

Recommendation Systems Applied in Turkish Real Estate Market

**Mehmet Erkek¹, Kamil Çayırılı², Hakan Taş³, Ali Hepşen⁴
and Tefvik Aytekin⁵**

Abstract

Today consumers are confronted with a very large number of products and services to choose from. This makes it difficult for users to find relevant products among a huge number of alternatives. A recommendation system is an extensive class of web applications that involves predicting the user responses to the options and helps users to find products of interest by analyzing their past transactions such as product views and purchases. There is also a similar problem in the real estate industry where thousands of properties are available for rentals or sales. In this work we firstly presented the details of a real estate recommender system developed for Zingat.com and then, we explained how we implemented a fully functional recommendation system for property listings in Turkey. 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 system which uses collaborative filtering and content-based filtering, separately. 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 solutions.

Mathematics Subject Classification: 97R40

Keywords: Artificial intelligence, machine learning, real estate, recommendation engine, zingat.com

¹ Zingat.com.

² Vocational School, Nisantasi University.

³ Zingat.com.

⁴ Faculty of Business Administration, Istanbul University.

⁵ Faculty of Engineering and Natural Sci., Bahcesehir Univ.

1. Introduction

A recommendation system is an extensive class of web applications that involves predicting the user responses to the options [2]. 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 a huge pool of information base [1]. Recommender system, often termed as recommendation engine, discovers data patterns in the dataset 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 [14], news [1], e-commerce [16], and travel [13]. 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 (for more information: <https://www.trulia.com/blog/tech/recommender-engine/>) and Zillow (for more information: <https://www.slideshare.net/njstevens/recommendations-at-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 are proposed to solve this problem, it is known that recommender systems generally contribute to the formation of a long tail [8]. 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 [12]. In our project we also measure the diversity of recommendations lists in order to take precautions if needed.

The field of large-scale recommender systems is another active research topic [17]. 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.

2. 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 [4] and collaborative filtering [7]. 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 [5] and matrix factorization [11]. 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. 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 [3]. There are two aspects of diversity: individual and aggregate diversity. Individual diversity [6] measures how diverse an individual recommendation list is. On the other and aggregate diversity [10] measures how diverse the recommended items across all recommendation lists. There are two generally used measures of aggregate diversity which are defined below.

Theorem 1.

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

Theorem 2.

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

where U is the set of users, $L(u)$ is the recommendation list of user u , $\text{rec}(i)$ 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 we will provide our results with respect to the measures.

3. Methodology

As for collaborative filtering we tried two main approaches. One is called the item based collaborative filtering [15]. 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 we can do it 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). We use different implicit feedback types 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.

We also tried another commonly used recommender algorithm named Alternating Least Squares (ALS) [9]. This algorithm is a matrix factorization based method which can use implicit feedback from users. Similar to item-based method here we also use the implicit feedback data to build the ALS model.

For both methods we use the following evaluation protocol: 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 built using the test set, for every user in the test set we generated a recommendation list of 5 items and calculate F1-score.

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 we get 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 we 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: ALS

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

In addition to collaborative filtering we also developed a content-based recommendation system. 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 we 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, we 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. We use variations of the following general formula to calculate the distance between two properties x and y .

Theorem 3.

$$Similarity(x, y) = \alpha_1 \frac{\min(area_x, area_y)}{\max(area_x, area_y)} + \alpha_2 \frac{\min(price_x, price_y)}{\max(price_x, price_y)} + \alpha_3 Distance_{xy}$$

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. $Distance_{xy}$ 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, we set some threshold values such that only houses below this threshold (for every feature) are recommended. We find these ratios by analyzing the properties a user visits in a single session. For every such session we

calculated the price, area, and location sensitivity of the users. The results we found are represented in the below tables for two different cities, namely, Ankara and İzmir.

Table 3: Sensitivity Values for Ankara

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

Table 4: Sensitivity Values for İzmir

	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. We first find the average pairwise distance between the houses visited in a single session 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.

We also measure diversity values of the recommendation lists using the measures defined above. DiveristyInTopN, 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 [10; 6] to further improve these results without harming accuracy much.

4. 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.

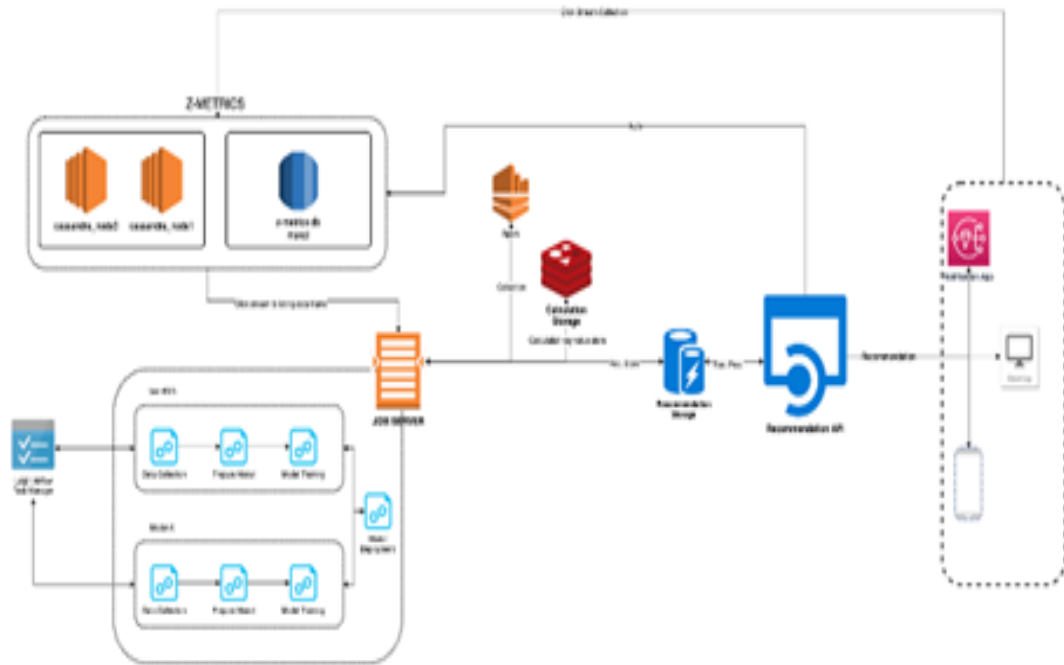


Figure 1: System Architecture

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.

5. Conclusion

Recommendation systems, as discussed, are an extensive class of web applications that involves predicting the user responses to the options and helps users to find products of interest by analyzing past user transactions such as product views and purchases. There is also a similar problem in the real estate industry where thousands of properties are available for rentals or sales. In this work we firstly presented the details of a real estate recommender system developed for Zingat.com and then, we explained how we implemented a fully functional recommendation system for property listings in Turkey. 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 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 solutions.

References

- [1] A.M. Beam, Automating the news: How personalized news recommender system design choices impact news reception. *Communication research*, 41(8), (2014), 1019-1041.
- [2] R. Burke, Hybrid recommender systems: Survey and experiments, *User modeling and user-adapted interaction*, 12(4), (2002), 331-370.
- [3] P. Castells, N.J. Hurley and S. Vargas, Novelty and diversity in recommender systems, *Recommender systems handbook*, Springer, Boston, MA, 2015.
- [4] M. De Gemmis, et al., Semantics-aware content-based recommender systems, *Recommender systems handbook*, Springer, Boston, MA, 2015.
- [5] C. Desrosiers and G. Karypis, A comprehensive survey of neighborhood-based recommendation methods, *Recommender systems handbook*, Springer, Boston, MA, 2015.
- [6] T. Di Noia, et al., Adaptive multi-attribute diversity for recommender systems, *Information sciences*, 382, (2017), 234-253.
- [7] M.D. Ekstrand, J.T. Riedl and J.A. Konstan, Collaborative filtering recommender systems, *Foundations and trends in human-computer interaction*, 4(2), (2011), 81-173.
- [8] D.M. Fleder and K. Hosanagar, Recommender systems and their impact on sales diversity, *Proceedings of the 8th ACM conference on electronic commerce*, (2007), 192-199.
- [9] Y. Hu, Y. Koren and C. Volinsky, Collaborative filtering for implicit feedback datasets, *2008 Eighth IEEE international conference on data mining*, (2008), 263-272.
- [10] M.Ö. Karakaya and T. Aytakin, Effective methods for increasing aggregate diversity in recommender systems, *Knowledge and information systems*, 56(2), (2018), 355-372.
- [11] Y. Koren, R. Bell and C. Volinsky, Matrix factorization techniques for recommender systems, *Computer*, 8, (2009), 30-37.
- [12] M. Kunaver and T. Požrl, Diversity in recommender systems—A survey, *Knowledge-based systems*, 123, (2017), 154-162.
- [13] R. Logesh, V. Subramaniaswamy and V. Vijayakumar, A personalised travel recommender system utilising social network profile and accurate GPS data, *Electronic government, an international journal*, 14(1), 2018, 90-113.
- [14] M.S. Pera and Y.K. Ng, A group recommender for movies based on content similarity and popularity, *Information processing & management*, 49(3), (2013), 673-687.
- [15] B.M. Sarwar, et al., Item-based collaborative filtering recommendation algorithms, *The Tenth International World Wide Web Conference*, (2001), 285-295.
- [16] K. Wei, J. Huang and S. Fu, A survey of e-commerce recommender systems, *2007 International conference on service systems and service management*, (2007), 1-5.

- [17] T. Ye, et al., LSRS'17: Workshop on large-scale recommender systems, Proceedings of the eleventh ACM conference on recommender systems, (2017).