Developing Tourism Users' Profiles With Data-driven Explicit Information

Rasoul Norouzi[a], Hamed Baziyad[a], Elham Akhondzadeh[b*], Amir Albadvi[c]

[a] M.Sc., Department of Information Technology, Faculty of Industrial and Systems Engineering, Tarbiat Modares University, Tehran, Iran

[b] Assistant Professor, Department of Information Technology, Faculty of Industrial and Systems Engineering, Tarbiat Modares University, Tehran, Iran

[c] Professor, Department of Information Technology, Faculty of Industrial and Systems Engineering, Tarbiat Modares University, Tehran, Iran

Abstract

In recommender systems (RSs), explicit information is often preferred over implicit because it is much more accurate than implicit or predicted information, e.g., the user can enter information about his interests directly into the system, and the system will generate accurate recommendations for him. Receiving explicit information, however, may be difficult for a system. Explicit demographic information might be uncomfortable for some users, and extremely common questions, such as race, gender, income, and age, can lead to bias and unfair recommendations. As a result, in this study, we propose a method in which information collected from a new user does not contain demographic information, and enquired explicit information is data-driven. Users' interest in tourism activities is used to identify seven categories of tourism. The mapping between extracted categories and activities is established with a multi-label classification (MLC) algorithm. The user's interest in 18 tourism activities is predicted by rating only seven tourism categories. Common MLC algorithms with different classifiers were used to evaluate the proposed method. It has been found that the best performance was achieved for binary relevance with the naive Bayes classifier, which was also superior to the collaborative filtering system models used as baselines. The proposed method can capture users' interests and develop their profiles without receiving demographic information. Also, compared to CF, in addition to a slight advantage in metrics, it only requires seven ratings to predict user interest in 18 activities. In contrast, CF algorithms require at least 15 user ng records to predict user interest in unknown activities (3-4 activities) to achieve a performance close to the proposed method.

[*] Corresponding author
Email address: elham.akhondzadeh@modares.ac.ir (Elham Akhondzadeh Noughabi)

Keywords: Recommender System, Multi-Label Classification, Cold Start, Explicit information, Tourism

1. Introduction

Personalization is the ability to provide tailored content and services to users based on the knowledge about their preferences and tastes [1]. Personalization techniques are mainly related to recommender systems (RS), which aim to filter irrelevant information and to provide personalized information to each particular user [2]. RS can be defined as a personalization tool that provides people with a list of items that best fit their individual preferences, restrictions, or tastes [3]. One of the interesting applications of RS lies in trip planning area [4].

Tourists are often confused about where to go when reaching new and unfamiliar places as there could be a wide variety of choices for consideration [5]. Besides, they typically have a limited amount of time and budget available; thus, it is almost impossible to visit all tourist attractions during a trip, especially to large cities [6]. As a result, tourists have to select the most compelling points of interest (POIs) according to their preferences. Then, they plan an itinerary, taking into account the time available to reach the POIs concerning their accessibility and opening hours [7].

The use of modern technologies such as collaborative filtering (CF) of classical recommender systems is considered an effective solution within the tourism industry [8]. CF is one of the recommender systems' approaches, helping people make their choice based on the opinions of those similar to them. The similarity between users is calculated based on the scores they have given to the list of items. When the system finds out which people are closer to each other based on their interests and choices, other "similar" users' favorites are suggested to the intended user. In this approach, to find out which recommendation is favorable and which is not, obtaining feedback is necessary. CF systems use user-item matrix to predict users' interest in items. In such matrices, each row, column and cell respectively represent a user, an item, and a user giving rate to an item [9].

A further paradigm in cross-domain collaborative filtering is proposed by Yu et al., (2019) [10], in which a model is proposed that solves the problem of different auxiliary domains' importance in the target domain. They propose a cross-domain collaborative filtering algorithm that takes advantage of latent factors in auxiliary domains to expand user and item features. In the proposed algorithm, the recommendation is formulated as a classification problem in the target domain, where user and item location serve as features and ratings as labels. Then, Funk-SVD decomposition is employed to extract extra user and item features from user- and item-side auxiliary domains, respectively, with the purpose of expanding the two-dimensional location feature vector. In the final step, a C4.5 decision tree algorithm was used to predict missing ratings. To balance recommendation accuracy and efficiency, Yu et al., (2021) [11] examines how to select significant subsets from all the auxiliary domains. A Two-Sided CDCF based on Selective

Ensemble Learning is proposed that considers both accuracy and efficiency (TSSEAE). The model solves a bi-objective optimization problem for selective ensemble learning, concentrating on a subset of auxiliary domains to achieve a balance between accuracy and efficiency.

Despite the high effectiveness of CF-based recommender systems in the tourism area, they suffer from two main challenges issues of sparsity and cold-start. *Sparsity* happens when many user-item matrix cells suffer from the lack of rates given by users [12, 13]. This makes training of machine learning models, especially in memory-based algorithms, challenging. By growing the number of items and users, the sparsity and the dimension of the user-item problems become more severe and more problematic. To solve the problem of sparsity, Yu et al., (2018) propose a two-sided cross-domain collaborative filtering model. It is assumed that there are two auxiliary domains, i.e., a user-side domain and an item-side domain, where the user-side auxiliary domain shares the same aligned users, and the item-side has the same aligned items. As a first step, they conceptualize the user and item features in the context of bi-orthogonal tri-factorization. The recommendation problem is then converted into a classification problem, using the inferred user and item features as feature vectors and the rating as the class label. Using both user-side and item-side shared information, the model can transfer knowledge from auxiliary domains more effectively and infers domain-independent user and item features.

The *cold-start* problem occurs when entering a new user into the system; since there is no record of the user interests and rates to the items, it is impossible to predict what the user would be attracted to. The same problem can exist for newly added items, which in literature, it refers to items cold-start. The cold-start problem is usually handled by using hybrid systems or expanding users and item profiles through gathering explicit and implicit information [15]. Trust-aware recommenders are one solution for dealing with the cold start problem. Ahmadian et al., (2014) [16] use reliability measurements to improve the accuracy of trust-aware recommender systems and remove people whose predictions are unreliable while preserving good coverage. Another paper presents a variant of the profile expansion technique to alleviate the cold-start problem in recommender systems. For this purpose, the authors consider the user's demographic information (e.g., age, gender, and occupation) and the user's rating information to enrich the neighborhood set. In particular, two distinct strategies are used to embellish the rating profiles of users by adding some additional ratings. The proposed expansion of rating profiles significantly affects the performance of recommender systems, particularly those experiencing a cold start issue [17].

The RSs can automatically learn the user's preferences by analyzing their explicit or implicit feedback. Explicit data might be given by the user in different ways, for instance by requiring them to fill a questionnaire about their preferences and interests. The system can infer implicit interests through the analysis of the user's behavior [2].

The explicit information is often preferred over implicit information because it is more accurate than the predicted or implicit information, i.e., the user can directly enter information about his interest, and then the system will generate accurate recommendations for him [18]. However, receiving explicit information could be challenging for a system. Users might feel uncomfortable providing explicit demographic information, and extremely common questions, such as one's race, gender, income, or age, could cause bias and unfair recommendations [19]. To this end, in this study, we have proposed a method in which the information collected from a new user does not contain demographic information, and the enquired explicit information is data-driven.

In this method, tourism activities are categorized by using exploratory factor analysis (EFA). New users with no rating record of tourism activities are asked to rate each of these categories on a scale of one to five points. The data, rated categories, is then mapped to the activities by a multi-label classification algorithm (MLC), which predicts what activities the user is likely to enjoy; in other words, it will develop a tourism profile for the user. By using the proposed RS and mapping activities to their associated categories, respondents are required to answer fewer questions. The proposed RS can indeed predict tourism activities with fewer data about users. In terms of evaluation, the proposed RS works better than CF-based models.

The rest of the paper is structured as follows: The next section provides an overview of the multi-label classifiers and their applications in RS; the research methodology section deals with how to identify tourism activities, how to extract tourism categories, proposing algorithms to predict the user's favorite activities, and how to evaluate the presented method. In the result section, we review and analyze the method's ability in capturing and predicting user interests. Lastly, in the conclusion section, we review our method and discuss this paper's achievements, research limitations, and future research suggestions.

2. Multi-label Classification and Related Works

In machine learning, single-label classification is one of the commonly used methods in which each instance in the dataset associates with a unique class label from a set of disjoint class labels L. Depending on the number of these classes, the problem can be either a binary classification (when $|L| = 2$) or a multi-class classification (when $|L| > 2$). However, in the multi-labeling problems, each instance can be associated with multiple classes. In such algorithms, the goal is to learn from a set of instances to label each instance's class or classes in L [20]. MLC approaches are categorized into a) problem transformation and b) algorithm adaptation methods.

In problem transformation, the MLC problem transforms into one or more single-label classification problems. Therefore, it does not need any change or adaptation to traditional algorithms, and those algorithms can be applied to the problem [21]. Problem transformation methods are divided into three main algorithms: Binary Relevance (BR), Label Power Set (LP), and Classifier Chain (CC). Using these three problem transformation algorithms, this study applies five classifiers, namely Support Vector Machine (SVM), Decision Tree (DC), Random Forest (RF), Naïve Bays (NB), and K-Nearest Neighbor (KNN). In adaptation algorithms, instead of transforming the problem, the algorithms are changed and modified to handle multi-label data. We used two adaptation algorithms, namely Binary Relevance KNN (BRKNN) and Multilabel K Nearest Neighbors (MLKNN) (Spyromitros et al., 2008; Tsoumakas & Katakis, 2007). Besides these approaches, ensemble learning algorithms can learn from multi-label data natively without any transformation in the base algorithms or the problem. Ensemble methods are learning algorithms that construct a set of classifiers before classifying new data points by taking a (weighted) vote of their predictions [23]. This study used Random Forest (RF) and Extra Tree classifiers (ET) as ensemble algorithm candidates.

MLC has many applications in various domains including text classification [24, 25], image classification [26], bioinformatics [27], genre classification [28], and social media analysis [29]. More details could be found in [30] and [31] publications. Moreover, MLC has leveraged its power into RSs world too. Carrillo et al., (2013) [32] demonstrated the MLC ability to recommend items and deal with RS common problems including data sparsity. Zheng et al., (2014) [33] have used MLC to recommend users' contexts in such a way that instead of recommending the item to the user, the user-related contexts are predicted based on the items selected by the user and the ratings given to each item. To this end, they transformed the problem into an MLC problem and showed that MLC algorithms are more capable of recommending and predicting than the base algorithms. Rivolli et al., (2017) [34] used the MLC algorithms to recommend track foods. They obtained a set of data using a questionnaire comprised of two stages. In the first stage, the user answers 21 questions, which are the attributes describing the user. These questions are viewed as predictive attributes. The second stage of the questionnaire includes 12 food alternatives in which the user is asked to specify their preferences to each of them. These alternatives are associated with classes' labels or target attributes. The results indicate that the adaption algorithm showed weaker performance in comparison to the transformation methods. Elhassan et al., (2018) [35] used MLC to provide remedial actions to address students' shortcomings in Learning Outcome Attainment Rates. In their model, each instance is a student described by a set of characteristics such as field of study, academic level, grades, and so on. Moreover, the related tags for each student are equal to their remedial actions. The results show that the chain classification method with the decision tree classifier gives the best outcome for the given dataset.

However, despite the wide range of studies done about MLC applications in RS, there is still insufficient attention and evaluation of MLC capabilities. One of those capabilities is using MLC to address the cold-start problem and reduce the amount of received explicit information of a new user. This study is an attempt to fill mentioned gaps and show the performance of MLC algorithms in comparison with CF algorithms as base models. To the best of our knowledge, this is the first work in addressing the cold-start problem in developing tourism users' profiles with explicit data-driven and MLC algorithms.

3. Research Methodology

From a data-driven viewpoint, there are two main steps in building the proposed recommendation: the first step is to extract tourism categories by measuring users' interests in tourism activities. The second step is to associate categories and activities with a data-driven connection. To establish such a connection, data collection and training the MLC algorithm are required. Therefore, this connection allows the prediction of the user's interest in activities using his ratings in extracted categories. The tourism sites and activities of Tehran, the capital city of Iran, have been selected as the case study of the presented method. In this section, we respectively address the identification process of tourism activities, the first phase of data collection using the questionnaire, applying factor analysis on the first dataset to extract tourism categories, second phase data collection using the questionnaire, training, reporting the performance of diverse MLC algorithms on the second dataset, and finally in external evaluation comparing our method best result with CF algorithms as a baseline model. The stages of the proposed method are shown graphically in Fig.1.

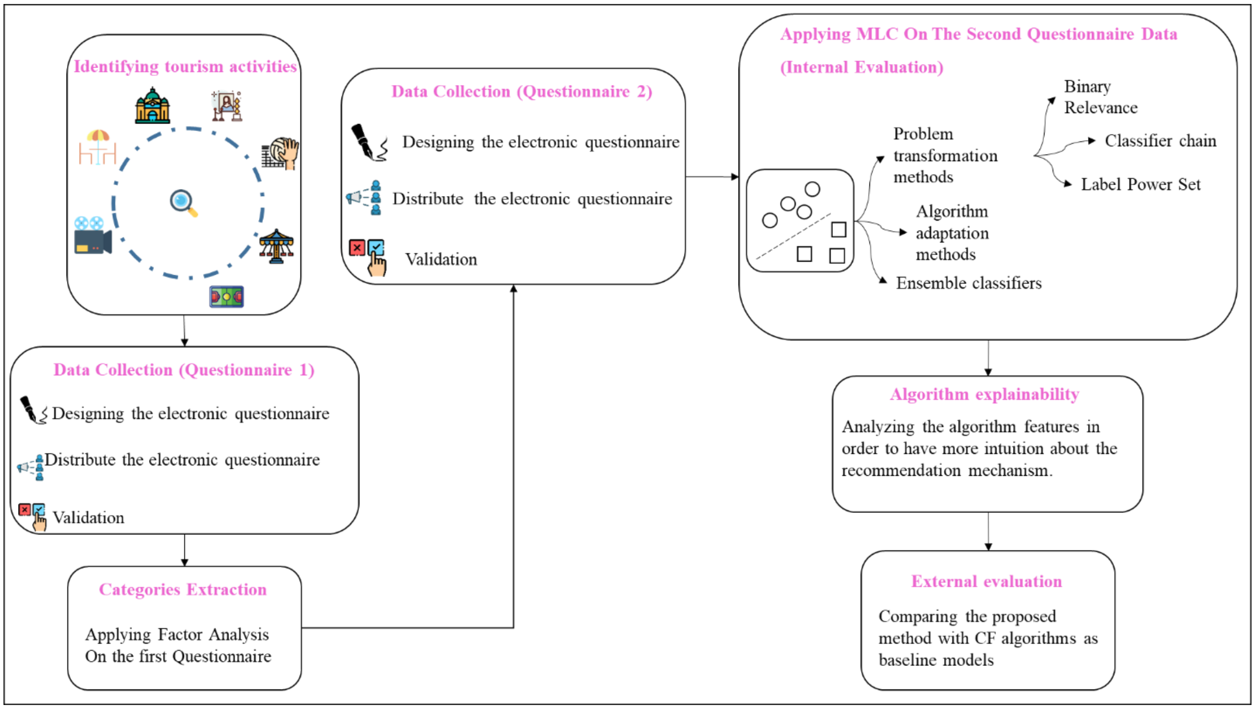
Identifying tourism activities
Data Collection (Questionnaire 2)
Designing the electronic questionnaire
Distribute the electronic questionnaire
Validation
Applying MLC On The Second Questionnaire Data (Internal Evaluation)
Problem transformation methods
Binary Relevance
Classifier chain
Label Power Set
Algorithm adaptation methods
Ensemble classifiers
Data Collection (Questionnaire 1)
Designing the electronic questionnaire
Distribute the electronic questionnaire
Validation
Categories Extraction
Applying Factor Analysis On the first Questionnaire
Algorithm explainability
Analyzing the algorithm features in order to have more intuition about the recommendation mechanism.
External evaluation
Comparing the proposed method with CF algorithms as baseline models

Fig. 1 A schematic representation of the stages in this study

3.1. Identifying Tourism Activities

To identify tourism activities in Tehran, we used previous research [36, 37, 38], analytical reports of the British Tourism Organization[1] and the content of the Tripadvisor[2]. The point to keep in mind was that many of the tourism activities mentioned in the papers and the British Tourism Organization's analytical reports, such as nightclubs or beach tours, do not exist in Tehran. Therefore, by combining and modifying the activities in the mentioned sources, 18 types of tourism-related activities in Tehran were identified, namely going to the cinema, theaters, museums, holy sites, historical sites, sports events, sports activities, art and book exhibitions, music events, malls, public gardens, restaurants, cafe, zoo, rural places, rivers and lakes, and mountains.

3.2. Data Collection (Questionnaire 1)

Reviewing and rating these 18 tourism activities can be a tedious and challenging task for a user. Thus, reducing these 18 activities into fewer and more interpretable categories makes the user more comfortable in recognizing and categorizing the content. In order to generate the category layer, it was necessary to obtain data; therefore, based on the Likert scale (a scale between one and five where one indicates the slightest interest in an activity, and five denotes the most), a questionnaire was designed to measure users' interest in each of the 18 activities. To better guide the users, we introduced several POIs in Tehran as instances for each of the mentioned activities. As an example, Saadabad Palace and Negarestan Mansion were mentioned as instances for the historical sites. After designing the questionnaire, it was distributed randomly on social media platforms, such as Facebook, Twitter, and messenger applications. The total number of 272 questionnaires were collected, and the reliability of the designed questionnaire was proved by calculating Cronbach's alpha equal to 0.846, which was more than the cutoff required of 0.7 [39]. A schema of the data collected by the questionnaire is shown in Table.1. Due to page size limitations, only a few of the activities and scores were given to each activity by participants in the questionnaire are shown.

Table 1 A sample of the data collected by the first questionnaire

Activities Index	Palace	Museum	Mansion	Cinema	Theater	Holy Sites	Mall	Public Garden	Restaurant	Cafe	Zoo	Rural Places	Lakes	…	Mountains
1	5	5	5	3	4	4	2	1	1	3	2	4	4	…	4
2	4	2	3	4	5	3	4	4	5	5	3	5	5		4
3	4	4	4	3	4	3	4	4	4	3	3	4	4		4
4	5	5	2	5	5	1	4	4	5	5	5	4	2		2

[1] www.visitbritain.org/archive-great-britain-tourism-survey-overnight-data

[2] www.tripadvisor.com/Attractions-g293998-Contexts-Iran.html

5	4	3	4	3	2	3	1	4	2	4	2	3	3		3
…	…														3
272	2	2	1	4	3	4	5	3	5	5	2	1	4		4

3.3. Extracting Tourism Categories (Factor Analysis of Questionnaire 1)

Factor analysis (FA) primary goal is to summarize data for revealing relationships and patterns by regrouping variables into a limited collection of clusters based on shared variance. FA utilizes mathematical methods to simply interrelate measures for discovering patterns in a set of variables. FA was applied in various types of fields, such as behavioral and social sciences, medicine, economics, and geography [40], and is divided into two main classes, namely exploratory factor analysis (EFA) and confirmatory factor analysis (CFA). EFA is used when the research goal is to discover the number of influencing variables or to find variables that go together. FA is useful for studies such as questionnaires based on a few to hundreds of variables which can be reduced to a smaller set to simplify interpretations. Therefore, not only is focusing on a smaller set of variables easier than considering too many keys but also, by clustering them into some categories, it makes variables meaningful. In this paper, EFA was applied for accessing meaningful categories of variables. The determinant score for our data is 0.0000135, which is more than 0.00001, and indicates a violation in the assumption of correlation of variables; in such a case, to extract the factors, it is recommended to use Principal Axis Factor [40]. We used the Varimax rotation method with 30 iterations based on the default value in SPSS software for rotation. To check the adequacy and suitability of the dataset for EFA, Kaiser-Meyer-Olkin measure (KMO) and Bartlett's test of sphericity were applied. The minimum value of the KMO index for the factor analysis is 0.5, which in our research is 0.76. The Bartlett test takes a statistical hypothesis, and its null hypothesis states that the correlation matrix is an identity matrix, so there is no significant relationship between the variables. As can be seen in Table.2, the p-value is not in the rejection area (the value of sig must be less than 0.05, which is zero for our data).

Table 2 KMO and Bartlett’s Test

Kaiser-Meyer-Olkin Measure of Sampling Adequacy		0.76
Bartlett’s Test of Sphericity	Approx. Chi-Square	1833.255
	Df	325
	Sig	0

To determine the number of significant factors, Kaiser's criteria states that only factors with Eigenvalues of one or more should retain. According to Fig.2 (Scree Plot), the best number of factors after rotation for this dataset is 7.

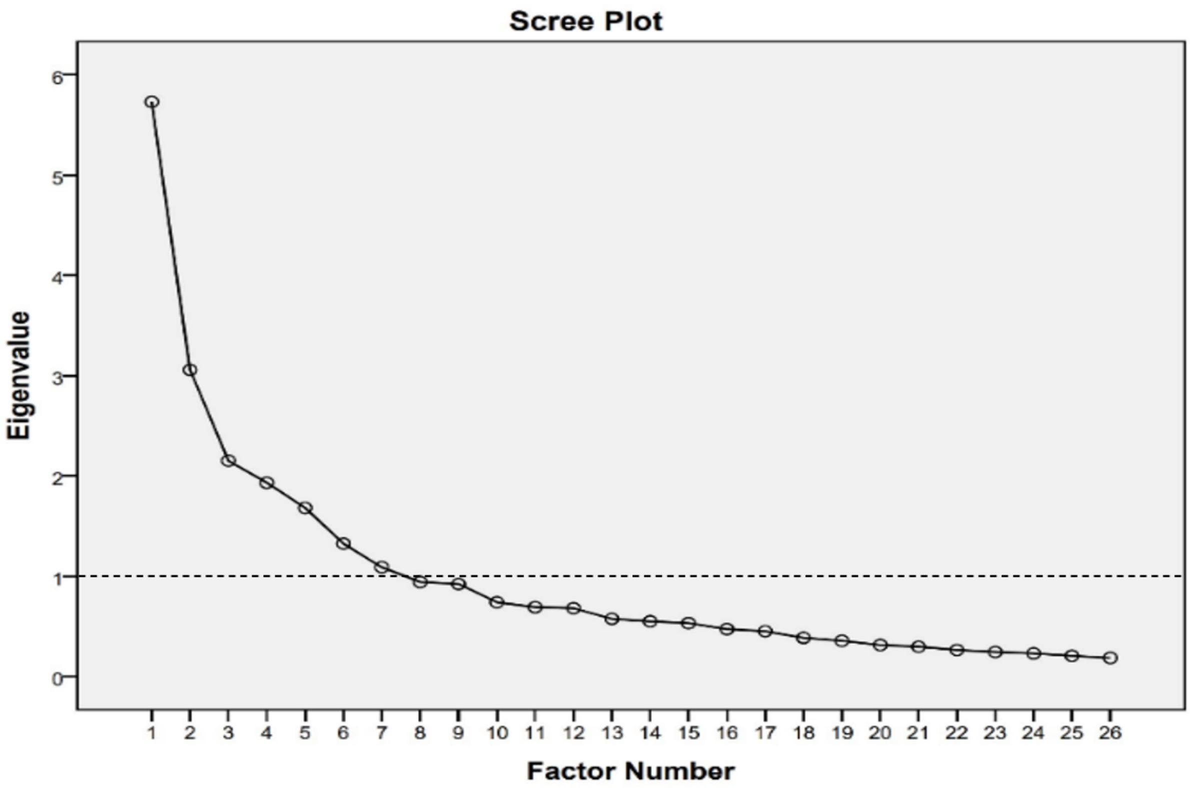

Fig. 2 Scree Plot for determining the optimum number of factors, factors below eigenvalue

As factor naming does not follow a specific rule, here we named each factor based on the associated variables that describe the factor (Table.3).

Table 3 Extracted factors name and their associates' variables

Factor	Descriptor Variables
Historical	Museums, Historical Sites
Fun	Restaurants, Cafe, Male
Ecotourist	Rivers and Lakes, Mountains, Rural Places
Sportive	Sport Activities, Sport Events
Cultural	Music Events, Art and Book Exhibitions, Cinema, Theaters
Religious	Mosques and Churches, Holy Sites
Urban-related	Zoo, Public Gardens, Parks

3.4. Data Collection (Questionnaire 2)

To train MLC algorithms, we need data to map the connection between categories and activities. The advantage of this connection is that different states can be considered, and the interest of the new user in tourism activities can be predicted simply based on rating the seven categories extracted by factor analysis

(Fig.3). Therefore, a dichotomous questionnaire was designed. The first part asked the users to determine their interest rate for every category on a five-point Likert scale. The second part asked participants to indicate their fondness for each of the 18 activities using binary values of 0 for not being interested and 1 for being interested. We randomly distributed this questionnaire via social media and messaging applications to Tehran residents. In total, 578 questionnaires were collected, and the calculated Cronbach's alpha was 0.859.

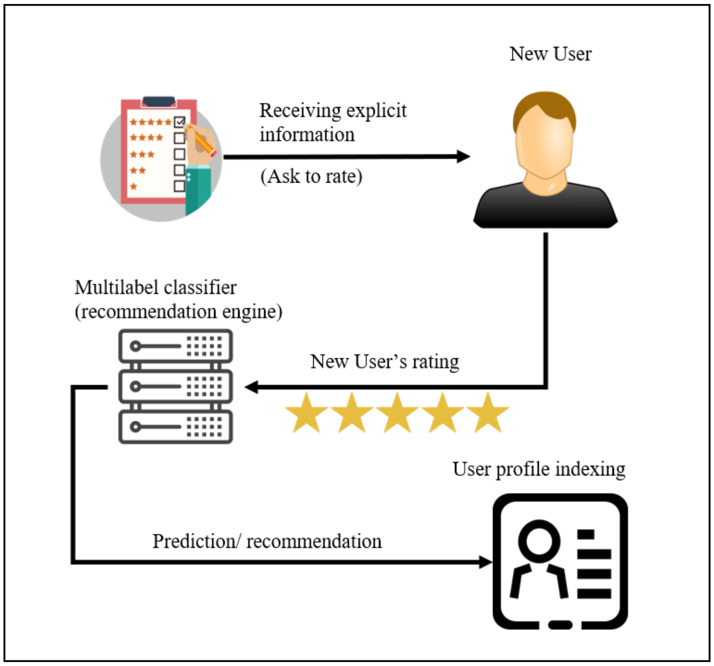
New User
Receiving explicit information
(Ask to rate)
Multilabel classifier (recommendation engine)
New User's rating
User profile indexing
Prediction/ recommendation

Fig. 3 The schema of proposed method for developing a new user's profile

Table.4 shows a view of the data gathered from the second questionnaire. Because of space limitations, only a few categories, activities, and scores are displayed on the page.

Table 4 An outline of the data collected by the second questionnaire

User ID	Categories				Activities						
	Historical	Fun	…	Urban-related	Palace	Museum	Mansion	Cinema	Theater	…	Mountains
1	5	1		1	0	0	1	1	0		0
2	2	3		4	0	0	1	0	1		1
3	3	3		3	1	1	0	0	0		0
4	5	3		4	1	0	1	0	0		0
4	2	2	…	2	0	0	0	0	1	…	1
6	3	3		3	1	0	0	1	1		0
7	1	5		1	0	1	1	1	1		1

…			…							…	
578	5	3	4	3	1	1	1	0	1	0	0

3.5. Multi-Label Classification Problem Definition

Let X be the Users-Category matrix, and $\mathrm{L} = \{\lambda_1, \lambda_2, \dots, \lambda_k\}$ be a finite set of labels or activities. A user $x \in X$, represented in terms of features vector $x = (x_1, x_2, \dots, x_m)$, which these features vector is referred to the given rates of a user to each of extracted categories; therefore, the user x is associated with a subset of labels $l \in 2^{\mathrm{L}}$. Notice that if we call this set L be the set of relevant labels of x, then we could call the complement $\mathrm{L} \backslash l$ to be the set of irrelevant labels of x. Let denote the set of relevant labels l with a binary vector $Y = (y_1, y_2, \dots, y_k)$, where $y_i = 1 \Leftrightarrow \lambda_i \in L. \mathcal{Y} = \{0,1\}^k$ is the set of all such possible labeling. Therefore:

Given a training set, $S = (x_i, Y_i), 1 \leq i \leq n$, consisting n training instances, $(x_i \in X, Y_i \in \mathcal{Y})$ $i.i.d$[3] drawn from an unknown distribution D, the goal of the multi-label learning is to produce a multi-label classifier $h: X \rightarrow \mathcal{Y}$(in other words, $h: X \rightarrow 2^L$) that optimizes some specific evaluation function (i.e., loss function) [20].

This study uses the second questionnaire data as a training data set for MLC algorithms. Consequently, when a new user is entered into the system, by rating each of the categories from 1 to 5, his/her interest in the 18 activities will be predicted. In the transformation approach, all three algorithms (BR, LP, CC) with LR, DT, RF, SVM, KN classifiers are used. For adaptation algorithms, BRKNN and MLKNN, and for ensemble algorithms, ET and RF classifiers are utilized. We implement MLC algorithms using Python version 3.5 with the scikit-learn and scikit-multilearn packages. All the classifiers' hyperparameters in this study are the package's default values.

It is important to consider imbalanced labels' issues, as shown in Fig.4. The number of classes is not equal in any of the labels, which may cause problems in some algorithms' learning processes. To solve this problem, we used the scikit learn package class weight balancing feature in all classifiers except the NB, MLKNN, and BRKNN; because those cannot benefit from this technique.

3.6. Evaluation

In this research, the evaluation stage is vital in two ways. First, in internal evaluation, in response to the first research question, we measured and compared the performance of different MLC algorithms in capturing and predicting users' interests. Second, the proposed method performance was compared to CF algorithms according to state-of-the-art to address the second research question. 5-fold cross-validation with three metrics was used for this. Cross-validation can train algorithms with little data.

[3] independent and identically distributed

Moreover, since all samples are used both as training and test data in the algorithm learning process, it is favorable to compare different algorithms' performance on classification problems. There are a few different metrics for the MLC evaluation, but it is necessary to select those that can be used for CF methods. One of our goals during the evaluation stage is to compare the proposed framework with state-of-the-art CF techniques. To assess this, we selected Precision, Recall, and F1-Score metrics. Since the weight-balancing technique is not applicable for some classifiers (NB, MLKNN, BRKNN), we used the macro-average criteria that do not take label imbalance into account.

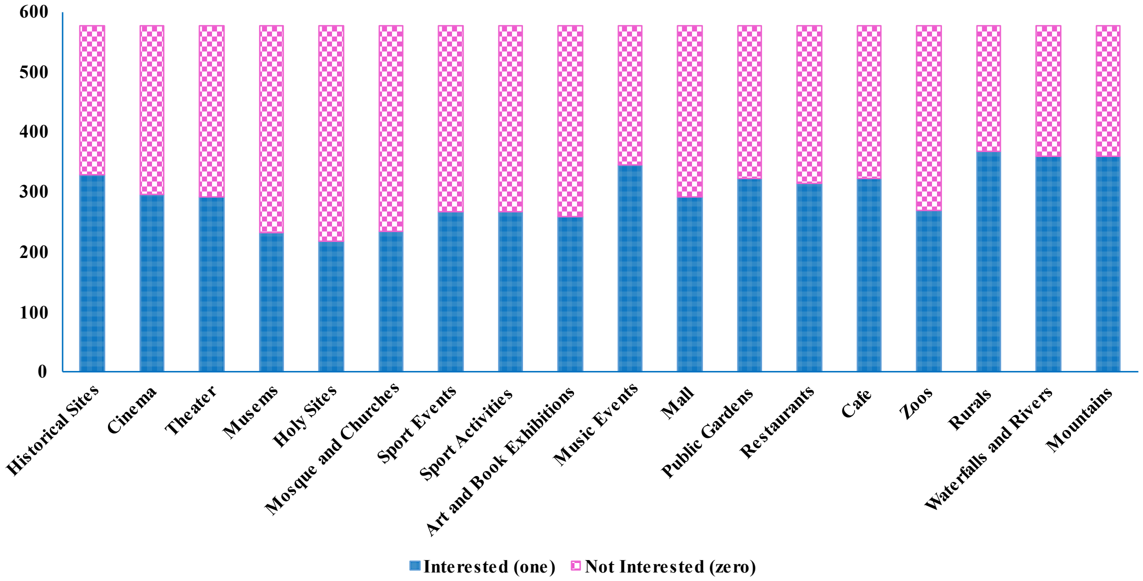

Fig. 4 Labels imbalance, vertical pivot represents the number of samples, and the horizontal pivot shows the label names

3.6.1. Metrics

Let T be a multi-label dataset consisting n multi-label examples (x_i, Y_i), $1 \le i \le n$, $\big(x_i \in \mathrm{X}, Y_i \in \mathcal{Y} = \{0,1\})^{k}\big)$, with a label set L, $|\mathrm{L}| = k$. Let h be a multi-label classifier and $Z_i = h(x_i) = \{0,1\}^k$ be the set of label memberships predicted by h for the example x_i. Therefore:

Precision (p): Precision is the proportion of predicted correct labels to the total number of actual labels, averaged over all instances. In our case, Precision indicates how much of the predicted activities are correct for the user [41][20].

$$Precision, P = \frac{1}{n}\sum_{i}^{n}\frac{|Y_i \cap Z_i|}{|Z_i|}$$

Recall: Recall is the proportion of predicted correct labels to the total number of predicted labels, averaged over all instances. In our case, Recall indicates how much the algorithm has been able to predict the user's favorite activities [41][20].

$$Recall, R = \frac{1}{n}\sum_{i}^{n}\frac{|Y_i \cap Z_i|}{|Y_i|}$$

F1-Score: Definition for Precision and Recall naturally leads to the following definition for F1-score [41][20].

$$F1 = \frac{1}{n}\sum_{i}^{n}\frac{2 * |Y_i \cap Z_i|}{|Y_i| + |Z_i|}$$

3.6.2. The Proposed Method versus Baseline Model

In this part, the framework presented should be compared to CF algorithms, which predict a user's interest in an activity by observing his interactions with other activities. The problem should be transformed into a CF problem to evaluate the CF prediction ability with our proposed method. We defined a scenario in which users are given a set of activities at random and asked to indicate their interest in each one in a binary manner. As a next step, the user's scoring record is fed into the CF algorithm, which predicts his other interests based on his scoring record. This step was made possible by using the second questionnaire data. As for the CF problem, we did away with users' ratings of categories and only retained users' binary ratings of activities. Finally, our method's best score was compared to that of two state-of-the-art CF algorithms.

CF systems generally have memory-based and model-based techniques for the recommendation. There are no assumptions on data in the memory-based technique, and it essentially depends on the nearest neighbors' search to find the closest pairs of items or users. When the recommendation is based on measuring the similarities between items, it is an item-item method, and when it is based on measuring users' similarities, it is a user-user method. We focus on the user-user method for our problem because the number of items (activities) is few. In the case of few items, the variance of the item-item method is low, and its bias is high; thus, personalization may suffer. On the other hand, model-based techniques rely upon assumptions made about the data and build a model that explains the interactions between users and items.

To formulate our problem for CF systems, we only need the user activity matrix (users' interest in activities) without user-category (their rating to each category). Thus, let we have N users and K activities, define the user-activity matrix $A \in \{0,1\}^{N\times K}$:

$$A_{ui} = \begin{cases} r_{ui}, user\ u\ interested\ on\ activity\ i & if\ such\ rating\ exicit \\ ? & if\ no\ such\ rating \end{cases}$$

CF systems try to replace all the "question marks" in A by some optimal guesses; the goal is to minimize the RMSE (root mean square error) when predicting the user interests on a test set (which is, of course, unknown during the training phase), that is to minimize:

$$rmse = \sqrt{\frac{1}{|S_{test}|} \sum_{(i,u)\in S_{test}} (r_{ui} - \hat{r}_{ui})^2}$$

Where $(u,i) \in S$ test if User u interest activity i in the test set, $|S_{test}|$ is its cardinality, r_{ui} is the true rating, and $\hat{r}_{ui}$ is the prediction based on the recommendation system [42].

Following is a description of six CF algorithms that are appropriate for our data and type of problem compared with the method presented in this study. In this study, we intend to compare a variety of algorithms of varying capabilities with the presented method. The two first algorithms are basic, and they do not do much work, but they are still appropriate for comparing performance. The two other algorithms relate to model-based and memory-based techniques. The last two algorithms are two popular and powerful models based on neural networks.

Random Predictor (RP) Predicts the rating of the training set based on its distribution, which is assumed to be normal. $\hat{r}_{ui}$ is the prediction resulting from a normal distribution N ($\hat{\mu}$,$\hat{\delta}^2$), where a maximum likelihood estimator is used to estimate $\hat{\mu}$ and $\hat{\delta}$ from training data.

$$\hat{\mu} = \frac{\mathbf{1}}{|\boldsymbol{R_{train}}|} \sum_{\boldsymbol{r_{ui}} \in \boldsymbol{R_{train}}} \boldsymbol{r_{ui}}$$

$$\hat{\delta} = \sqrt{\sum_{\boldsymbol{r_{ui}} \in \boldsymbol{R_{train}}} \frac{(\boldsymbol{r_{ui}} - \mu)^{\mathbf{2}}}{|\boldsymbol{R_{train}}|}}$$

This model aims at providing a basis for comparisons between different models and random prediction.

BaselineOnly predicts the baseline estimate for given user and item.

$$\hat{r}_{ui} = b_{ui} = \mu + b_u + b_i$$

It is assumed that bias b_u is zero if user u is unknown. This is also true for item i with b_i.

In a **memory-based approach**, we choose BSKNN that taking into account a baseline rating; A baseline estimate for an unknown rating r_{ui} is denoted by b_{ui} and accounts for the user and item effects:

$$b_{ui} = \mu + b_i + b_u$$

The parameters b_u and b_i indicate the observed deviations of user u and activity i, respectively, from the average, μ denotes the overall average rating, and λ_1 the regularization term. To predict $b_{u,i}$ that is to minimize the problem:

$$\min \Big(\sum_{(u,i)\epsilon \text{ꝃ}} (r_{ui} - u - b_u - b_i)^2 + \lambda_1 \big(\sum_u b_u^2 + \sum_i b_i^2\big)\Big)$$

Therefore, the prediction $\hat{r}_{ui}$ is set as:

$$\hat{r}_{ui} = \mu_u + \frac{\sum_{v \epsilon N_i^k(u)} sim(u,v).(r_{vi} - \mu_v)}{\sum_{v \epsilon N_i^k(u)} sim(u,v)}$$

Where $sim(u,v)$ denotes, similarity measurement between user u and v, and $N_i^k(i)$ only include neighbors for which the similarity measure is positive [43].

The prediction $\hat{r}_{ui}$ is set as:

$$\hat{r}_{ui} = \mu + b_u + b_i + q_i^T p_u$$

If user u is unknown, then the bias b_u and the factors p_u are assumed to be zero. The same applies for item i with b_i and q_i.

To estimate the unknown parameters, minimize the problem:

$$\sum_{r_{ui} \in R_{train}} (r_{ui} - \hat{r}_{ui})^2 + \lambda(b_i^2 + b_i^2 + \|q_i\|^2 + \|p_u\|^2)$$

For more details, see Surprise package's document and Koren, (2010).

Neural Matrix Factorization (NeuMF) [44] takes advantage of the flexibility and non-linearity of neural networks to replace the dot products in matrix factorization in order to improve the model's expressiveness. To formularize NeuMF:

$$\hat{r}_{ui} = f\left(\mathbf{P}^T \mathbf{v}_u^U, \mathbf{Q}^T \mathbf{v}_i^I \mid \mathbf{P}, \mathbf{Q}, \Theta_f\right)$$

where $\mathbf{P} \in \mathbb{R}^{M \times K}$ and $\mathbf{Q} \in \mathbb{R}^{N \times K}$, denoting the latent factor matrix for users and items, respectively; M and N denote the number of users and items; $\mathbf{v}_u^U$ and $\mathbf{v}_i^I$ are feature vectors that describe user u and item i; and Θ_f denotes the model parameters of the interaction function f. Since the function f is defined as a multi-layer neural network, it can be formulated as

$$f\left(\mathbf{P}^T \mathbf{v}_u^U, \mathbf{Q}^T \mathbf{v}_i^I\right) = \phi_{\text{out}}\left(\phi_X\left(\dots \phi_2\left(\phi_1\left(\mathbf{P}^T \mathbf{v}_u^U, \mathbf{Q}^T \mathbf{v}_i^I\right)\right) \dots\right)\right),$$

where ϕ_{out} and ϕ_x respectively denote the mapping function for the output layer and x-th neural collaborative filtering (CF) layer, and there are X neural CF layers in total.

Standard Variational Auto-Encoder (VAE) [45] The Standard-VAE considered in this paper takes user rating $x_u$as input. Through the encoder function $g_\phi()$, the user input is encoded to learn the mean, m_u, and standard deviations σ_u of the K-dimensional latent representation. The latent vector for each user, z_u is sampled using m_u, σ_u. The decoder function $f_\theta()$ is then used to transfer the latent vector from K-dimensional space to a probability distribution π_u in the original N-dimensional space. This distribution indicates the probability that each of the N-activities will be liked by user u.

$$g_\phi(x_u) = m_u, \sigma_u \; z_u \sim N(m_u, \sigma_u)$$

$$f_\theta(z_u) = \pi_u$$

The output is a probability distribution over the K items. In this model, ELBO is used as the objective function/loss.

$$\text{loss} \;= \log\, p_\theta(x_m \mid z_m) + KL\big(q(z_m) \parallel p(z_m \mid x_m)\big)$$

Where, x_m is the activity feature vector while z_m is its latent representation. Here, the first part of the equation considers the log-likelihood for an activity given its latent representation and the second part is the Kullback-Leibler (KL) divergence measure. The log-likelihood function considered is given as,

$$\log\, p_\theta(x_u \mid z_u) = \sum_i x_{ui} \log\, \sigma(f_{ui}) + (1 - x_{ui}) \log\, \big(1 - \sigma(f_{ui})\big)$$

where $\sigma(x) = 1/(1 + \exp\,(-x))$ taken over all the items i. The KL Divergence is calculated for the latent state of the model, z_u.

Another problem is left, the output of the mentioned algorithm is in the range [0,1], but the desired output should be binary, which one and zero respectively denotes a user interest and dislike of an activity. To solve this problem, we suggest using a threshold α where:

$$output = \begin{cases} 1, & \hat{r}_{ui} > \alpha \\ 0, & else \end{cases}$$

For both selected algorithms, to choose the best threshold value, the F1-score is calculated for different α values, the α value having the best F1-score is chosen as the threshold.

4. Result and Discussion

This section includes two parts of evaluation; in the first part, MLC algorithms' results is reviewed, and in the next part, the results of the best MLC algorithm is compared with the CF problem-solving approach.

4.1. MLC Results (Internal Evaluation)

The performance of MLC algorithms is shown in table 5. The three columns represent the Precision, Recall, and F-1 score; the higher the value, the better the result. For MLC transformation algorithms, we used "Algorithm-Classifier" to denote the algorithm, e.g., BR-NB denotes the use of binary relevance (BR) as a transformation algorithm and Naïve Bayes as a classifier. Moreover, for Ensemble methods, we used the "Ensemble-Classifier" form. The results demonstrate that the proposed method can capture and predict users' interests with just a few explicit inputs.

Table 5 Comparison of the MLC algorithms' performance

MLC algorithms	Precision	Recall	F1-Score
BR-RF	0.748	0.722	0.725

BR-SVM	0.726	0.715	0.712
BR-LR	0.733	0.721	0.718
BR-NB	0.75	0.74	0.733
BR-DT	0.657	0.647	0.647
CC-RF	0.727	0.633	0.658
CC-SVM	0.724	0.692	0.699
CC-LR	0.718	0.694	0.692
CC-NB	0.677	0.663	0.656
CC-DT	0.657	0.673	0.66
LP-RF	0.62	0.615	0.61
LP-SVM	0.657	0.659	0.652
LP-LR	0.648	0.601	0.616
LP-NB	0.629	0.654	0.594
LP-DT	0.641	0.655	0.643
BRkNNa	0.731	0.666	0.674
MLkNN	0.708	0.693	0.683
Enssemble-ET	0.697	0.698	0.69
Enssemble-RF	0.715	0.699	0.696

According to the metrics results, the best performance is related to BR algorithms, while the LP algorithm shows lower metrics values than others. The precision score and the Recall closeness in most classifiers indicate that the class imbalance has not affected the learning process. Since there are 18 tourism activities, BR algorithms transform the problem into 18 separate problems regardless of the interdependence of labels. This algorithm's satisfactory performance indicates that the detected activities are distinctive from each other, and the algorithm has been able to map the category layer to the activity layer space well. For example, the RF classifier with the BR algorithm showed better results than other algorithms with the RF classifier.

Nevertheless, the LP algorithms' disappointing results are due to their problem-solving approach. LP converts the MLC problem into a multi-class classification problem with 2^L possible class values. Since the dataset used in this study has few records and many labels, it is not easy to train such algorithms on such a data set. Nevertheless, algorithms' outcomes and superiority may change as the number of data increases. Recall's high score refers to the classifier's success rate in identifying and proposing the user's favorite activities. The higher this score is, the more the recommender has recommended the user's favorite tourism activities. On the other hand, Precision indicates what percentages of the activity recommended to the user were actually the user's favorite activities. Of course, sometimes, a precision error can also be welcome, i.e., when an activity outside of the user's favorite activities is recommended, and the user feedback to that is positive; therefore, it can be seen as a way to prevent over-personalization. If a recommender offers all the activities to the user, its recall score will be 100%, but its precision score will be low. Also, if it tries to suggest a smaller number of activities to the user, its Recall will be low, and its precision score will be high. In such cases, the F1-score, a harmonic average of the two mentioned metrics, can be a valuable criterion for comparing classifiers performance. To select the best algorithm based on the F-score results, BR-NB has the best performance among the categories. Its Precision, Recall, and F1 scores are 0.75, 0.74, and 0.733, respectively. The evaluation results of the two adaptive algorithms, BRKNN and MLKNN, were similar and did not have a significant advantage over each other, as are the ensemble algorithms.

4.1.1. Feature Importance Explanation

As an aid to understanding how MLC predicts users' interest in particular activities, Table 6 sets out a breakdown of the features' importance for each activity. As our best algorithm, Naive Bayes, cannot handle feature importance analysis, we have chosen the BR-RF algorithm. Even though the results in table 4 seem pretty straightforward, our case study concerns Tehran, so some of the feature importance analysis results need to be clarified.

Cultural and fun features play an important role in theater and cinema, but their significance varies slightly. In comparison with cinema, the fun aspect of theater has decreased in importance, while the cultural aspect has increased. In fact, theater is a more cultural experience for individuals than films.

In visiting museums activity, aside from cultural and historical features, the religious feature is also very effective because of the many religious museums in Tehran, such as the Quran Museum.

For activities such as visiting public gardens and parks, in addition to urban-related features, cultural and historical features are also important. Some public gardens in Tehran have classical architecture buildings, which give the gardens a cultural and historical significance, e.g., Negarestan garden.

The urban-related feature is the second most effective for all activities such as visiting rural areas, mountaineering, lakes, and rivers. Many tours and tourism categories introduce these activities as weekend tourism activities. As Tehran has easy access to the mentioned points of interest and many of them are within walking distance of residential areas, it is understandable for people to have a similar urban view of such activities.

Table 6 Feature importance analysis of activities

	Historical	Fun	Ecotourist	Sportive	Cultural	Religious	Urban-related
Historical Sites	0.411	0.084	0.074	0.050	0.199	0.093	0.089
Cinema	0.044	0.211	0.042	0.084	0.420	0.138	0.061
Theater	0.083	0.128	0.099	0.118	0.468	0.064	0.042
Museums	0.228	0.071	0.085	0.071	0.363	0.141	0.041
Holy Sites	0.085	0.088	0.028	0.132	0.040	0.465	0.162
Mosques and Churches	0.284	0.052	0.140	0.033	0.090	0.333	0.068

Sport Events	0.054	0.039	0.094	0.544	0.086	0.075	0.108
Sport Activities	0.064	0.096	0.100	0.541	0.043	0.079	0.077
Exhibitions (Art and Book)	0.116	0.124	0.044	0.104	0.432	0.132	0.048
Music Events	0.088	0.318	0.035	0.080	0.300	0.055	0.124
Mall	0.042	0.543	0.035	0.042	0.102	0.127	0.109
Public Gardens and Parks	0.106	0.097	0.214	0.032	0.154	0.063	0.334
Restaurants	0.043	0.560	0.070	0.038	0.088	0.137	0.065
Café	0.041	0.552	0.032	0.074	0.160	0.085	0.056
Zoo	0.042	0.162	0.074	0.115	0.081	0.229	0.297
Rural Places	0.155	0.060	0.414	0.046	0.071	0.079	0.175
Lakes and Rivers	0.057	0.076	0.366	0.080	0.055	0.068	0.297
Mountains	0.114	0.080	0.396	0.124	0.092	0.061	0.132

4.2. BR-NB VS BSKNN and SVD (External Evaluation)

We utilized 5-fold cross-validation to evaluate the CF base models. For the adjustment of the hyperparameters, we used the different values examined in other studies and compared the results by repeating the experiment. Finally, we report the highest result for each algorithm as well as the hyperparameters associated with that result. In Table 7, the values of hyperparameters for each model are listed.

Table 7 CF base models and their associated hyperparameter values.

Model Name	Hyper Parameters
RP	No hyper parameters
BaselineOnly	number of epochs = 20, regularization = 0.02, user regularization = 15, item regularization = 10, learning rate= 0.05, optimizer = Stochastic Gradient Descent)
BSKNN	max number of neighbors = 40, max number of neighbors = 1, similarity = user base
SVD	number of epochs = 20, number of factors =20, batch size =64, regularization = 0.02, learning rate=0.007, optimizer = Stochastic Gradient Descent)
NeuMF	number of epochs = 30, number of factors =10, hidden layers = [10,10], batch size =64, regularization = 0.02, learning rate=0.002, weight decay=1e-5, optimizer = Adam,)
VAE	number of epochs = 300, hidden dimension=12, latent dimension=6, batch size =32, regularization = 0.02, learning rate= 1e-3, optimizer = Adam)

Since the CF base algorithms' output falls between 0 and 1, a threshold was used to convert it to a binary output. Fig.5 shows each of the algorithms' F1-Score for different α values. As expected from the performance of these algorithms, with the increase of α, the process of recommending activities to the user becomes more rigorous, consequently, the amount of recall score decreases, and the score of precision increases. Thus, using the F1 score as a harmonic average of precision and recall could provide a reasonable basis for determining a threshold value.

According to Fig.5, the F1 score for both the SVD and the BSKNN lies in the range of 0.45 to 0.55, with the difference of the F1 within this range being negligible for both. Accordingly, a value of 0.5 was determined for both algorithms. The basic models, RP and BaselineOnly, behave differently. The RP predicts labels randomly based on an assumed normal distribution over data. The changes in RP performance until 0.5 are not noticeable, but after this threshold, the changes are more pronounced; 0.5 is an appropriate value for its threshold. A threshold value of 0.55 is considered appropriate for the BaselineOnly model. A positive aspect of this model is that the gap between precision and recall is very small, regardless of the threshold value. The VAE model is sensitive to threshold changes, and the gap between precision and recall is impressive. F-score performance is almost indistinguishable up to the threshold point of 0.4; however, above this point, there is a marked decrease in performance. Based on our observations during the experiment to adjust the hyperparameters, we found that with more training, this model tends to reduce output values to near zero. By increasing the threshold, the false negatives also increase, and hence there is a decrease in the recall, leading to a lower F-score. In this study, 0.4 represents the ideal threshold value.0.4. NeuMF exhibits similar behavior to VAE in the F-score with threshold changes. In adjusting the hyperparameters of this model, we observed that the outputs of this model are more stable. In addition, the gap between precision and recall was smaller than in the VAE case. Figure 5 shows that 0.45 is the optimal threshold for this model.

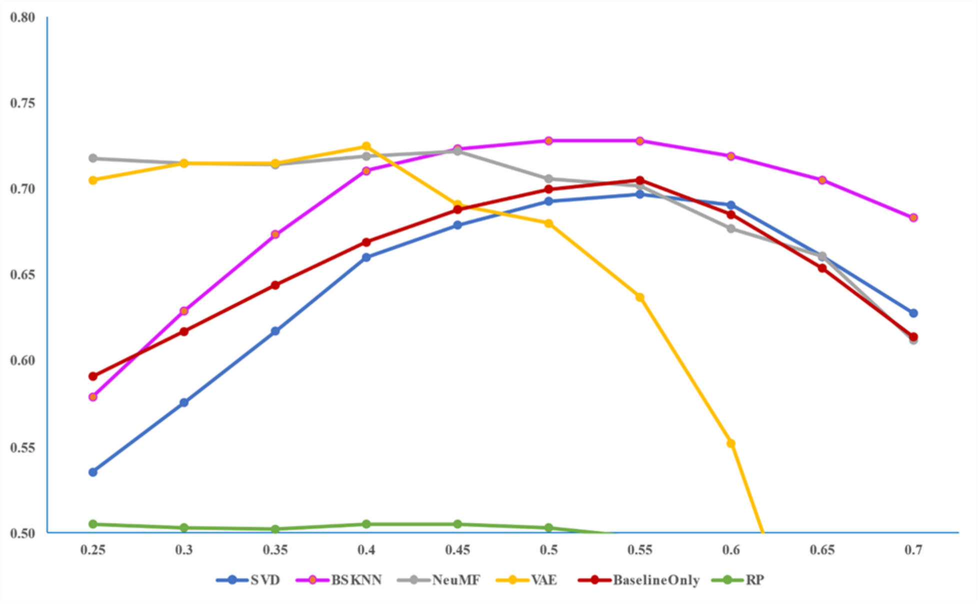

Fig. 5 BSKNN and SVD F1-Scores for different α values

To compare the proposed method with the selected CF models, we selected the BR-NB classifier, which has the superior performance among the MLC algorithms. Fig.6 Shows the Precision, Recall, and F1 scores for each of them. The weakest results are obtained for the SVD model, which is even worse than the baseline-only model, but still performs much better than random prediction. It appears that the linear model of this algorithm does not adequately describe the data collected by this researcher. NeuMF was proposed in order to overcome the shortcomings of linearity in SVD, and the results of this model demonstrate how non-linear modeling of NeuMF yields superior results to SVD. Out of the two neural network-based models proposed in this study, NeuMF displays more stable results than VAE, while also having a lower gap between precision and recall. As we explained earlier, we have observed that, during the training of neural networks-based models on our data, they tend to reduce their outputs that are between 1 and 0 to as close to 0 as possible. Consequently, both models have a high recall for thresholds below 0.5, which contributes to their F-score. While the recall is highest in the VAE model, on the other hand, its performance in precision is quite poor. However, the conditions for the NeuMF model are more favorable. Among the CF-based models, the BSKNN model produces relatively better results and the highest F-score. Moreover, the difference between its precision and recall is negligible. In our opinion, the reason for the good performance of this model may be associated with the type of data and the manner in which it was collected. Memory-based models are in general very sensitive to outliers; however, the data that we collected via the questionnaire allowed us to avoid the occurrence of any outlier during the data collection phase. The proposed method with Br-NB classifier outperformed all other CF-based models. Even so, it is pertinent to note that in the CF system, only a single step of data collection is needed, not the additional step for tourism category extraction. Yet, the advantage of our method lies in the fact that the user interacts with only seven categories; in other words, we have reduced the problem dimensions by mapping the user's input to activities.

Another critical point is that we also evaluated CF models with the 5-fold cross-validation method. The total number of activities is 18. In each step of cross-validation, 4-fold (for each user, nearly 15 given rate records) is considered as training data and one-fold as test data (for each user, about three given rate records). This means that the CF models' results are based on having about 15 records of user interaction with tourism activities and predicting user interests to nearly three activities. However, in the proposed method, the user interests in all 18 tourism activities are predicted by rating [1,5] scale (explicit data) to each of the seven tourism categories without having any interaction records.

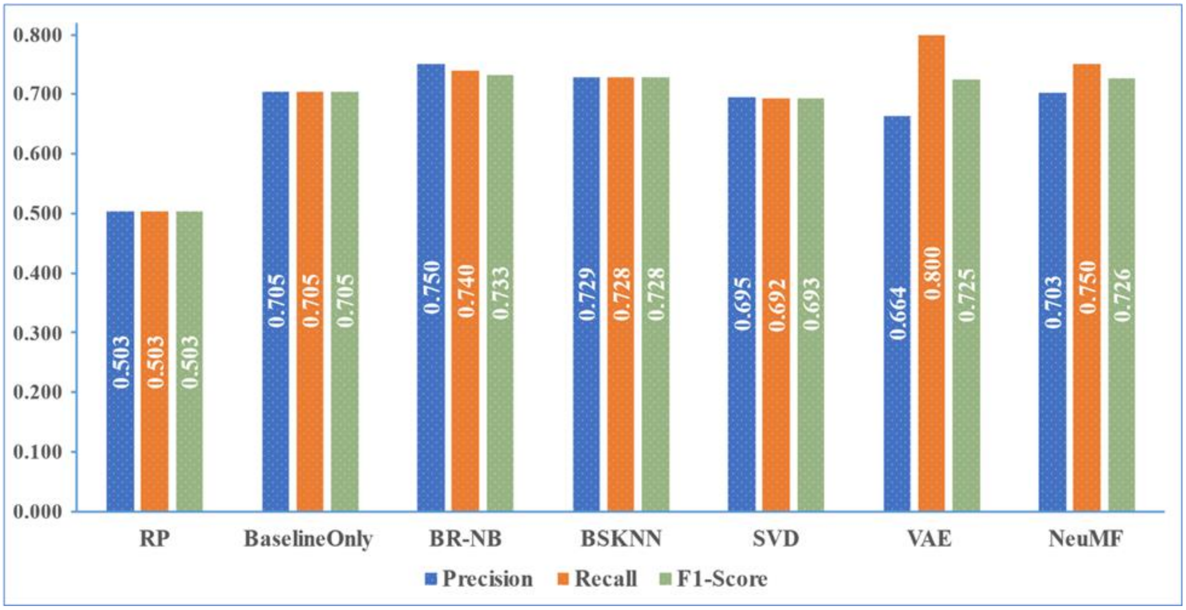

Fig. 6 Comparison of BR-NB results in the proposed method with two state-of-the-art algorithms, BSKNN and SVD, in CF systems

5. Conclusion and Future Research

In this study, we present a method of receiving explicit information that addresses the cold-start problem of tourism users without including sensitive or demographic information, and the enquired explicit information is data-driven. For this purpose, several tourism activities were identified in the city of Tehran. Then tourism categories were obtained by applying FA to questionnaire data that measured users' interests in the identified tourism activities. Accordingly, an MLC problem was formulated, in which new users' ratings of each category represented explicit input that is mapped to identified activities. This decoding process by MLC predicts whether the user is interested in an activity or not. We used a second questionnaire to collect the required data for training and testing the MLC algorithms. In the internal evaluation phase, we compared the results of different MLC algorithms, and the BR-NB results performed best compared to the other classifiers. According to the literature review and the performance outcome of different MLC algorithms in this study, they can have different performances given to the problem and data. In other words, there is not any definite superiority for any of the algorithms. In the external evaluation phase, we also compared the best classifier result in our proposed method with state-of-the-art CF algorithms as a baseline model, i.e., BSKNN and SVD. Our method outperformed the two mentioned algorithms, although this was a slight advantage. Aside from somewhat superior metrics, reducing the problem space from 18 activities to seven tourism categories makes profile development easier because the user does not need to interact with all 18 activities to develop profiles using the CF method. We found that our proposed method was able to capture and predict users' interests from a few explicit information provided by new users.

Comparatively, CF algorithms require more users rating records to achieve a close performance to our method.

References

[1] M. Gao, K. Liu, and Z. Wu, "Personalisation in web computing and informatics: Theories, techniques, applications, and future research," *Inf. Syst. Front.*, vol. 12, no. 5, pp. 607–629, 2010, doi: 10.1007/s10796-009-9199-3.

[2] J. Borràs, A. Moreno, and A. Valls, "Intelligent tourism recommender systems: A survey," *Expert Syst. Appl.*, vol. 41, no. 16, pp. 7370–7389, 2014, doi: https://doi.org/10.1016/j.eswa.2014.06.007.

[3] L. Sebastia, I. Garcia, E. V. A. Onaindia, and C. Guzman, "e-Tourism: a tourist recommendation and planning application," *Int. J. Artif. Intell. Tools*, vol. 18, no. 05, pp. 717–738, Oct. 2009, doi: 10.1142/S0218213009000378.

[4] T. Tlili and S. Krichen, "A simulated annealing-based recommender system for solving the tourist trip design problem," *Expert Syst. Appl.*, vol. 186, p. 115723, 2021, doi: https://doi.org/10.1016/j.eswa.2021.115723.

[5] W.-S. Yang and S.-Y. Hwang, "iTravel: A recommender system in mobile peer-to-peer environment," *J. Syst. Softw.*, vol. 86, no. 1, pp. 12–20, 2013, doi: https://doi.org/10.1016/j.jss.2012.06.041.

[6] R. A. Abbaspour and F. Samadzadegan, "Time-dependent personal tour planning and scheduling in metropolises," *Expert Syst. Appl.*, vol. 38, no. 10, pp. 12439–12452, 2011, doi: https://doi.org/10.1016/j.eswa.2011.04.025.

[7] P. Vansteenwegen, W. Souffriau, G. Vanden Berghe, and D. Van Oudheusden, "The City Trip Planner: An expert system for tourists," *Expert Syst. Appl.*, vol. 38, no. 6, pp. 6540–6546, 2011, doi: https://doi.org/10.1016/j.eswa.2010.11.085.

[8] Z. Jia, Y. Yang, W. Gao, and X. Chen, "User-Based Collaborative Filtering for Tourist Attraction Recommendations," in *2015 IEEE International Conference on Computational Intelligence & Communication Technology*, 2015, pp. 22–25, doi: 10.1109/CICT.2015.20.

[9] F. Ricci, L. Rokach, and B. Shapira, "Recommender Systems: Introduction and Challenges BT -

Recommender Systems Handbook," F. Ricci, L. Rokach, and B. Shapira, Eds. Boston, MA: Springer US, 2015, pp. 1–34.

[10] X. Yu, F. Jiang, J. Du, and D. Gong, "A cross-domain collaborative filtering algorithm with expanding user and item features via the latent factor space of auxiliary domains," *Pattern Recognit.*, vol. 94, 2019, doi: 10.1016/j.patcog.2019.05.030.

[11] X. Yu, Q. Peng, L. Xu, F. Jiang, J. Du, and D. Gong, "A selective ensemble learning based two-sided cross-domain collaborative filtering algorithm," *Inf. Process. Manag.*, vol. 58, no. 6, 2021, doi: 10.1016/j.ipm.2021.102691.

[12] M. Ahmadian, M. Ahmadi, S. Ahmadian, S. M. J. Jalali, A. Khosravi, and S. Nahavandi, "Integration of Deep Sparse Autoencoder and Particle Swarm Optimization to Develop a Recommender System," in *2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2021, pp. 2524–2530, doi: 10.1109/SMC52423.2021.9658926.

[13] S. Ahmadian, N. Joorabloo, M. Jalili, and M. Ahmadian, "Alleviating data sparsity problem in time-aware recommender systems using a reliable rating profile enrichment approach," *Expert Syst. Appl.*, vol. 187, 2022, doi: 10.1016/j.eswa.2021.115849.

[14] X. Yu, Y. Chu, F. Jiang, Y. Guo, and D. Gong, "SVMs classification based two-side cross domain collaborative filtering by inferring intrinsic user and item features," *Knowledge-Based Syst.*, vol. 141, pp. 80–91, 2018.

[15] Z. Sun *et al.*, "Research commentary on recommendations with side information: A survey and research directions," *Electron. Commer. Res. Appl.*, vol. 37, p. 100879, 2019.

[16] S. Ahmadian, P. Moradi, and F. Akhlaghian, "An improved model of trust-aware recommender systems using reliability measurements," 2014, doi: 10.1109/IKT.2014.7030341.

[17] F. Tahmasebi, M. Meghdadi, S. Ahmadian, and K. Valiallahi, "A hybrid recommendation system based on profile expansion technique to alleviate cold start problem," *Multimed. Tools Appl.*, vol. 80, no. 2, 2021, doi: 10.1007/s11042-020-09768-8.

[18] H. Khalid and S. Wu, "Reducing the cold-start problem by explicit information with mathematical set theory in recommendation systems," *Int. J. Eng. Comput. Sci.*, vol. 5, no. 8, pp. 17613–17626, 2016.

[19] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, and A. Galstyan, "A survey on bias and fairness in machine learning," *ACM Comput. Surv.*, vol. 54, no. 6, pp. 1–35, 2021.

[20] M. S. Sorower, "A literature survey on algorithms for multi-label learning," *Oregon State Univ. Corvallis*, vol. 18, pp. 1–25, 2010.

[21] G. Tsoumakas and I. Katakis, "Multi-label classification: An overview," *Int. J. Data Warehous. Min.*, vol. 3, no. 3, pp. 1–13, 2007.

[22] E. Spyromitros, G. Tsoumakas, and I. Vlahavas, "An empirical study of lazy multilabel classification algorithms," in *Hellenic conference on artificial intelligence*, 2008, pp. 401–406.

[23] L. Rokach, A. Schclar, and E. Itach, "Ensemble methods for multi-label classification," *Expert Syst. Appl.*, vol. 41, no. 16, pp. 7507–7523, 2014, doi: https://doi.org/10.1016/j.eswa.2014.06.015.

[24] A. Omar, T. M. Mahmoud, T. Abd-El-Hafeez, and A. Mahfouz, "Multi-label Arabic text classification in Online Social Networks," *Inf. Syst.*, vol. 100, p. 101785, 2021.

[25] A. Schulz, E. L. Menc\'\ia, and B. Schmidt, "A rapid-prototyping framework for extracting small-scale incident-related information in microblogs: Application of multi-label classification on tweets," *Inf. Syst.*, vol. 57, pp. 88–110, 2016.

[26] J. Wang, Y. Yang, J. Mao, Z. Huang, C. Huang, and W. Xu, "Cnn-rnn: A unified framework for multi-label image classification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2285–2294.

[27] J. Zhang, Z. Zhang, Z. Wang, Y. Liu, and L. Deng, "Ontological function annotation of long non-coding RNAs through hierarchical multi-label classification," *Bioinformatics*, vol. 34, no. 10, pp. 1750–1757, May 2018, doi: 10.1093/bioinformatics/btx833.

[28] C. Sanden and J. Z. Zhang, "Enhancing Multi-Label Music Genre Classification through Ensemble Techniques," in *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2011, pp. 705–714, doi: 10.1145/2009916.2010011.

[29] X. Chen, M. Vorvoreanu, and K. Madhavan, "Mining Social Media Data for Understanding Students' Learning Experiences," *IEEE Trans. Learn. Technol.*, vol. 7, no. 3, pp. 246–259, 2014, doi: 10.1109/TLT.2013.2296520.

[30] V. S. Tidake and S. S. Sane, "Multi-label Classification: a survey," *Int. J. Eng. Technol.*, vol. 7, no. 1045, 2018.

[31] G. Tsoumakas, I. Katakis, and I. Vlahavas, "Mining multi-label data," in *Data mining and knowledge discovery handbook*, Springer, 2009, pp. 667–685.

[32] D. Carrillo, V. F. López, and M. N. Moreno, "Multi-label classification for recommender systems," *Trends Pract. Appl. Agents Multiagent Syst.*, pp. 181–188, 2013.

[33] Y. Zheng, B. Mobasher, and R. Burke, "Context Recommendation Using Multi-label Classification," in *2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)*, 2014, vol. 2, pp. 288–295, doi: 10.1109/WI-IAT.2014.110.

[34] A. Rivolli, L. C. Parker, and A. C. P. L. F. de Carvalho, "Food Truck Recommendation Using Multi-label Classification BT - Progress in Artificial Intelligence," 2017, pp. 585–596.

[35] A. Elhassan, I. Jenhani, and G. Ben Brahim, "Remedial actions recommendation via multi-label

classification: a course learning improvement method," *Int. J. Mach. Learn. Comput.*, vol. 8, no. 6, pp. 583–588, 2018.

[36] R. Barta, C. Feilmayr, B. Pröll, C. Grün, and H. Werthner, "Covering the Semantic Space of Tourism: An Approach Based on Modularized Ontologies," 2009, doi: 10.1145/1552262.1552263.

[37] A. Moreno, A. Valls, D. Isern, L. Marin, and J. Borràs, "SigTur/E-Destination: Ontology-based personalized recommendation of Tourism and Leisure Activities," *Eng. Appl. Artif. Intell.*, vol. 26, no. 1, pp. 633–651, 2013, doi: https://doi.org/10.1016/j.engappai.2012.02.014.

[38] P. Prantner, Y. Ding, M. Luger, Z. Yan, and C. Herzog, "Tourism ontology and semantic management system: state-of-the-arts analysis," in *IADIS International Conference WWW/Internet*, 2007, pp. 111–115.

[39] P. K. Kopalle and D. R. Lehmann, "Alpha Inflation? The Impact of Eliminating Scale Items on Cronbach's Alpha," *Organ. Behav. Hum. Decis. Process.*, vol. 70, no. 3, pp. 189–197, 1997, doi: https://doi.org/10.1006/obhd.1997.2702.

[40] A. G. Yong, S. Pearce, and others, "A beginner's guide to factor analysis: Focusing on exploratory factor analysis," *Tutor. Quant. Methods Psychol.*, vol. 9, no. 2, pp. 79–94, 2013.

[41] D. Ganda and R. Buch, "A survey on multi label classification," *Recent Trends Program. Lang.*, vol. 5, no. 1, pp. 19–23, 2018.

[42] Z. Wen, "Recommendation system based on collaborative filtering," *CS229 Lect. notes*, 2008.

[43] Y. Koren, "Factor in the Neighbors: Scalable and Accurate Collaborative Filtering," *ACM Trans. Knowl. Discov. Data*, vol. 4, no. 1, Jan. 2010, doi: 10.1145/1644873.1644874.

[44] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural Collaborative Filtering," in *Proceedings of the 26th International Conference on World Wide Web*, 2017, pp. 173–182, doi: 10.1145/3038912.3052569.

[45] K. Gupta, M. Yelahanka Raghuprasad, and P. Kumar, *A Hybrid Variational Autoencoder for Collaborative Filtering*. 2018.