets," in *8th IEEE International Conference on Data Mining,* Pisa, Italy, 2008. doi: https://doi.org/10.1109/icdm.2008.22