



A case study for intelligent event recommendation

Mahsa Badami¹ · Faezeh Tafazzoli² · Olfa Nasraoui¹

Received: 30 December 2016 / Accepted: 29 March 2018 / Published online: 3 May 2018
© Springer International Publishing AG, part of Springer Nature 2018

Abstract

Social networks, along with their “event” organization, planning, and sharing tools, play an important role in connecting and engaging individuals and groups. These online spaces thrive with multifaceted activities and interests which give rise to rich content and user interaction that often crossover to the world of events. For these reasons, the data trails associated with “events” in the virtual world can be complex and challenging to understand and predict. This paper presents our efforts to build an interpretable framework to analyze event data and recommend relevant events to social media users with different preferences. The datasets for this challenge were provided by a competition on Kaggle. We conduct an extensive data analysis and exploration to help gain a better understanding of the data. We then proceed to the critical phase of feature engineering, storytelling and modeling for computing event recommendations. We explore fuzzy approximate reasoning for modeling because of its rich linguistic expression ability which allows handling uncertainty, while maintaining human interpretability of the built models and predictions. This interpretability is critical in the data mining enterprise because data mining often requires team collaboration and yields results that need to be consumed by people of diverse technical and non-technical background. Such teams tend to question the meaning of models and emphasize the importance of telling stories from the data. We evaluate our event recommendation system on a real-world dataset with more than one million events and 38,000 users. The proposed methodology achieved 70% accuracy, outperforming existing event recommendation algorithms.

Keywords Recommender system · Event recommendation · Collaborative filtering · Adaptive neuro-fuzzy classifier · Data science · Big data · Feature engineering · Data mining · Explainability · Interpretability

1 Introduction

1.1 Recommender systems

Recommender systems (RSs) first appeared in early works in cognitive science, information retrieval and consumer behavior in marketing [44,71,75]. However, they became an important research area in the mid-1990s when the major

goal of RS was to estimate the rating for the items in order to be selected and shown to a user who had not seen those items before. Using this estimation, the system was able to decide whether to recommend the items to the user or not; intuitively, a RS would choose the items with the highest rating [3]. Over the years, with the fast development of recommender systems and their wider applications, RSs started to learn user’s preferences, such as likes and dislikes [7]. Recommender systems are currently widely used in industrial e-commerce systems such as Amazon and Netflix, where both the users and the system have to deal with enormous information overload in order to be able to provide a personalized recommendation (e.g., content, books, services, movies) [46].

A wide variety of recommender systems have been developed for various domains such as music [13,54], movies [12], books [61], e-commerce [73], location recommendation [36, 88,95], point of interest (POI) recommendation [20,45,87], next-POI recommendation [32]. For example, transaction-based recommender systems (TBRSS) generate new transactional recommendations [81], such as next-item/basket

✉ Mahsa Badami
mahsa.badami@louisville.edu

Faezeh Tafazzoli
faezeh.tafazzoli@louisville.edu

Olfa Nasraoui
olfa.nasraoui1@louisville.edu

¹ Knowledge Discovery and Webmining Lab, Department of Computer Engineering and Computer Science, University of Louisville, Louisville, USA

² Multimedia Research Lab, Department of Computer Engineering and Computer Science, University of Louisville, Louisville, USA

recommendation [73,81,82]. Recently, an attention-based transaction embedding model (ATEM) was proposed to effectively recommend the next item within a transactional context [82]. ATEM learns an attentive context through a shallow wide-in-wide-out neural network which outputs the next choice with a high probability.

These approaches work well when significant preference information is available but face challenges in highly sparse settings. The most extreme case of sparsity, known as cold start, occurs when no or few preference information is available for a given user or item. In such cases, hybrid approaches are typically employed to generate a personalized recommendation by incorporating auxiliary content information (multimodal information, side information, etc.). [20] proposed a content-based approach to deal with the cold-start problem in the context of music, where the goal is to model the preference of users from observed user-item associations and then use it to predict the items a user may like, based only on the last song the user has listened to. Their method is based on the idea that most recently played songs will most likely reflect the users current emotional state or activity and thereby contribute to the modeling of the listening context. In essence, by mining both sequential behavior and content features simultaneously, this next-item recommendation approach helps the user explore new items.

Even though the next-event recommendation system seems quite similar to the next-item recommendation concept, an effective event recommender system does not need to learn the transition between the items, but mainly the relevance. The reason is that an event is a one-time occurrence item with a short lifetime unlike most items, (read more in Sect. 1.3). In addition, event recommendation highly depends on the users profile, his/her network as well as event characteristics. For example, one is more interested to go to an event for the first time if more of his/her friends have shown interests in the event. Hence, in this work, we are interested in developing a framework built on user preferences and item properties.

Finally, because our main focus is on event recommender systems, we will cover mainly this family of recommender systems in the next subsection.

1.2 Event recommender systems

Aside from the traditional online social networks, event-based social networks (EBSNs) like Meetup and Facebook [69] have experienced a rapid growth in recent years and are changing people's ways of social interaction. Nowadays, EBSNs are one of the most popular services in social media [7,64] due to their ability to create potential social interactions among their members. For instance, currently, Meetup has 30.30 million members who attend 608,036 meetup events, monthly in 182 countries [49]. In addition,

79% of millennials prefer to buy an experience over buying an object. Considering the fact that a quarter of the US population consists of millennials [50], this preference means a rise in the experience economy.

Event recommender systems, as a main part of EBSNs, play a central role by suggesting relevant events to the user, and at the same time assisting event organizers to predict the overall interest in a particular event. Many approaches have been proposed to recommend different items such as movies or books; however, there are few studies that aim to suggest forthcoming events to users [3,7,46]. Yet, social activities and events are an inherent part of our lives, making it critical to help answer event-related questions such as “what”, “how”, “where”, “when”, “why” and “who” [46]. Regardless of these questions, it is hard to give information and recommend events that may interest people. Certain relationships between people may affect their decisions [70]. People declare their likes and dislikes in different levels, and these meaningful preferences compose a person's virtual and physical social universe [14]. We refer to any entity with the following features as an event: (a) having a participatory aspect, i.e., an activity is offered and people can decide whether to join or not, (b) requiring the physical presence of people at the location where the activity takes place, or (c) being offered within a given time interval and requiring a certain amount of time to be completed [14].

In the context of event recommenders, most existing work [29,47,69] learns the behavior of a member from his/her attendance history by developing geographical, social and/or temporal models to compute the relevance score between the member and the upcoming event. Zhang and Chow [92,94] employed a nonparametric kernel density estimation (KDE) method to model the geographical check-in distribution of locations for each user over the latitude and longitude coordinates. In [93], friend-based collaborative filtering was used to recommend new locations for users based on their friends visited locations. [17] proposed a social event recommendation method that exploits a users social interaction relations and collaborative friendships. Another study [38] on Meetup investigated how social network, user profiles and geo-locations affect user participation when the social event is held by a single organizer.

Many companies are competing against each other to delight people with spot-on event recommendation [69], such as:

- *Facebook* events are a great source for events, and Events For You was introduced in 2014. They look at the users' information to give a recommendation i.e., liked pages, groups, communities, events attended by friends, etc., much like in EBSN approaches. The events are represented using Facebook's own open graph vocabulary,

giving description, name, place, start time and possible ticket link.

- *Eventbrite* recommends events using location, topic and type filters set by the user. It also tracks past attended events and information from their social graph.
- *Meetup* is an EBSN that is often used in relevant academic work on event recommendation. Users can join groups that are creating events, and they can share and comment on the events. Events can be closed to outside users from the group. At the time of writing, this paper, Meetup.com, with over 24 million members, with approximately 200,000 groups in 181 countries, generates over 3 million RSVPs every month.

However, this is not the only use of such systems. Event recommendations are now promoted to use for education [72], marketing [8], investment [48], pitching ideas for startups, galleries for new artists, political events for rallies, sociology events for activists, etc.

1.3 Challenges in event recommendation

Event recommendation is a challenging task since an event is considered as an ephemeral one-time occurrence item that can only be experienced (or consumed) once, and then, it expires, unlike most other items (such as movies or music) in the context of classical recommendation systems [23]. Moreover, due to their short lifetime, recommending events can be particularly challenging [70]. In addition, since events that need to be recommended are by their nature, newly created items, event recommendation systems face severe challenges due to highly sparse data and they suffer from a constant new-item cold-start problem. Additionally, the lack of effective services that support users in finding and managing events in social applications makes the prediction process even more challenging [14].

In a real-life scenario, different from traditional recommendation systems, an event recommender deals with three new types of information, i.e., heterogeneous online and offline social relationships, geographical features of events and implicit rating data from users. Such applications require encountering and combining these types of data and dealing with features of items and user behavior which are often subjective, vague and imprecise [22].

Fuzzy set theory and techniques can offer an attractive solution to handle this fuzziness and uncertainty [42]. A fuzzy logic system (FLS) can be defined as the nonlinear mapping of an input dataset to a scalar output dataset [42]. When a problem has dynamic behavior, fuzzy logic is a suitable tool. In other words, fuzzy logic finds its strength in providing accurate solutions to problems that involve the manipulation of several variables. It is generally difficult for a user to express his/her interest in an event with exact numbers, and

this is a strong motivation to use fuzzy set theory and techniques because they lend themselves well to handling such fuzziness and uncertain issues [91]. Although representing user preferences and item features as fuzzy sets have been used in the previous research [10,56,58,59,68,83,90,91,96], fuzzy logic has not taken a prominent role in the literature on recommender systems [22,43,57,58]. Fuzzy set theory-based methods, however, have earned a good reputation for producing low-cost and good-quality solutions, given incomplete and uncertain information, as is common in recommender systems.

1.4 Summary of contributions

Traditional event recommendation systems deal with three new types of information, i.e., heterogeneous online and offline social relationships, geographical features of events and implicit rating data from users. However, most classical solutions fail in encountering and combining these types of data since features corresponding to items and user behavior are often subjective, vague and imprecise. It is generally difficult for a user to express his/her interest in an event with exact numbers, and this is a strong motivation to use techniques based on fuzzy set theory since they lend themselves well to handling such uncertainty.

Among different proposed approaches for personalized event recommender system, there are very few methods considering fuzzy logic as a suitable tool to deal with dynamic behavior. Although representing user preferences and item features as fuzzy sets have been used in the previous research, fuzzy logic has not taken a prominent role in the literature on recommender systems. Fuzzy set theory-based methods, however, have earned a good reputation for producing low-cost and good-quality solutions, given incomplete and uncertain information, as is common in recommender systems.

To this end, we propose a soft computing framework for recommending events to automatically learn a function based on a neuro-fuzzy inference system that models the likelihood of each users' interest, which can then be used to sort a list of recommendations for the user. This is a novel approach to use adaptive neuro-fuzzy classifier in order to build an event recommendation system. In this paper, we describe a pipeline which leverages a fuzzy set-based personalized recommender system to predict the events in which users will be interested based on the events they have responded to in the past, user demographic information, and the events they have seen and clicked on in a certain application.

Additionally, our system is one of the first initiatives to recommend general events to real users on a large-scale dataset with more than a million events and 38,000 users. The datasets used in this paper were provided by a competition on Kaggle in which data were collected from an event

recommendation application (app) over a two-year experiment. In our decision-making process, we rely only on the information provided by the app organizers. This makes the approach more challenging compared to other event recommendation studies where external resources could be applied to enhance the data. Hence, because we realize that the intricate characteristics of the data have a significant influence on the design and performance of recommender systems, we devoted considerable efforts to the entire data engineering process, encompassing data cleaning, preprocessing and exploration, before applying a fuzzy set-based approach to predict the relevancy of an event, to be able handle the uncertainty challenges resulting from ambiguous, imprecise or missing data.

Another important novelty of our work is that the proposed framework yields an interpretable event recommendation system. This feature provides users with a better understanding of data and behavior of predictive models. Therefore, we highlighted the critical stage of storytelling which can reveal interesting facts about our dataset, facilitating the customers' behavior understanding for business applications which utilize data mining and machine learning techniques. Interpretability and explainability are crucial for trusting AI and machine learning algorithms and are becoming a serious legal mandate in the regulated verticals of many industries [2,55]. Hence, contrary to other classical event recommendation systems, our proposed framework helps business analysts and industry researchers understand and trust the model and recommendation results better.

To summarize, our study is novel in several aspects:

1. We propose a soft computing framework for recommending events to users based on their demographic information, interest history, the behavior of people within their social network, timeframe of events, and the descriptions of the events.
2. Our system is one of the first initiatives to recommend general events to real users on a large-scale dataset with more than a million events and 38,000 users. Moreover, our proposed content-based recommender system combines the social, temporal, and contextual factors as well as geo-location information, in a systematic manner, i.e., after an extensive analysis of their effect on the recommendation process.
3. The proposed approach yields an interpretable event recommendation system, and this interpretability supports the desiderata of real-life everyday human-directed data science, where expert users of the system and consumers may have a desire to understand what stories hide in the data and why a predictive model works the way it does.
4. We conduct extensive analyses and evaluation on real-world datasets provided by an event recommendation mobile application. We compare our framework with

competitive predictive model baselines including K nearest neighbors [25], support vector machines (SVM) [24] and Naive Bayes [30]. We find that the neuro-fuzzy model outperforms other methods on standard evaluation metrics including accuracy, F -measure and statistical tests, as well as interpretability.

5. We analyze the effect of missing data on the recommendation process and show the ability of our approach to handle real-world imperfect data compared to the aforementioned benchmarks.

6. We give importance to the critical stage of storytelling which can reveal and convey interesting facts about our dataset.

1.5 Problem statement

The datasets used in this paper are provided by a competition on Kaggle [39] in which data have been collected from an event recommendation app over a two-year period.¹ We rely only on the information provided by the app organizer to make a recommendation decision. Many factors may influence a user's interest; thus, we aim to develop a framework to model these factors using fuzzy approximate reasoning to capture uncertainty in representation and neural networks for predictive modeling.

We formalize the event recommendation problem as a learning problem, where the learning function $f(<user, event>) \rightarrow \{interested, notinterested\}$ takes a user and an event as inputs and produces a confidence value indicating the level of the user's interest in attending this particular event. Based on the estimated interest degree, the system decides whether to recommend the event, as follows, with U being the set of users, and E being the set of events.

$$f : U \times E \rightarrow L = \{interested, notinterested\} \quad (1)$$

$$(user, event) \rightarrow l$$

1.6 Unique features of our proposed methodology

In our methodology, we capture predictive information through a simple feature engineering process and do not impose more computational complexity on an online social event recommendation service. Moreover, since our original aim was to solve the Kaggle challenge, no rating data, which is one of the important sources of information for recommendation tasks, were provided in the Kaggle dataset.

We thus present an event recommender system which is able to recommend a wide range of relevant events since social media services require a generic recommender system.

¹ The event recommendation engine challenge was the first competition launching under the "Kaggle Startup Program". Starting January 2013, 223 teams took participation in the competition over 40 days.

In general, the above-mentioned research efforts show that our proposed data science pipeline is a viable approach for the goal of making predictions using the set of relevant features that we designed and ended up using in order to make event recommendations. In our study, we used a combination of features which have proved to have high impact on event recommendation [33]. To the best of our knowledge, there is no published large-scale real-world study on the subjects we discuss in the present paper.

In general, our proposed system is one of the first initiatives to recommend **general** events to real users **on a large scale**. Moreover, our proposed content-based recommender system combines the social, temporal, and contextual factors as well as content and geo-location data, in a systematic manner, i.e., after an extensive analysis of their effect on the recommendation process.

Our approach is the first to employ fuzzy approximate reasoning and to compare to conventional state-of-the-art data mining methods.

The rest of this paper is organized as follows: Section 2 provides an overview of the recommendation systems and event recommendation systems in particular. Section 3 dives into the first data science stages of analyzing context, data quality and feature engineering. This is followed by the entire data science pipeline methodology in Sect. 4. Section 5 presents our experimental results. Finally, Sect. 6 concludes by summarizing the proposed approach and its contributions.

2 Related work

2.1 Event recommendation systems

Real-time micro-blogging and social network services have recently witnessed exponential growth [1,62]. With millions of active web users, an event recommender system, a specialized type of recommender system, helps users to find an unseen event. Event recommenders, like other-item recommenders, can be roughly divided into two categories (with additional possible hybrid solutions): collaborative filtering and content-based filtering. The collaborative filtering approach [40,51] builds a model based on finding similar users, i.e., with similar behavior, as well as the user's past behavior (e.g., the past preference ratings, page views, or purchases either by the active user or by other similar users). The content-based approach [28] utilizes the events with the most similar characteristics in order to recommend additional events with the same properties. Moreover, there are hybrid approaches that consider both similar users and similar items to predict the user's interests [22,41]. There is also pioneering work in the social event recommendation literature [46,84] some of which have considered geographical information [18,69]. However, most existing methods require

explicit preference constraints, or cover only a limited portion of available information in event recommendation tasks.

As mentioned earlier, one of the challenges in event recommender systems is to model the three important sources of information, i.e., the geographical features, heterogeneous online and offline social relationships and user implicit rating data, jointly. To address this issue, Qiao et al. [69] recently presented a Bayesian latent factor model which considers this information for an offline recommendation process. They tested their model on real-world datasets. Their approach extracts hidden relationships between groups of users and events.

In another innovative direction, Guo et al. [34] presented a model for group activity organization and recommendation in real-world settings. They developed a framework, called MobiGroup, which recommends relevant activities to the users. In addition to group computing, intelligent tagging and context learning, they used a combination of mobile sound sensing and cross-community mining techniques. However, their experiment was limited to a small set of users (45 participants) and considered only the activities for “hot” and “social” posts. Generally, existing event recommendation approaches are mainly designed for specific events, for instance, iCITY [11] which is a content-based recommender system for cultural events in the city of Torino, Italy.

In another direction, there has been interesting research to investigate the factors in the recommendation of different items that influence the user's choice [6,63]. For example, Cena et al. [14] studied a set of these factors in order to provide empirical evidence on the information and the extent that should be taken into account by an event recommender system. To achieve that goal, they trained several content-based scoring functions to predict the score assigned to a given event by a user. They designed two online surveys where people could express their interest in participating in certain events, all related to wine and food.

Using a set of these features including content, time, location and social network, Waga et al. [79] presented a context-aware recommender system in which user profiles are employed by monitoring their behavior. Their recommender system uses the MOPSI geo-tagged database, which contains a user-generated photograph collection and service database. However, since the model is implemented as a prototype solution within MOPSI, the only reported evaluation is based on user satisfaction and there is no other validation metric to evaluate the model performance.

2.2 Fuzzy approximate reasoning

Fuzzy approximate reasoning is based on multi-valued logic which attempts to provide a better understanding of the result of a statement by having a more approximate representation than precise [78]. It is a computational framework that makes

use of fuzzy set theory and fuzzy assignment of elements to sets [89]. Fuzzy approximate reasoning makes use of these membership degrees to evaluate the degree of truth of logical propositions instead of using Boolean logic. Thus, it can be characterized by linguistic terms rather than by numbers. This characteristic makes it very efficient in dealing with complicated problems when conventional analytical methods are infeasible or too expensive. In essence, not only do fuzzy sets handle ambiguity elegantly, but they also provide a computational mechanism to express a linguistic expression in a mathematical form that is suitable for computation and reasoning. This rich representation, that bridges the linguistic and the numerical worlds, makes fuzzy approximate reasoning attractive for use in certain data science applications.

Fuzzy inference systems A fuzzy inference system (FIS) is a paradigm in soft computing which is capable of handling computing with vague, imprecise or missing data [89]. It performs a nonlinear mapping from an input space to an output space by deriving conclusions from a set of fuzzy if-then rules and known facts [37]. Typically, a FIS is composed of the following major components: a Fuzzifier that assigns a membership degree to each crisp input dimension in the input fuzzy sets; a Knowledge Base characterized by fuzzy sets of linguistic terms; a Rule Base containing a set of fuzzy if-then rules; an Inference engine that performs fuzzy reasoning, and a Defuzzifier that generates crisp output values. A graphical representation of a generic FIS is shown in Fig. 1.

Fuzzy recommender systems Fuzzy logic can offer rich insights to deal with non-stochastic uncertainty, which is often present in the recommendation process. Yager [85] presented a user's preference as a fuzzy set over the items visited by the user. Using the membership degree, this framework infers the preference of a new item by the user. Similarly, Zenebeet et al. [90,91] built a framework in which items are represented as fuzzy sets by employing different fuzzy set-based similarity measures, including correlation-like, proximity, cosine and fuzzy set theoretic.

Perny and Zucker [65,66] proposed a hybrid approach in order to deal with decision-making problems. They presented the term “Collaborative Decision Support” (CDS) for the first time to handle users who seek recommendation for their personal choices. The presented approach combines a

content-based and a collaborative filtering method. Cornelis et al. [22] pursued another hybrid approach based on fuzzy logic which recommends an event to a user if other similar users are interested in it, or if it is similar to any previous event attended by the user. However, no evaluation or clear insight is provided on its performance, especially for large-scale datasets.

In another interesting direction, Cao and Li [10] focus on linguistic terms in order to allow users to express their needs for an item. Following the same idea, Cornelis et al. [23] represented user preferences using two fuzzy relations, positive and negative sentiment. Their recommender system predicts positive and negative feelings of the user regarding an item using the aforementioned fuzzy relations. Nasraoui and Petenes [58] used fuzzy approximate reasoning to infer recommendations in the context of clickstream user sessions, which are implicit preferences. The fuzzy sets were built by clustering user sessions into anonymous group user profiles. Recently, Wu et al. [83] presented a fuzzy preference tree-based recommender system for personalized business-to-business e-services. The model is able to recommend tree-structured items based on fuzzy logic capturing both extensionally and intentionally expressed preferences of users.

3 A close look at context, data quality and feature engineering

Unlike the related areas of machine learning, pattern recognition and data mining, data science is not only concerned with learning, modeling, mining or recognition [4,80]. It is first and foremost about understanding the data using scientifically grounded methods, while (and unlike the related field of statistics) being able to anchor and relate this understanding into the original context of the application domain that gave rise to the data, and while also handling larger datasets with higher dimensionality, noise and heterogeneity of sources [27]. One of the fundamental principles of data science is hence, knowing the data, which therefore requires starting with a thorough preview of different parts of the dataset. A proper data analysis and exploration leads to greater confidence in the adopted approach as well as experiments, and can play an important role to avoid misusing biased data and to aid in interpreting the results throughout the entire data science pipeline, including the results of predictive modeling [80]. Moreover, this crucial first stage gives considerable insight about user behavior which can guide the design of a powerful predictive pipeline.

In this section, we first describe the datasets provided by the Kaggle competition and then elaborate the preprocessing approaches that will be applied. We also explain the feature engineering process that extracts powerful features from the

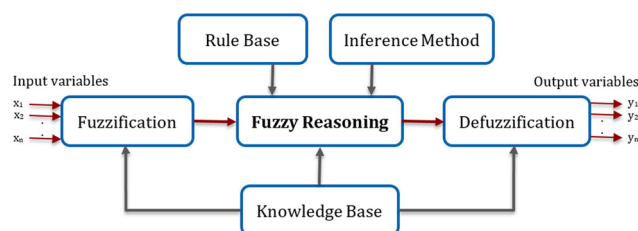


Fig. 1 Architecture of fuzzy inference system

Table 1 Basic statistics of the original dataset

#User–events	Interested	Not interested	Unknown	#Users	#Events
15,399	26%	3%	69%	38,210	1,048,576

datasets, and then, we proceed to the critical stage of story-telling which can reveal and convey interesting facts about our dataset.

3.1 Data description

We use datasets which were provided by a sponsor on the Kaggle website [39].

Five different datasets were given: **train**, **users**, **user_friends**, **events**, and **event_attendees**.

Train has six columns: **user**, **event**, **invited**, **timestamp**, **interested** and **not_interested**. **Event** and **user** are IDs identifying events and users in the system. The binary variable **invited** indicates whether the user has been invited to the corresponding event. **Timestamp** shows the approximate time when the user has checked the event in the app. Finally, **interested** and **not_interested** are binary variables indicating whether a user has clicked on the “Interested” or “Not Interested” button for an event. The **Users** set contains demographic data about some of the users using the app, and it has features such as: user’s location, gender and birth year. The user’s social data are included in the **user_friends** set which consists of the list of user’s friends. The **Events** set has information regarding each event such as location, start time and a bag-of-words description. The bag of words contains the frequency of the 100 most common word stems occurring either in the name or description of a large random subset of available events. Finally, **event_attendees** provides information on the users attending each event. Table 1 shows the properties of the train dataset. There are 15,399 unique pairs of user–event in which 26% of the users are interested in the event, 3% are not interested and 69% did not indicate their preferences.

Considering all provided sets, there are 38,210 users and 1,048,576 events. This is a large dataset, compared to other available event recommendation datasets. It is also quite complex and hence requires a more thorough analysis.

3.2 Diving into the data

In the section, we take a closer look at the data and attempt to discover interesting facts behind some of the data peculiarities to reveal stories about the data. This is a very special phase of real-life data science projects.

1. User’s demographic features Figure 2 shows a histogram of the participants age. As we can see, most of the participants are young, with only 12 users who are older than 85.

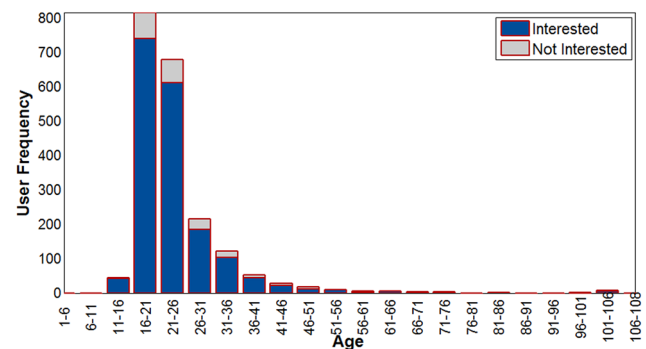


Fig. 2 Number of users of different ages

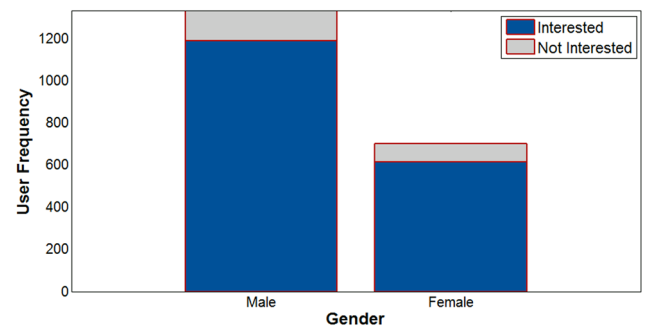


Fig. 3 Number of female and male users

These outliers could be the result of random filling of profiles by some users. Based on the provided data, most of the participants are male, as shown in Fig. 3.

2. Reasons behind time range imbalance Figure 4 shows the distribution of seasons of the events. Most of the events for which users provided labels occurred in Fall and Winter. Based on the dataset, users started using the app in the period of June 22, 2012, to December 12, 2012, with events scheduled from June 22, 2012, to December 31, 2013. Hence, the provided dataset covers the data regarding users active in the last 6 months of 2012. This explains the reason why the main response from users is mainly in these seasons, and specifically in Fall. Users tend to plan for temporally closer events rather than events further in the following year. Note that there may be more than one event on the same date; for example, on the weekend of Halloween of 2012 alone, there were 500 different events. Another reason for the higher number of events in Winter could be several important Winter events such as Christmas, Kwanzaa, Hanukkah, New Year’s Eve and Valentine’s day. As a result, there are more happenings during these seasons.

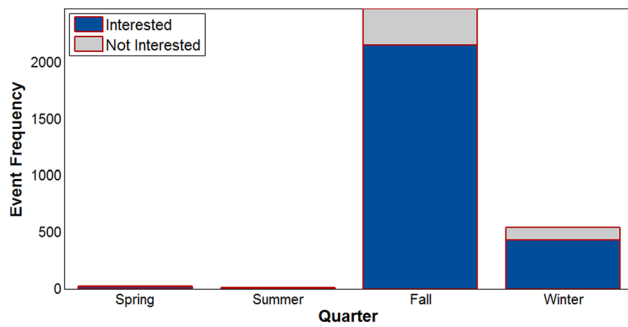


Fig. 4 Frequency of events in 4 quarters



Fig. 5 Frequency of events during weekdays/weekend

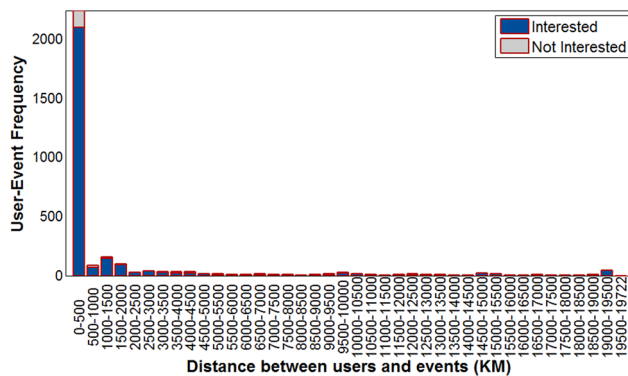


Fig. 6 Event frequency versus distance between user's location and event's location

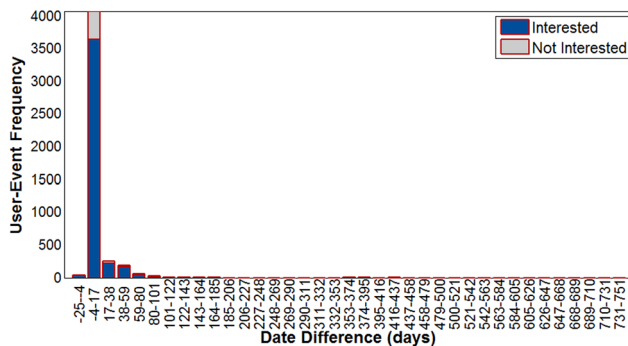


Fig. 7 Event frequency versus difference between the date user checks an event and the event's start date

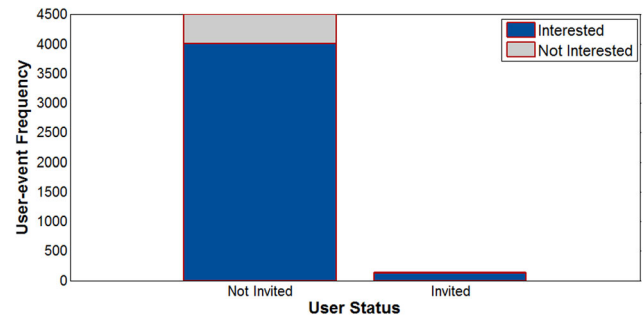


Fig. 8 Popularity of events sorted based on their dates

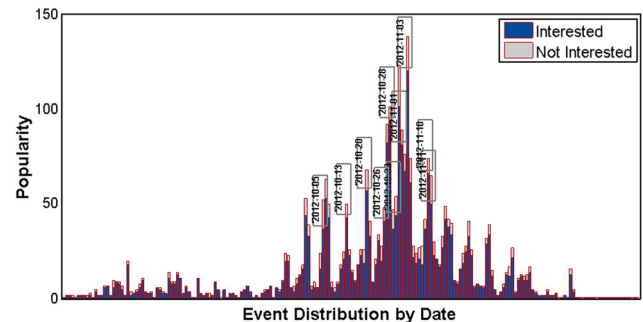


Fig. 9 Date distribution of events

3. *Seasonal events that dominate attendance* Figure 5 shows the distribution of weekend versus weekday events. Although one might expect more events during weekends compared to weekdays, since the category of events is not reported, we should be careful not to draw any hasty conclusions at this point. Events that are school related would generally be scheduled during weekdays. Moreover, the higher number of days for weekdays could have led to higher frequency.

4. *Location* Another interesting finding based on the dataset is that users tend to go to events within 10 miles of distance, as shown in Fig. 6

5. *Commit at the last minute* Figure 7 indicates that users show more interest in the events which are scheduled within less than a week. Both previous findings are sensible since people are more willing to express interest to go to nearby events, a few days ahead.

6. *Popular events* Figures 8 and 9 depict the popularity and the frequency of each event with time, respectively. Recall that the last date seen by users is December 12, 2012; so, we cannot claim any findings regarding other events such as Christmas and New Year. These figures confirm that the Halloween week was very popular.

7. *Personal connections to events*

- Did my friend invite me? Dealing with social media, one may suggest that inviting a user to an event can have a positive effect on user's interest, since it brings the event to the user's attention. Figure 10 confirms this assumption

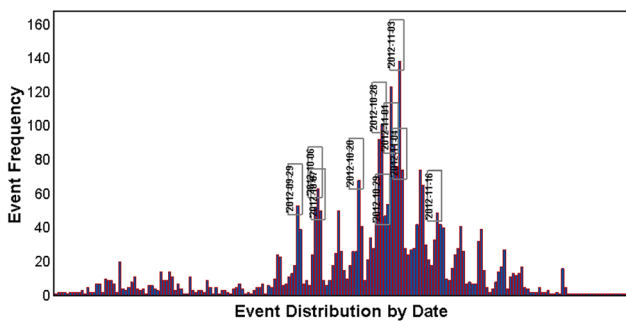


Fig. 10 Event attendance frequency of users who are invited to an event versus those who are not invited

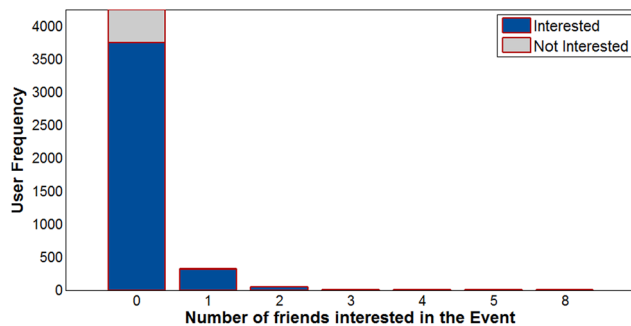


Fig. 11 Distribution of counts of friends of users interested in an event

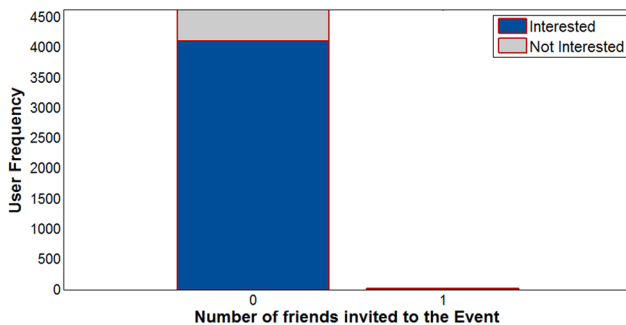


Fig. 12 Attendance frequency of users depends on whether friends are invited to an event

as all invited users show interest in the events (no gray area in the right bar). On the other hand, not being invited does not have any negative effect in this app, since the users were mainly interested into the events (more blue area in the left bar).

- Are my friends going? Figures 11 and 12 show the frequency of the number of user's friends who were interested and/or invited to a particular event, respectively. Intuitively, people show more interest in an event when their friends are going to attend as well. Here, although there are very few cases where user's friend are either invited or interested in an event, the user has shown to be very likely to click on "interested" in case his/her friends are possibly going. This is a feature which could play an

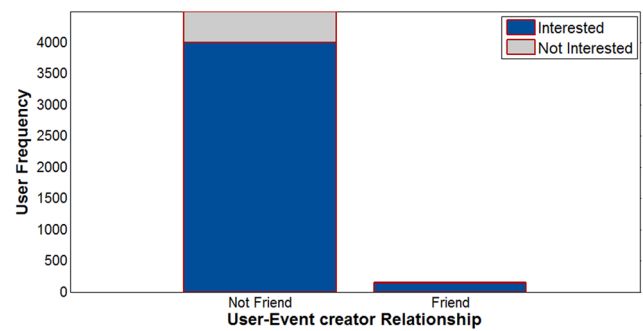


Fig. 13 Attendance frequency of users depends on whether friend created the event that s/he is invited to

important role in predicting user's interest following the common human behavior.

- Did my friend organize the event? Figure 13 indicates that being friends with the event's creator has a positive effect on the user's interest in the corresponding event. However, as shown in the left column of Fig. 13, not all event creators have friends among event attendees. We consider these important behavioral features in order to be able to capture the social effect of the users network into the designed features and then to build an accurate event recommendation system.

3.3 Feature engineering

Feature engineering in data mining crucially depends on exploiting the intricacies of domain knowledge. Our goal in this phase is to engineer features that may contribute to the recommendation process. These features are extracted for a given (user, event) pair based only on the provided dataset and no additional source of information other than the observations revealed through the analysis in Sect. 3.2. These features are roughly categorized into three groups.

User-level features Motivated by our data-driven observations in Sect. 3.2, a good recommender system should be able to capture the behavior developed among users within a social group. Therefore, we extract several behavioral attributes with promising impact on event recommendation. Since users can be influenced by their friends, features related to the number of friends who are either invited or interested in an event can be a factor in determining whether the user would be interested as well. We construct two features to capture this information, **invited_friends**, and **interested_friends**. In addition, another interesting feature that we construct is **creator_is_friend** which checks whether the event creator is a friend of the user or not. Users show more interest in going to an event which is organized by their own friends, as shown in Fig. 13.

Table 2 Dataset summary

#User–events	Interested	Not interested	#Users	#Events	#Features
4645	89%	11%	2034	3055	114

Event-level features We observed in Sect. 3.2, how information regarding the timing of an event affects the user's decision. One such information is seasonality. Therefore, we construct two features based on an event's date. One feature, **is_weekend**, basically checks whether the event will happen during weekends or not. Another feature, **quarter**, which is based on the event date, defines the season/quarter an event occurs in. Intuitively, more events are scheduled for summer, and most of the users are interested in such events. Since most US holidays are during weekends, we do not investigate whether the event happens during holidays or not. Another interesting feature is **popularity** which shows the number of users who are interested in an event. Based on our findings (Fig. 8), an event with higher popularity attracts more users.

Text-based (content) features, also, can be a source of valuable information. Here, the only text-based information is the event's description available in a bag-of-words format. We consider the words as high-dimensional features since the meanings of the words are not provided. (We were given mere IDs.) To reduce the size of the feature space, we select the top 5 discriminating words based on mutual information gain with respect to the class label (user is interested or not) [86]. Information gain is defined for each feature as:

$$\begin{aligned} \text{InfoGain}(\text{Class}, \text{Attribute}) \\ = H(\text{class}) - H(\text{Class}|\text{Attribute}) \end{aligned} \quad (2)$$

where H is the information entropy which measures the unpredictability of information content.

User–event-level features The data regarding the interactions between a user and an event contains valuable information. Many users (especially those with a busy schedule) would rather attend events that are not chronologically too close to one another; thus, we construct a feature, **time_difference**, to capture the difference between the occurrence time of an event and the time that the user checks the event's Ad. In addition, users prefer to attend events occurring within a close distance to their location. So, we constructed a feature, **distance**, to reflect the impact of geo-location. We then removed the raw features that were used to construct these features, such as latitude and longitude of an event and user location. In the next step, we analyze the extracted features and focus on data quality.

3.4 Challenges related to data quality

There are several challenges in preparing this dataset before prediction, including unknown class labels, missing values,

imbalanced classes combined with power laws, sparsity, and the presence of outliers. Below, we summarize these challenges.

Challenge 1: Missing attendance The first challenge is existence of unknown labels. This happens when a user does not show any reaction toward an event. Therefore, we do not have access to the class label for these users and thereby cannot interpret any information for the prediction. As mentioned earlier, approximately 70% of labels are unknown, which we are going to remove in this phase. These unknown labeled instances can be used for user evaluation of the proposed method in the future.

Challenge 2: Imbalanced attendance Another challenge is the issue of imbalanced classes, where the data related to the not-interested class makes up only 3% of the dataset. There are several approaches to handle imbalanced datasets. Oversampling and undersampling in data analysis are techniques used to adjust the class distribution of a dataset (i.e., the ratio between the different classes represented). Oversampling and undersampling are opposite and roughly equivalent techniques [31]. They both involve using a bias to select more samples from one class than from another. The usual reason for oversampling is to correct the existing bias in the original dataset.

Challenge 3: Missing event location or user demographics Having missing values is another challenge occurring frequently in real-world datasets. Recommender systems often suffer from sparsity which leads to difficulty in the recommendation process. Missing values have a significant effect on inferences about data. Missing data reduce the representativeness of the data and can therefore distort inferences about the population. One way to avoid such problem is to try to prevent data from missingness during the data collection process. However, it is impossible in the recommendation process, since neither users nor items provide complete information. In our datasets, there are 5 attributes which have missing values: user's birth year and the location of user and event (latitude, longitude). In order to handle missing values, we searched for missing locations in other datasets; however, there were only very few records for which we could fill the missing values.

As listed in Table 3, the locations of both users and events have the most missing values. It is worth noting that missing values in locations have about 10% overlap between user location and event location. For example, instances of not-interested class have no missing values in user birth year. Based on the data, there is an interesting observation that most

Table 3 Percentages of missing values in the whole dataset and per class (interested, not interested and unknown)

Total percentage	Birth year 1%			User location (lat., long.) 13%			Event location (lat., long.) 25%		
Percentage per class	1%	0%	0%	11%	24%	65%	24%	29%	47%
Class label	Interested	Not interested	Unknown	Interested	Not interested	Unknown	Interested	Not interested	Unknown

Table 4 Data percentage in the provided datasets

	User location	Event location	
User location	–	1%	Interested
Event location	6%	–	Not interested

users who did not fill their birth year are male. In order to gain better understanding, percentages of overlap between user and event location for each class are included in Table 4. We can see a higher percentage of overlaps in the not-interested class. This is an important feature, because a user cannot express his/her interest before knowing the event's location.

The final feature set includes the following attributes: **invited, time_difference, is_weekend, quarter, creator_is_friend, distance, popularity, invited_friends, interested_friends, age, gender, bag_of_words = C2, C3, C5, C7 and C11.**

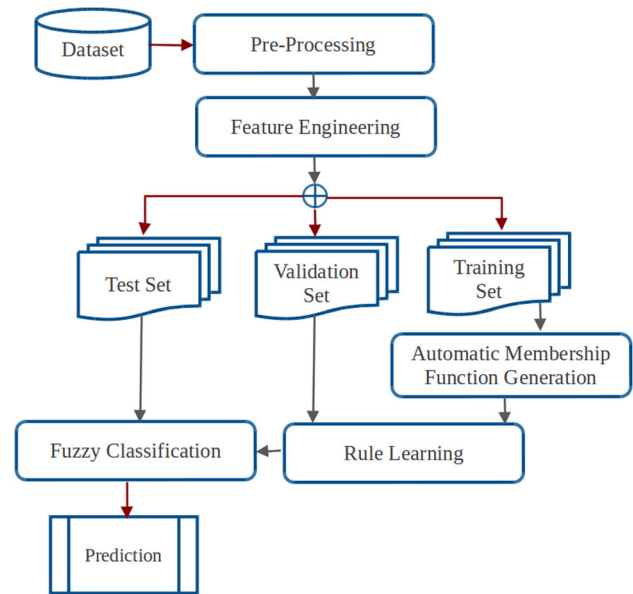
4 Proposed recommendation system modeling pipeline

We propose a personalized event recommendation system that consists of the following three main steps: (1) automatic membership function generation, (2) rule learning and (3) fuzzy classification.

Figure 14 depicts the flowchart of the proposed pipeline.

4.1 Automatic membership function generation

Understanding the data, as we have done in Sect. 3.2, plays an important role in successful feature engineering and consequently in the resulting behavior of the system. In the context of FISs, initializing the fuzzy classifier with a good set of rules can boost the system considerably. Rules are generated based on the relationship between different features. These relevancies connect various parts of multiple features to a certain class. Clearly, generating a set of rules with maximum overlap with the real distribution of the data would affect the performance of the final FIS to a great extent. Different approaches have been employed in the literature for this purpose [21,60]. Many methods, however, perform this part manually to take advantage of the initial knowledge regarding the distribution of different features. This tech-

**Fig. 14** Flowchart of the proposed event recommendation framework

nique, however, is not feasible in the presence of too many features or in the case of data with many missing values. As for recommender systems, it might not be possible to find a perfect distribution for all features.

To overcome these issues, in the first phase of our algorithm, we use a clustering-based approach to automatically generate membership functions (MFs). The data is divided into two groups based on the class labels. Then, a clustering algorithm is performed within each group, in order to create k clusters. The reason behind performing clustering within each class is to consider the probability of features with different distributions. By isolating the classes, more meaningful MFs can be extracted which could describe the data in a better way. On the other hand, performing cluster analysis on the dataset leads to obtaining similar regions of the feature space which could be analyzed as one. Cluster analysis reduces the computational cost for high-dimensional spaces, as is often the case in recommendation systems dealing with real-world data. In addition, cluster analysis helps with generating fewer rules which are useful for the FIS created in the next steps. We employ a simple yet fast clustering algorithm, k -means, as the clusters obtained by k -means depicted the most intra-cluster similarity and also the generated clusters were more representative in terms of significant difference

between the cluster features. Based on the final clusters, a set of membership functions was extracted for each cluster. Considering a cluster c_i , a membership function μ_{ir} is generated for each attribute r based on (3). Here, we employ a Gaussian membership function (MF) to obtain smooth partial derivatives of its parameters.

$$\mu_{ir} : (\phi_{ir}, \sigma_{ir}) \quad (3)$$

where ϕ is a center of cluster c_i for attribute r and σ is the variance of attribute r considering the instances assigned to into cluster c_{ir} , $r = 1, m$. Here, we employ a Gaussian membership function to obtain smooth partial derivatives of its parameters. These MFs will be used as an initialization for adaptive neural network-based fuzzy inference system to generate the final rules based on which the FIS will be trained.

4.2 Rule learning

Based on the MFs initialized in the previous step, the initial gradient, search direction and training parameters will be estimated. As mentioned earlier, linguistic hedges are applied to the fuzzy sets in the rules and are adapted using the scaled conjugate gradient (SCG) algorithm [15,52]. The SCG algorithm is used to determine the optimum values of nonlinear parameters. The SCG is faster than the steepest descent and some second-order derivative-based methods. Also, it is suitable for large-scale problems. Minimization by gradient descent is based on the linear approximation $E(w + y) \approx E(w) + E'(w)^T y$, which is the main reason why the algorithm often shows poor convergence. Another reason is that the algorithm uses constant step size, which, in many cases, is inefficient and makes the algorithm less robust. Including a momentum term in the back propagation algorithm is an attempt in an ad hoc fashion to force the algorithm to use second-order information from the network.

With this approach, some distinctive features are emphasized by power values, and some irrelevant features are eliminated. The power effects in any feature are generally different for different classes. During the training process, these parameters are tuned until the recognition rate converges to a certain value or the maximum number of iterations is achieved. The final parameters of all MFs will be employed to generate a FIS. This inference system is composed of a number of rules which will later be used for prediction of test labels.

4.3 Fuzzy classifier

In this phase, we employ a neuro-fuzzy classification system, which is based on fuzzy rules and has been successfully applied to various classification tasks [75]. Neuro-fuzzy

classifiers define the class distributions and learn by training on the input–output relations, whereas fuzzy systems describe the system using natural language. Neural networks are employed for tuning or training the system parameters in neuro-fuzzy applications. Fuzzy classification systems, which are based on fuzzy rules, have been successfully applied to various classification tasks. A common method being used for these kinds of applications is the adaptive neuro-fuzzy inference system (ANFIS) which is basically a function approximator. The usage of ANFIS [28,40,41] for classifications, however, is unfavorable. For example, having a dataset with three classes labeled as 1, 2 and 3, the ANFIS outputs are not integers. For that reason, the ANFIS outputs are rounded and used to determine the class labels. But the problem is that sometimes ANFIS can give 0 or 4 as class labels. These cases are clearly not acceptable. Therefore, we employed an adaptive neuro-fuzzy classifier. An adaptive neuro-fuzzy classifier (ANFC) consists of input, membership function, fuzzification, defuzzification, normalization and output layers.

We employ ANFC with linguistic hedges (LH) [15] since it is able to deal with linguistic hedge values such as words in our case. The LHs that are constructed using the power of fuzzy sets introduce the importance of the fuzzy sets for fuzzy rules. They can also change the primary meaning of fuzzy membership functions to a secondary meaning. For a continuous linguistic term, A for input variable x , the modified version fuzzy set would be: $A^s := \{(x, (\mu_A(x))^p) | x \in X\}$ where p denotes the linguistic hedge value of A.

ANFC is based on fuzzy rules. Linguistic hedges are applied to the fuzzy sets of rules and are adapted by the SCG algorithm, along with other network parameters. The tuned LH values of fuzzy sets improve the flexibility of fuzzy sets which has a huge impact on the recognition rate, especially in case of overlapping classes.

A fuzzy classification rule that has two inputs X_1, X_2 and one output y is defined with LHs as: IF X_1 is A_1 with p_1 hedge AND X_2 is A_2 with p_2 hedge THEN y is c_1 class. A recommendation example is IF user_age is low AND event_popularity is high THEN class_label is interested class. A_1 and A_2 denote linguistic terms that are defined on the X_1 and X_2 feature spaces; p_1 and p_2 denote linguistic hedges, respectively; and c_1 denotes the class label of the output y . Figure 15 shows the adaptive neuro-fuzzy classifier with linguistic hedges (ANFCLH) architecture. The training data are categorized by three classes, c_1, c_2 and c_3 [15]. The feature space $X_1 \times X_2$ is separated into fuzzy regions. Each input is represented as three linguistic terms; thus, we have nine fuzzy rules. In this model, the nodes in the same layer have the same type of node function. Each node in Layer 1 is associated with a parameterized bell-shaped membership function. Although these initial membership functions are set heuristically and subjectively, they provide an easy interpre-

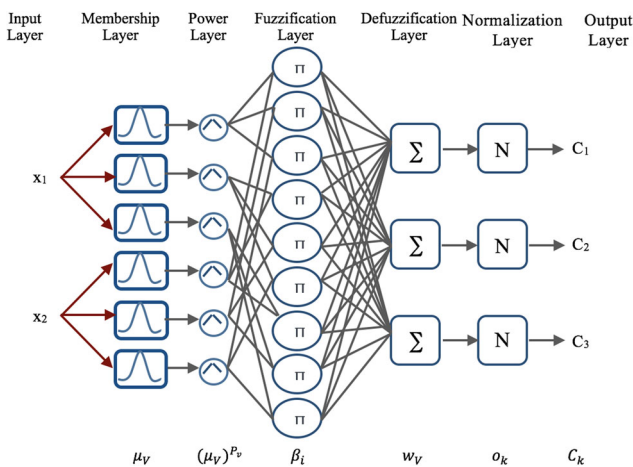


Fig. 15 Adaptive neuro-fuzzy classifier (ANFC) with linguistic hedges (LH) architecture [15]

tation parallel to human thinking. The parameters are then tuned with backpropagation, a gradient descent method, in the learning process based on a given training dataset. Each node in Layer 2 generates a signal corresponding to the conjunctive combination of individual degrees of matching. The linear combination of the firing strengths of the rules in Layer 3 is calculated, and a sigmoidal function in Layer 4 is applied to calculate a degree of belonging to a certain class. The crisp outputs of fuzzy rules are finally determined by a weighted average operator.

Our proposed algorithm can be summarized in the pseudocode in Algorithm 1.

Algorithm 1 Personalized Adaptive Neuro-Fuzzy event recommendation system with linguistic hedges.

Input: Training set ($X_{n \times m}$), Validation set ($V_{n' \times m}$), Testing set ($Y_{n' \times m}$), Selected features (M), Predefined number of clusters (K)

Output: Predicted Label ($L_{n'}$)

- 1: Initialize Membership Function set as empty, $MF \leftarrow \{\}$
- 2: Split X into two groups based on class labels, $G_i \subseteq X, i = 1, 2$
- 3: For each group $G_i, i = 1, 2$
- 4: Cluster G_i into K clusters, $C_{ij}, j = 1, \dots, k$, using the k -means algorithm $f_{K\text{-means}}$
- 5: For each cluster $C_{ij}, j = 1, \dots, k$
- 6: Create Gaussian membership function using (3): $\mu_{ijr}, r = 1, \dots, m$
- 7: Update $MF \leftarrow MF \cup \mu_{ijr}$
- 8: $f_{ANFC} \leftarrow$ Initialize ANFC algorithm (f_{ANFC}) with MF
- 9: $f_{ANFC} \leftarrow$ Update f_{ANFC} using validation set V
- 10: $L \leftarrow$ predict labels for testing set (Y) using f_{ANFC}
- 11: Return L

Table 5 Proportion of data in each class, before and after oversampling using SMOTE [16]

	Interested (%)	Not interested (%)
Before	89.0	11.0
After	50.1	49.9

5 Experimental results and discussion

The social event recommendation task is a relatively new subject in research. Hence, there are no official benchmark evaluation datasets. To the best of our knowledge, we used the largest dataset in event recommendation for evaluating our approach. The provided dataset² consists of 2034 users, 3055 events and 4645 reported user engagements in terms of interested/not interested, as listed in Table 1. As explained in section III, for each user–event pair, 16 attributes have been used. These attributes capture the information regarding users, events and their relations at different levels. In addition, we engineered seasonal and behavioral features which capture social activities. Since the attributes have different value ranges, we performed normalization on the whole dataset during preprocessing. After removing instances with unknown labels, there remained 4645 pairs of user–event which are labeled as interested or not interested, as listed in Table 2.

In order to overcome the problem of imbalanced data, we employed synthetic minority oversampling technique (SMOTE), which is a well-known resampling approach in machine learning [16]. SMOTE aims to oversample the minority class by creating synthetic examples. Table 5 shows the distribution of the classes before and after oversampling. To achieve a fair recommendation performance, we employed a modification of the tenfold cross-validation procedure. We split the data into three sets: training, validation and testing sets. After rotating these sets, 10 times, we take average on the testing set to obtain the recommendation performance. To handle missing values, we removed any user–event pair with missing values from the training set. In order to make our experiment identical to real-world systems, however, we leave missing values in the validation and test sets. Table 6 lists the size of training, validation and testing sets along with the proportion of their missing values.

As mentioned earlier, the proposed adaptive neuro-fuzzy event recommendation system consists of three main steps: (1) automatic membership functions (MFs) generation, (2) rule learning and (3) fuzzy classifier. In order to generate

² Since the competition has not released the testing set containing the ranked recommendation list, we cannot evaluate our method on the test set. Hence, we used only provided training set and split it into training, validation and testing set.

Table 6 Distribution of missing data

	Training set	Validation set	Testing set
#User–events	4121	2061	2061
Missing values%	1.0%	2.1%	2.0%

MFs, the training set is divided into two groups based on class labels. Then the k -means clustering algorithm is applied within each group with a predefined value for the number of clusters, k . We choose $k = 5$ to be able to show rule examples in this paper. It is obvious that larger clusters lead to more rules which cover the dataset better. This way, each feature dimension is partitioned into 5 homogeneously distributed overlapping regions. Analysis of the distribution of different features in each cluster for the interested and not interested classes shows that both classes have very similar feature distribution. This leads to an even more challenging classification model. These clusters are used to create Gaussian membership function sets, which will be employed as an initialization step for ANFCLH. The membership degree of each input to the specified fuzzy region is measured as mentioned before.

The ANFC model is trained using training and validation sets in 2 steps: (1) updating MFs and (2) extracting fuzzy rules. The training set is used to train the ANFC, whereas the validation dataset is used to verify the accuracy and effectiveness of the trained ANFC. At each epoch, the model's generality is checked in order to avoid overfitting. The fuzzy rule viewer of the established model is shown in Fig. 16 which displays two sets of fuzzy rules, generated by the classifier in different data folds. We applied k -means ($k = 5$) within each class of interested/not interested; hence, there are 10 final rules for each set. The last column in each figure shows the class labels generated for each cluster, where value 1 and 2 indicate interested and not interested, respectively. Each rule creates a structure for the predicted behavior of users based on the user and event attributes.

5.1 From classifier models to linguistic expression

In this section, we illustrate the powerful interpretability that is allowed by using fuzzy approximate reasoning thanks to the human readable linguistic expressibility of the discovered rules.

The following example represents a sample IF–THEN fuzzy rule generated by ANFC using fold 1 of the training set (Rule #2 in Fig. 16a).

“IF user_event_time_difference is very low **AND** event_is_Weekend is False **AND** event_quarter is winter **AND** creator_is_friend is false **AND** distance is close **AND** event_popularity is high **AND** user_count_of_invited_friends

is very low **AND** user_count_of_interested_friends is low **AND** user_age is very young, **AND** user_gender is Male **AND** w2_count is very low **AND** w3_count is very low **AND** w5_count is very low **AND** w7_count is very low **AND** w11_count is very low **THEN** label is interested”.

In plain English, the rule says: “*On a winter weekday, a very young male gets an invitation for a popular event in which a few of his friends are interested. The event is close to his place and it takes place in a few hours. So he would be interested to go to the event*”. Similarly, we interpret another IF–THEN fuzzy rule shown in the same Fig. 16b.

“IF user_event_time_difference is very low **AND** event_is_weekend is False **AND** event_quarter is winter **AND** creator_is_friend is false **AND** distance is close **AND** event_popularity is low **AND** user_count_of_invited_friends is very low **AND** user_count_of_interested_friends is very low **AND** user_age is young, **AND** user_gender is Female **AND** w2_count is very low **AND** w3_count is very low **AND** w5_count is very low **AND** w7_count is very low **AND** w11_count is very low **THEN** label is not interested”.

This rule tells a different story: “*It is a Winter weekday! A young woman finds out about an event happening in her neighborhood. Although the event takes place in a few hours, there are only a few users who showed interest in it, none of which are her friends. So, she will not be interested to go either*”.

These two stories that were generated from fuzzy linguistic rules represent valuable information as they are understandable interpretations of the proposed event recommendation system mechanism. In contrast, most existing approaches do not convey their inner mechanism, in such an easy to understand manner.

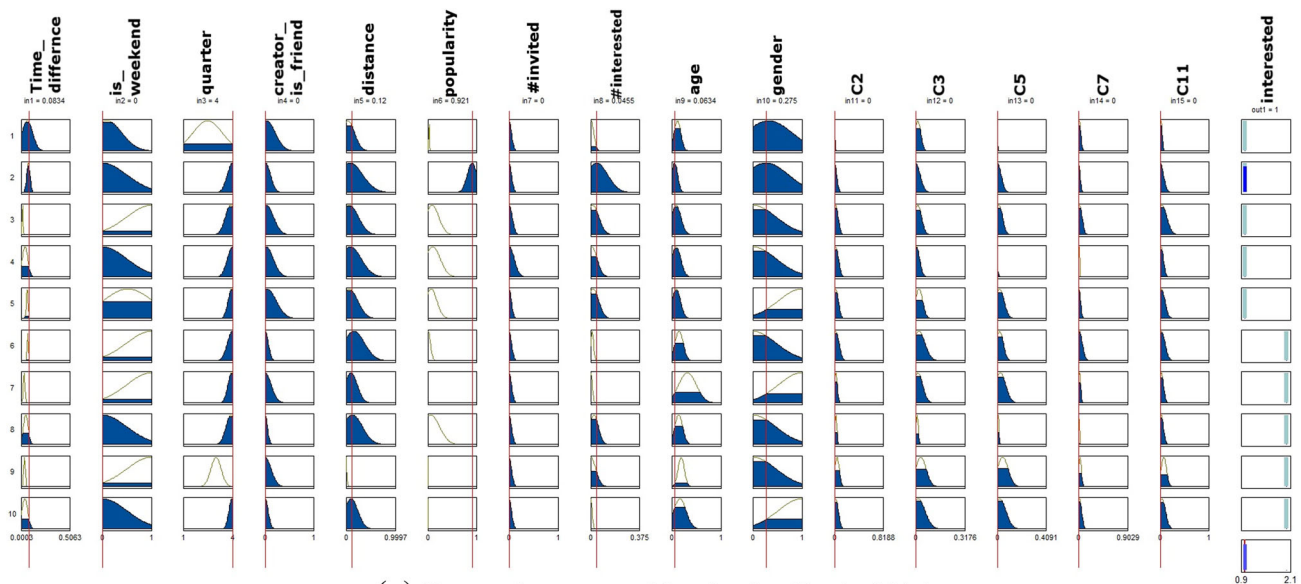
5.2 Evaluation metrics

We framed the event recommendation problem as a classification problem, and hence, we employ four classification measures to evaluate our proposed framework: accuracy, precision, recall and F -measure [5, 19]. These metrics are widely used in recommender systems to evaluate the quality of the recommendation and classification process.

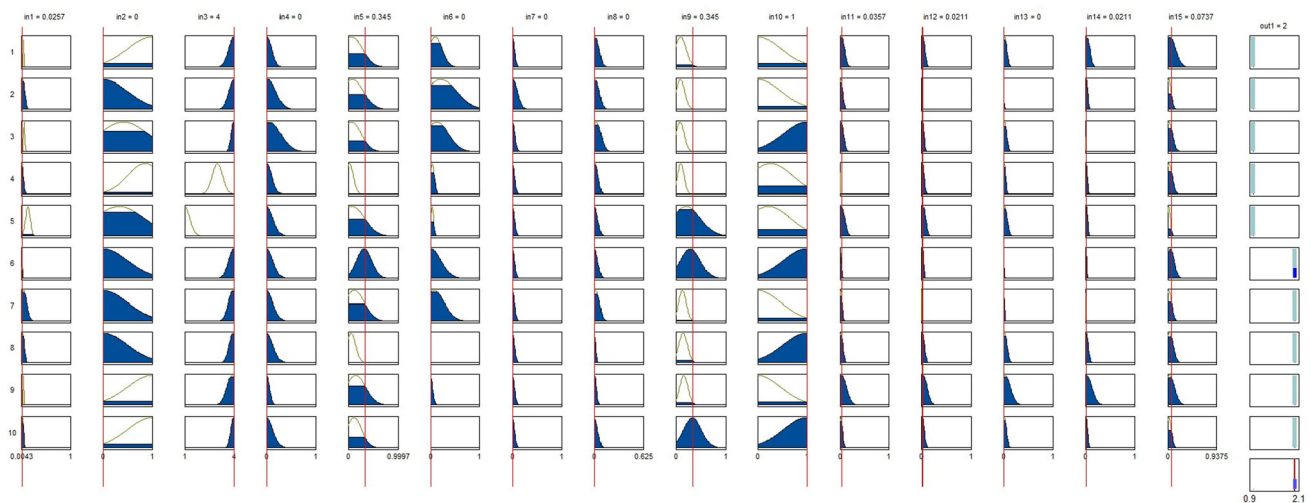
Accuracy is a statistical measure to evaluate how well a classifier works. It is defined as the ratio of true results.

$$\text{Accuracy} = \frac{(\text{TR} + \text{TN})}{(\text{TR} + \text{FR} + \text{TN} + \text{FN})} \quad (4)$$

where TR is the number of events recommended as relevant and that are really relevant, and TN is the number of events considered as non-relevant when they are truly non-relevant. FR is the number of events recommended as relevant although they are not, and FN is the number of events considered as non-relevant although they are relevant.



(a) Fuzzy rules generated by the classifier in fold 1



(b) Fuzzy rules generated by the classifier in fold 2

Fig. 16 Fuzzy rules (one rule per row) showing the input and output variables (one per column) of the ANFC model. **a** Fuzzy rules generated by the classifier in fold 1. **b** Fuzzy rules generated by the classifier in fold 2

Precision represents the probability that a truly relevant event has been recommended (TR). This metric is defined as the ratio of truly relevant events among all the recommended events, i.e.,

$$\text{Precision} = \frac{\text{TR}}{(\text{TR} + \text{FR})} \quad (5)$$

Recall measures the probability that a relevant event will be recommended, i.e.,

$$\text{Recall} = \frac{\text{TR}}{(\text{TR} + \text{FN})} \quad (6)$$

Equation 7 shows the *F*-measure which captures the harmonic mean of precision and recall.

$$F1 = \frac{2 \times (\text{precision} \times \text{recall})}{(\text{precision} + \text{recall})} \quad (7)$$

5.3 Comparison of recommendation methods

In this section, we compare the performance of the proposed framework to several baseline methods³ including *K*-nearest neighbors (KNN), support vector machines (SVM) and Naive

³ To our best knowledge, there is no comprehensive paper for the winner solution of this Kaggle competition. Due to these facts, we used only the

Fig. 17 Comparison of different methods using the complete test data, including missing data

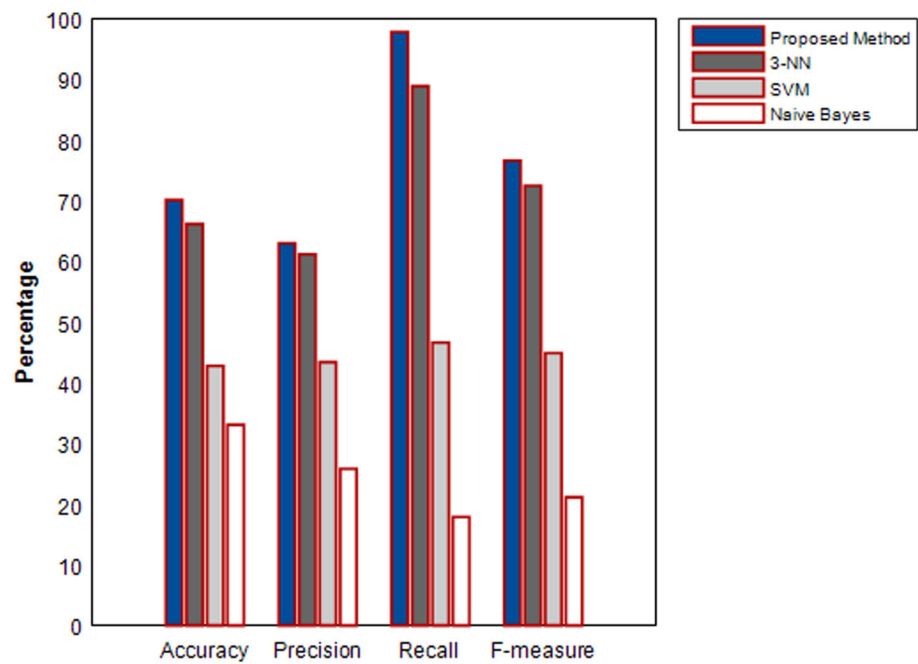


Table 7 *P* value and effect size of comparison of proposed approach with 3-NN, SVM and Naive Bayes

Tested hypothesis	Proposed versus 3NN	Proposed versus SVM	Proposed versus Naive Bayes
<i>P</i> value	0.037	0.001	0.001
Effect size	2.1	8.21	16.41

Bayes. KNN is a simple yet powerful machine learning approach that has been widely used for collaborative filtering recommender systems [67]. KNN classifies an event by a majority vote of its k nearest neighbors. SVM [9] is one of the well-used classifiers in machine learning, where it is an effective approach in high-dimensional space. This method finds an optimal hyperplane which categorizes the input data. Naive Bayes [74] is a probabilistic classifier which is based on Bayes theorem with strong (naive) independence assumptions between every pair of features. Nevertheless, naive Bayes classifiers have worked quite well in many real-world situations such as recommendation systems.

In order to have a fair comparison, we use the same cross-validation folds for all algorithms and report the average of the evaluation metrics. Figure 17 shows the accuracy of the proposed event recommendation pipeline compared to the aforementioned algorithms. The proposed approach outperforms other approaches with an average accuracy of 70%. We applied KNN with different values of k in which $k = 3$ achieved the highest accuracy. All algorithms are evaluated on the same testing sets containing missing values which is

a real-life scenario in recommender systems, as the provided information is often incomplete.

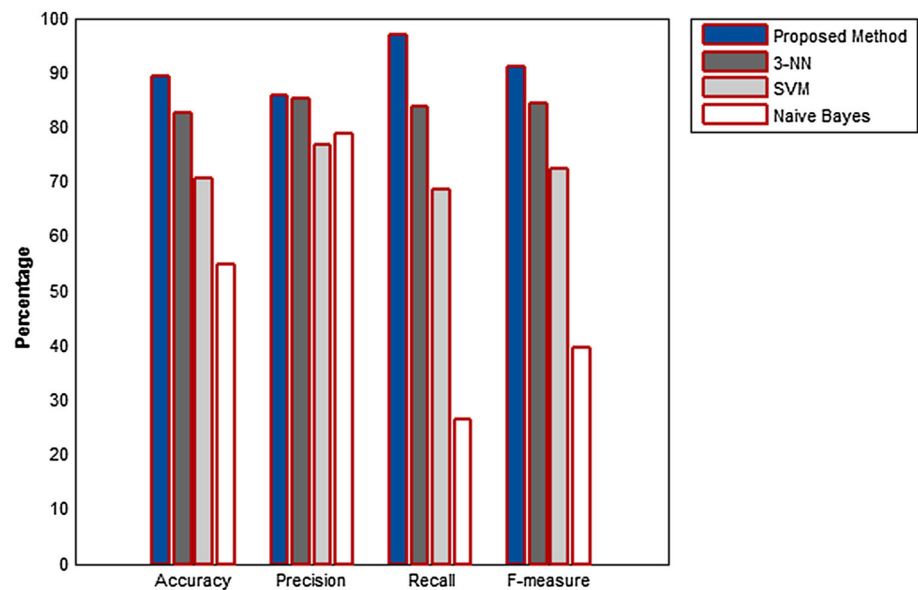
To assess the significance of the superiority of our event recommendation framework, we computed both *p* value [35] and Cohen's *d* effect size [53,76], considering 10 experiments. We tested the null hypothesis that there are no significant differences between each pair of algorithms [26,77]. We employ paired *t* tests to compare each pair of algorithms as their distribution is normal. Both metrics confirm the superiority of the proposed approach. As shown in Table 7, all the *p* values are less than 0.05, leading us to reject the null hypothesis that the differences between two approaches are not significant, with 95% confidence interval. The conclusion is that the differences are significant and our proposed approach outperforms other algorithms. All effect sizes are also more than 0.5 which shows a large difference effect and confirms the significant difference between the investigated approaches.

In addition, Fig. 17 indicates the superiority of our proposed approach compared to other baseline methods in terms of other evaluation metrics, namely precision, recall and *F*-measure.

We perform another experiment to analyze the impact of missing values. We remove any user–event pair with missing values from the test set. Each algorithm is trained on the same

available dataset and compared our model with the available baseline methods, using recommendation evaluation metrics.

Fig. 18 Comparison of different methods using the test set without missing data



training set as the previous step; however, they are tested on a new complete but smaller testing set. As we can see in Fig. 18, removing missing values results in a 28% increase in the accuracy of the proposed event recommender system. Although the rest of the algorithms also have performed better using the new testing set, the proposed approach still achieves a highest performance.

One of the crucial challenges of recommender systems is their ability to handle missing values since real-world data are inconsistent and incomplete. To investigate this issue in more depth, we consider a block of testing set with full values. Then, we gradually select $p\%$ of the data and hide their true values. As we can see in Fig. 19, adding more missing values causes a drastic decrease in the accuracy of the different approaches. However, our proposed event recommendation system is more robust to sparse data. In other words, the proposed recommendation framework can handle users or events with little or limited information. In the most extreme scenario, the proposed approach is able to make a relevant recommendation for a new user who has just entered the system (cold start). In addition, it will also predict if a newly created event will be favored by users or not.

In another direction, one may prioritize any of the suggested benchmarks considering the computational complexity. However, we emphasize that an important purpose of recommender systems and in the bigger picture, data science and data mining, is the ability to make an interpretation of the predictions, which is missing or harder to accomplish in the reviewed methods.

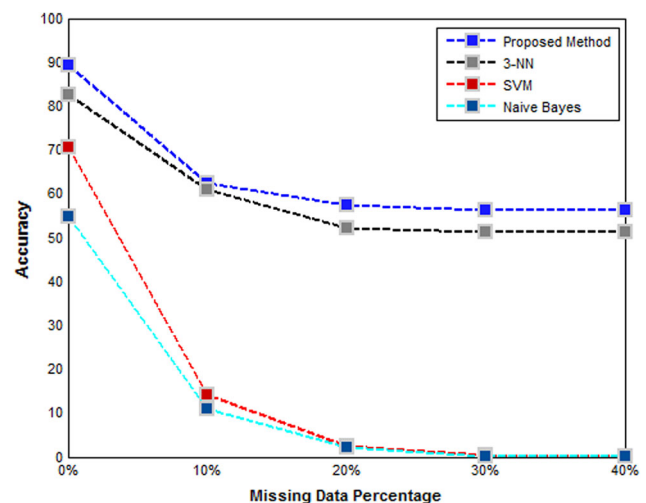


Fig. 19 Effect of missing data on the different algorithms

6 Conclusion

We presented a case study based on a framework to analyze real-life event data on social networks and recommend relevant events to users with different preferences. As part of the proposed pipeline, we conducted a thorough data analysis and exploration, to gain a better understanding of the data and shed light on the most practical ways to handle different issues such as combining several datasets, as well as handling a high proportion of missing values, missing labels and imbalanced data. After achieving a good understanding of the data during the exploratory phase, we proceeded to the next critical phase of feature engineering, storytelling and predictive modeling for computing the event recommendations for new users and events.

We have illustrated the power of a particularly desirable property of fuzzy sets, which is their rich linguistic approximate reasoning ability. This characteristic allows handling noisy and uncertain big data, while still allowing eloquent linguistic expressions from the computed accurate predictive models. These linguistic expressions of the models can provide a desirable interpretability that is critical in the data science field because data science often requires interdisciplinary team collaboration and yields results that need to be consumed by people of diverse technical and non-technical backgrounds, who therefore question the meaning of models and emphasize the importance of telling stories from the data. Event recommendations have a social impact since they play an important role in engaging users in activities. This can help engage users to get out of the virtual world and meet and connect with others who share similar interests and sometimes allow them to rally around causes that are critical to society such as environmental activism, advocacy and self-help groups. Event recommendations can thus help foster active, connected communities.

To conclude, our case study showcased a complete end-to-end pipeline that illustrated the superior performance of fuzzy approximate reasoning compared to competitive baselines such as SVM and K -NN predictions, without sacrificing in interpretability. Being able to express and interpret predictive models makes the results immediately meaningful within the context of the application domain, where the domain experts or the users of the recommender system, who are often not data scientists, can make sense of the recommendations without any need for technical expert translations.

References

1. Abdollahi, B., Badami, M., Nutakki, G.C., Sun, W., Nasraoui, O.: A two step ranking solution for twitter user engagement. In: Proceedings of the 2014 Recommender Systems Challenge, p. 35. ACM (2014)
2. Abdollahi, B., Nasraoui, O.: Using explainability for constrained matrix factorization. In: Proceedings of the Eleventh ACM Conference on Recommender Systems, pp. 79–83. ACM (2017)
3. Adomavicius, G., Tuzhilin, A.: Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *IEEE Trans. Knowl. Data Eng.* **17**(6), 734–749 (2005)
4. Baeza-Yates, R.: Data and algorithmic bias in the web. In: Proceedings of the 8th ACM Conference on Web Science, pp. 1–1. ACM (2016)
5. Bagherifard, K., Nilashi, M., Ibrahim, O., Ithnin, N., Nojeem, L.A., et al.: Measuring semantic similarity in grids using ontology (2013)
6. Baltrunas, L., Ludwig, B., Peer, S., Ricci, F.: Context relevance assessment and exploitation in mobile recommender systems. *Pers. Ubiquit. Comput.* **16**(5), 507–526 (2012)
7. Bobadilla, J., Ortega, F., Hernando, A., Gutiérrez, A.: Recommender systems survey. *Knowl. Based Syst.* **46**, 109–132 (2013)
8. Bodapati, A.V.: Recommendation systems with purchase data. *J. Mark. Res.* **45**(1), 77–93 (2008)
9. Burges, C.J.: A tutorial on support vector machines for pattern recognition. *Data Min. Knowl. Disc.* **2**(2), 121–167 (1998)
10. Cao, Y., Li, Y.: An intelligent fuzzy-based recommendation system for consumer electronic products. *Expert Syst. Appl.* **33**(1), 230–240 (2007)
11. Carmagnola, F., Cena, F., Console, L., Cortassa, O., Gena, C., Goy, A., Torre, I., Toso, A., Vernerio, F.: Tag-based user modeling for social multi-device adaptive guides. *User Model. User Adap. Inter.* **18**(5), 497–538 (2008)
12. Carrer-Neto, W., Hernández-Alcaraz, M.L., Valencia-García, R., García-Sánchez, F.: Social knowledge-based recommender system. Application to the movies domain. *Expert Syst. Appl.* **39**(12), 10990–11000 (2012)
13. Celma, Ò., Serra, X.: Foaming the music: bridging the semantic gap in music recommendation. *Web Semant.* **6**(4), 250–256 (2008)
14. Cena, F., Likavec, S., Lombardi, I., Picardi, C.: Should i stay or should i go? Improving event recommendation in the social web. *Interact. Comput.* **28**(1), 55–72 (2016)
15. Cetişli, B., Barkana, A.: Speeding up the scaled conjugate gradient algorithm and its application in neuro-fuzzy classifier training. *Soft. Comput.* **14**(4), 365–378 (2010)
16. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: Smote: synthetic minority over-sampling technique. *J. Artif. Intell. Res.* **16**, 321–357 (2002)
17. Chen, C.C., Sun, Y.C.: Exploring acquaintances of social network site users for effective social event recommendations. *Inf. Process. Lett.* **116**(3), 227–236 (2016)
18. Cheng, C., Yang, H., King, I., Lyu, M.R.: Fused matrix factorization with geographical and social influence in location-based social networks. *Aaai* **12**, 17–23 (2012)
19. Cho, J., Kwon, K., Park, Y.: Collaborative filtering using dual information sources. *IEEE Intell. Syst.* **22**(3), 30–38 (2007)
20. Chou, S.Y., Yang, Y.H., Jang, J.S.R., Lin, Y.C.: Addressing cold start for next-song recommendation. In: Proceedings of the 10th ACM Conference on Recommender Systems, pp. 115–118. ACM (2016)
21. Chua, T., Tan, W.: A new fuzzy rule-based initialization method for k -nearest neighbor classifier. In: IEEE International Conference on Fuzzy Systems, 2009. FUZZ-IEEE 2009, pp. 415–420. IEEE (2009)
22. Cornelis, C., Guo, X., Lu, J., Zhang, G.: A fuzzy relational approach to event recommendation. *IICAI* **5**, 2231–2242 (2005)
23. Cornelis, C., Lu, J., Guo, X., Zhang, G.: One-and-only item recommendation with fuzzy logic techniques. *Inf. Sci.* **177**(22), 4906–4921 (2007)
24. Cortes, C., Vapnik, V.: Support-vector networks. *Mach. Learn.* **20**(3), 273–297 (1995)
25. Cover, T., Hart, P.: Nearest neighbor pattern classification. *IEEE Trans. Inf. Theory* **13**(1), 21–27 (1967)
26. Demšar, J.: Statistical comparisons of classifiers over multiple data sets. *J. Mach. Learn. Res.* **7**(Jan), 1–30 (2006)
27. Dhar, V.: Data science and prediction. *Commun. ACM* **56**(12), 64–73 (2013)
28. Di Noia, T., Mirizzi, R., Ostuni, V.C., Romito, D., Zanker, M.: Linked open data to support content-based recommender systems. In: Proceedings of the 8th International Conference on Semantic Systems, pp. 1–8. ACM (2012)
29. Du, R., Yu, Z., Mei, T., Wang, Z., Wang, Z., Guo, B.: Predicting activity attendance in event-based social networks: content, context and social influence. In: Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing, pp. 425–434. ACM (2014)
30. Duda, R.O., Hart, P.E., Stork, D.G., et al.: Pattern Classification, vol. 2. Wiley, New York (1973)

31. Estabrooks, A., Jo, T., Japkowicz, N.: A multiple resampling method for learning from imbalanced data sets. *Comput. Intell.* **20**(1), 18–36 (2004)
32. Feng, S., Li, X., Zeng, Y., Cong, G., Chee, Y.M., Yuan, Q.: Personalized ranking metric embedding for next new poi recommendation. In: *IJCAI*, pp. 2069–2075 (2015)
33. Fränti, P., Chen, J., Tabarcea, A.: Four aspects of relevance in sharing location-based media: content, time, location and network. In: *WEBIST*, pp. 413–417 (2011)
34. Guo, B., Yu, Z., Chen, L., Zhou, X., Ma, X.: Mobigroup: enabling lifecycle support to social activity organization and suggestion with mobile crowd sensing. *IEEE Trans. Hum. Mach. Syst.* **46**(3), 390–402 (2016)
35. Hsu, H., Lachenbruch, P.A.: Paired t Test. *Wiley Encyclopedia of Clinical Trials* (2008)
36. Huang, H.: *Neuro-Fuzzy and Soft Computing—A Computational Approach to Learning and Machine Intelligence*. Prentice-Hall, Englewood Cliffs, NJ (2016)
37. Jang, J.S.R., Sun, C.T., Mizutani, E.: *Neuro-fuzzy and soft computing; a computational approach to learning and machine intelligence* (1997)
38. Jiang, J.Y., Li, C.T.: Analyzing social event participants for a single organizer. In: *ICWSM*, pp. 599–602 (2016)
39. Kaggle: Event recommendation engine challenge. <https://www.kaggle.com/c/event-recommendation-engine-challenge> (2013). Accessed 1 March 2013
40. Kayaalp, M., Özyer, T., Özyer, S.T.: A collaborative and content based event recommendation system integrated with data collection scrapers and services at a social networking site. In: *International Conference on Advances in Social Network Analysis and Mining*, 2009. *ASONAM'09*, pp. 113–118. IEEE (2009)
41. Khrouf, H., Troncy, R.: Hybrid event recommendation using linked data and user diversity. In: *Proceedings of the 7th ACM Conference on Recommender Systems*, pp. 185–192. ACM (2013)
42. Klir, G., Yuan, B.: *Fuzzy Sets and Fuzzy Logic*, vol. 4. Prentice Hall, New Jersey (1995)
43. Krishnapuram, R., Joshi, A., Nasraoui, O., Yi, L.: Low-complexity fuzzy relational clustering algorithms for web mining. *IEEE Trans. Fuzzy Syst.* **9**(4), 595–607 (2001)
44. Lilien, G.L., Kotler, P., Moorthy, K.S.: *Marketing Models*. Prentice Hall, New Jersey (1992)
45. Liu, B., Xiong, H., Papadimitriou, S., Fu, Y., Yao, Z.: A general geographical probabilistic factor model for point of interest recommendation. *IEEE Trans. Knowl. Data Eng.* **27**(5), 1167–1179 (2015)
46. Liu, X., He, Q., Tian, Y., Lee, W.C., McPherson, J., Han, J.: Event-based social networks: linking the online and offline social worlds. In: *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1032–1040. ACM (2012)
47. Macedo, A.Q., Marinho, L.B., Santos, R.L.: Context-aware event recommendation in event-based social networks. In: *Proceedings of the 9th ACM Conference on Recommender Systems*, pp. 123–130. ACM (2015)
48. Magnuson, A., Dialani, V., Mallela, D.: Event recommendation using twitter activity. In: *Proceedings of the 9th ACM Conference on Recommender Systems*, pp. 331–332. ACM (2015)
49. Meetup: About meetup. <https://www.meetup.com/about/> (2017). Accessed 21 March 2017
50. Millennials: Fueling the experience economy. <https://www.eventbrite.com/blog/academy/millennials-fueling-experience-economy/>. Accessed 21 Jan 2018
51. Minkov, E., Chawrow, B., Ledlie, J., Teller, S., Jaakkola, T.: Collaborative future event recommendation. In: *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, pp. 819–828. ACM (2010)
52. Møller, M.F.: A scaled conjugate gradient algorithm for fast supervised learning. *Neural Netw.* **6**(4), 525–533 (1993)
53. Mondloch, M.V., Cole, D.C., Frank, J.W.: Does how you do depend on how you think you'll do? A systematic review of the evidence for a relation between patients' recovery expectations and health outcomes. *Can. Med. Assoc. J.* **165**(2), 174–179 (2001)
54. Nanopoulos, A., Rafailidis, D., Symeonidis, P., Manolopoulos, Y.: Musicbox: personalized music recommendation based on cubic analysis of social tags. *IEEE Trans. Audio Speech Lang. Process.* **18**(2), 407–412 (2010)
55. Nasraoui, O.: Tell me why? Tell me more! explaining predictions, iterated learning bias, and counter-polarization in big data discovery models. In: *CCS@Lexington*. University of Kentucky (2017)
56. Nasraoui, O., Krishnapuram, R., Joshi, A.: Mining web access logs using a fuzzy relational clustering algorithm based on a robust estimator. In: *Proceedings of the Eighth International World Wide Web Conference*, Toronto, Canada (1999)
57. Nasraoui, O., Krishnapuram, R., Joshi, A.: Relational clustering based on a new robust estimator with application to web mining. In: *Fuzzy Information Processing Society, 1999. NAFIPS. 18th International Conference of the North American*, pp. 705–709. IEEE (1999)
58. Nasraoui, O., Petenes, C.: An intelligent web recommendation engine based on fuzzy approximate reasoning. In: *The 12th IEEE International Conference on Fuzzy Systems*, 2003. *FUZZ'03*, vol. 2, pp. 1116–1121. IEEE (2003)
59. Nasraoui, O., Zhang, Z., Saka, E.: Web recommender system implementations in multiple flavors: fast and (care) free for all. In: *SIGIR Open Source Information Retrieval Workshop*, pp. 46–53. Citeseer (2006)
60. Natekin, A., Knoll, A.: Boosting simplified fuzzy neural networks. In: *International Conference on Engineering Applications of Neural Networks*, pp. 330–339. Springer (2013)
61. Núñez-Valdéz, E.R., Lovelle, J.M.C., Martínez, O.S., García-Díaz, V., De Pablos, P.O., Marín, C.E.M.: Implicit feedback techniques on recommender systems applied to electronic books. *Comput. Hum. Behav.* **28**(4), 1186–1193 (2012)
62. Nutakki, G.C., Nasraoui, O., Abdollahi, B., Badami, M., Sun, W.: Distributed LDA-based topic modeling and topic agglomeration in a latent space. In: *SNOW-DC@ WWW*, pp. 17–24 (2014)
63. Odić, A., Tkalič, M., Tasič, J.F., Košir, A.: Predicting and detecting the relevant contextual information in a movie-recommender system. *Interact. Comput.* **25**(1), 74–90 (2013)
64. Ogundele, T.J., Chow, C.Y., Zhang, J.D.: Eventrec: Personalized event recommendations for smart event-based social networks. In: *2017 IEEE International Conference on Smart Computing (SMARTCOMP)*, pp. 1–8. IEEE (2017)
65. Perny, P., Zucker, J.D.: Collaborative filtering methods based on fuzzy preference relations. *Proc. EUROFUSE-SIC* **99**, 279–285 (1999)
66. Perny, P., Zucker, J.D.: Preference-based search and machine learning for collaborative filtering: the film-conseil movie recommender system. *Inf. Interact. Intell.* **1**(1), 9–48 (2001)
67. Peterson, L.E.: K -nearest neighbor. *Scholarpedia* **4**(2), 1883 (2009)
68. Pinckney, T., Dixon, C., Gattis, M.R.: Inferring user preferences from an internet based social interactive construct (2017). US Patent 9,754,308
69. Qiao, Z., Zhang, P., Cao, Y., Zhou, C., Guo, L., Fang, B.: Combining heterogeneous social and geographical information for event recommendation. In: *AAAI*, pp. 145–151 (2014)
70. Qiao, Z., Zhang, P., Zhou, C., Cao, Y., Guo, L., Zhang, Y.: Event recommendation in event-based social networks (2014)
71. Rich, E.: User modeling via stereotypes. *Cogn. Sci.* **3**(4), 329–354 (1979)
72. Rodríguez, A.C., Rorís, V.M.A., Gago, J.M.S., Rifón, L.E.A., Iglesias, M.J.F.: Providing event recommendations in educational sce-

- narios. In: Management Intelligent Systems, pp. 91–98. Springer (2013)
73. Rosaci, D., Sarné, G.M.: A multi-agent recommender system for supporting device adaptivity in e-commerce. *J. Intell. Inf. Syst.* **38**(2), 393–418 (2012)
 74. Russell, S., Norvig, P., Intelligence, A.: A modern approach. *Artificial Intelligence*. Prentice-Hall, Englewood Cliffs **25** (1995)
 75. Salton, G.: Automatic Text Processing: The Transformation, Analysis, and Retrieval of. Addison-Wesley, Reading (1989)
 76. Smith, M.H.: A sample/population size activity: is it the sample size of the sample as a fraction of the population that matters? *J. Stat. Educ.* **12**(2). <https://doi.org/10.1080/10691898.2004.11910735> (2004)
 77. Sullivan, G.M., Feinn, R.: Using effect size or why the p value is not enough. *J. Grad. Med. Educ.* **4**(3), 279–282 (2012)
 78. Tamayo, L.F.T.: Fuzzy logic. In: SmartParticipation, pp. 33–43. Springer (2014)
 79. Waga, K., Tabarcea, A., Fränti, P.: Context aware recommendation of location-based data. In: 2011 15th International Conference on System Theory, Control, and Computing (ICSTCC), pp. 1–6 (2011)
 80. Waller, M.A., Fawcett, S.E.: Data science, predictive analytics, and big data: a revolution that will transform supply chain design and management. *J. Bus. Logist.* **34**(2), 77–84 (2013)
 81. Wang, P., Guo, J., Lan, Y., Xu, J., Wan, S., Cheng, X.: Learning hierarchical representation model for nextbasket recommendation. In: Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 403–412. ACM (2015)
 82. Wang, S., Hu, L., Cao, L., Huang, X., Lian, D., Liu, W.: Attention-based transactional context embedding for next-item recommendation. In: AAAI (2018)
 83. Wu, D., Zhang, G., Lu, J.: A fuzzy preference tree-based recommender system for personalized business-to-business e-services. *IEEE Trans. Fuzzy Syst.* **23**(1), 29–43 (2015)
 84. Xin, X., King, I., Deng, H., Lyu, M.R.: A social recommendation framework based on multi-scale continuous conditional random fields. In: Proceedings of the 18th ACM Conference on Information and Knowledge Management, pp. 1247–1256. ACM (2009)
 85. Yager, R.R.: Fuzzy logic methods in recommender systems. *Fuzzy Sets Syst.* **136**(2), 133–149 (2003)
 86. Yang, Y., Pedersen, J.O.: A comparative study on feature selection in text categorization. *ICML* **97**, 412–420 (1997)
 87. Yuan, Q., Cong, G., Ma, Z., Sun, A., Thalmann, N.M.: Time-aware point-of-interest recommendation. In: Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 363–372. ACM (2013)
 88. Yuan, Z., Li, H.: Location recommendation algorithm based on temporal and geographical similarity in location-based social networks. In: 2016 12th World Congress on Intelligent Control and Automation (WCICA), pp. 1697–1702. IEEE (2016)
 89. Zadeh, L.A.: Fuzzy logic= computing with words. *IEEE Trans. Fuzzy Syst.* **4**(2), 103–111 (1996)
 90. Zenebe, A., Norcio, A.F.: Representation, similarity measures and aggregation methods using fuzzy sets for content-based recommender systems. *Fuzzy Sets Syst.* **160**(1), 76–94 (2009)
 91. Zenebe, A., Zhou, L., Norcio, A.F.: User preferences discovery using fuzzy models. *Fuzzy Sets Syst.* **161**(23), 3044–3063 (2010)
 92. Zhang, J.D., Chow, C.Y.: Core: exploiting the personalized influence of two-dimensional geographic coordinates for location recommendations. *Inf. Sci.* **293**, 163–181 (2015)
 93. Zhang, J.D., Chow, C.Y.: Geosoca: Exploiting geographical, social and categorical correlations for point-of-interest recommendations. In: Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 443–452. ACM (2015)
 94. Zhang, J.D., Chow, C.Y., Li, Y.: Lore: Exploiting sequential influence for location recommendations. In: Proceedings of the 22nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, pp. 103–112. ACM (2014)
 95. Zhang, J.D., Chow, C.Y., Li, Y.: iGeoRec: A personalized and efficient geographical location recommendation framework. *IEEE Trans. Serv. Comput.* **8**(5), 701–714 (2015)
 96. Zhang, Z., Nasraoui, O.: Mining search engine query logs for query recommendation. In: Proceedings of the 15th International Conference on World Wide Web, pp. 1039–1040. ACM (2006)