Turk J Elec Eng & Comp Sci
() : –
© TÜBİTAK
doi:10.3906/elk-

A new framework for the cold start problem in recommender systems

Enes ÇELİK[1,2,*], **Sevinç İLHAN OMURCA**[2]
[1]Department of Computer Science, Babaeski Vocational School, Kırklareli University, Kırklareli, Turkey,
ORCID iD: https://orcid.org/0000-0002-3282-865X
[2]Department of Computer Engineering, Faculty of Engineering, Kocaeli University, Kocaeli, Turkey,
ORCID iD: https://orcid.org/0000-0003-1214-9235

Received: .201 • **Accepted/Published Online:** .201 • **Final Version:** ..201

Abstract: In today's world, with rapidly developing technology, it has become possible to perform many transactions over the internet. In the meantime, the effects of many sectors such as commerce, television, journalism and even advertising in the online world have started to increase. As a result, providing better service to online customers in every field has become a crucial task. These developments have forced companies and sellers to recommend the right and suitable products to their customers. And finally, the recommender systems emerged as a field of study in order to ensure that the right and suitable products can be presented to the users. One of the biggest disadvantages of recommender systems is the cold start problem, which occurs when there is no information about a newly arrived user or product. For the solution to the cold start problem, the user needs to provide some external information about himself. For this reason, since social media is a common tool for all people today, in this study, we will focus on implicit knowledge that we can extract from users' social media data. In the proposed model, first, the behavioral profiles of the users are derived from the social media data of the users. Then, recommendation lists have been created for the cold start problem by implementing Boosting algorithms. Thus, a solution has been provided without requesting any external data other than social media information from the user for the cold start problem. Considering the experimental results, it has been observed that the proposed system gives better results than the current cold start problem solution proposals.

Key words: Recommender systems, cold start problem, social media, boosting algorithms, implicit knowledge

1. Introduction

With the rapid growth of the online world, there is a wide variety of options for users in all areas of interest. This situation is evaluated as a potential matter for many users. To solve this issue, it is necessary to reduce the users' options and present only the alternatives they are interested in or may be concerned about. In this case, the user's loyalty will ascend in the online world. Recommender systems aim to submit unique content and services to users in the online world. To do this, very different and huge data sources are used. Thus, it filters a wide variety of content in the online world for users and submits a solution to the problem by presenting only the items that will suit the user.

Recommender systems take various inputs, apply multiple transactions to them, and give tips to users using these converted values. Sometimes, it can make suggestions by evaluating the relationships between the items advised and the recommended for making the best matches [1]. Sometimes, recommender systems produce results by examining users' previous behavior. Recommender Systems are very salutary for users, companies, or sellers [2]. It shortens the process for users to access and make choices about items on an online movie platform.

*Correspondence: enes.celik@klu.edu.tr

Recommender systems have had widespread use today for reasons such as abbreviating the decision-making time of users, enabling customers to choose items by providing practical suggestions, boosting the profits of companies or sellers, and increasing the belonging of their users. Recommender systems have an extensive range of usage areas. It is actively used in fields such as E-commerce, music, Tv series, tourism, and digital libraries [3]. The general focus of recommender systems is to be able to suggest the right items to users. Recommender systems receive support from a variety of data to make advice. These data can be like publicly available evaluation data or obtained with the help of user relationships. In fact, information other than the user's knowledge of the platform to be recommended may also be used [4]. When there is no information belonging to a user in the system, historical data is not in question since the user does not make any grading when the user gets involved in the system for the first time. That is why it will be difficult for the recommender system to make inferences about such users and make offers [5]. In the same way, adding an item to the environment new, not having any history about this item will make it difficult to recommend the item to the right users. This issue is mainly known as the cold start problem in recommender systems. The above-mentioned external data for solving the cold start problem can be cited as a solution proposal [6]. The lack of ready-made data in the system where the user is located can allow for creating a user profile using external data, and various suggestions can be made according to this profile. Today, many websites consult various ways to use this and similar external data [7]. Examples of using external information are Twitter's request for users to provide information on topics they are interested in when they become a member and Netflix's request to users to choose three among ten popular movies at the time of the first membership. However, these external situations have several disadvantages. Asking users for this exterior information may not be suitable for users. Users may want to avoid answering these questions asked at the initial stage. Therefore, as we consider the power of social media, it can be seen that they are an excellent auxiliary resource for creating user profiles. For these exterior sources of information, it will be enough for users to remark only the addresses of their social media accounts. The fundamental contributions of this study can be summarized as follows:

- This model be used for "users" or "items" in recommendation systems
- It is not the particular solution to the cold start problem, and it can only be used to soften the problem
- It can be used immediately, without the need for extra data input from the user or wasting time
- The offered framework can be included in any recommender system that uses a social network
- The proposed model maintained confidentiality and ensured anonymity

The rest of the paper is organized as follows. In section 2 we present the related work in the cold start problem literature. In section 3 we describe in detail the proposed framework, how the dataset was constructed, how feature selection was made, classification algorithms, and evaluation metrics. Experimental findings are given in section 4. Section 5 concludes the present work and provides a direction for future works.

2. Related work

In the study [8], a model is advised using Cross-Level Association Rules to integrate content information about domain elements into collaborative filters to solve the cold start problem. A preference model which includes both user-item and item-item relationships has been introduced in recommender systems. In the study performed [9], a high-score recommender system has been developed by combining the items' attribute knowledge and the user's quality information to mitigate the cold start problem. A multidimensional matrix model has been proposed with the collaborative filtering algorithm. Also, helpful selection criteria have been designed and combined in an optimized framework in a cloud-based environment. In the study executed in [10], a unique method of constructing a model derived from detailed ratings has been proposed. The proposed method estimates the actual ratings and then defines and reduces the prediction errors for each user. A pre-calculated model with a collective error reflection is generated by taking advantage of this error information. In the study made in [11], existing approaches on Graph Neural Networks for users or items for the cold start problem, combining multiple properties equally, form the attribute placement of each node. Taking advantage of the proposed model feature weights and interactions between neighboring nodes, a new framework called Feature Importance, and Neighboring Node Interactions Graph Neural Network (FINI) is proposed. In the study conducted in [12], a new heuristic similarity measure has been presented that focuses on improving recommendation performance under cold start conditions, where only a small number of ratings are available that perform similarity calculations for each user. A new similarity measure perfected using optimization techniques based on Neural Learning has been presented in the study made in [13]. Based on the so-called ratings, a new proposal for a web service that can alleviate the cold start problem, integrating contextual information and an online learning model, has been submitted in the study [14]. Compared with the method without contextual information, the proposed approach to solving the cold start problem by integrating contextual information showed a high score ratio. In a study at [15], it has been shown that creating a tag set for the beginning with the association guideline and using various automatic filtering strategies produce efficient improvements. A new approach has been proposed that takes advantage of both positive and negative user feedback to recursive select entry tags and a genetic algorithm strategy to learn the recommender system. In the study performed in [16], the cold start problem arises from the lack of prior knowledge about new users and items in recommender systems. A hybrid approach combining collaborative filtering recommendations with demographic information is presented for this problem. The approach is based on the existing SCOAL (Simultaneous Co-Clustering and Learning) algorithm and has addressed the cold start problem, where no collaborative information is available for new users. The system produces reasonable estimates to correct the lack of information for the user. In [17]'s study, a social network that includes all users with a new collaborative filtering recommender system that offers soft ratings has been submitted. Presenting soft ratings introduces a new methodology for modeling subjective, qualitative, and imperfect information about user preferences. Additionally, the system is used to overcome the cold start problem and the infrequency of ratings with community preferences removed from the social network. In [18] work, a new hybrid approach is presented to alleviate the cold start problem in context-sensitive recommender systems. Hybrid model; The knowledge discovered by the community is combined into a rating system based on similarities, association rules, and probability criteria. In the study made in [19], a credit allocation algorithm grounded on a co-citation network is submitted, which captures the cold start problem, the co-credit of a multi-author article. It has been shown that the proposed method can be applied to academic papers at any period after publication with a higher degree of correctness than existing algorithms applied to new papers. In the study conducted at [20], a model has been advised for taking recommendations using similarity techniques and classification algorithms. The proposed approach was used to designate other users with similar behavior, for which demographic data were used by classification methods in the collaborative filtering system. According to the study carried out in [21], the Recommender System (RS- LOD) model with Linked Open Data was developed for the cold start problem, and the Matrix Factorization model (MF-LOD) with Linked Open Data was developed for the data sparsity problem. The LOD knowledge base is used to find sufficient information about the new user item in a cold start problem, and an improvement has been made to the matrix factorization model for data sparsity. In the study in [22], the cold start problem in recommender systems has traditionally been assumed to be relevant to most first-time users' interests based on popularity, timeliness, and positive ratings. In this study, it has been determined with heuristic algorithms that show users' consumption preferences are biased toward unpopular items. Two new recommender systems have been presented to alleviate this issue based on maximum user coverage; Maximum Coverage and Category-Exploration algorithms. To cope with the cold start problem, a new collaborative ranking model combining the Probabilistic Matrix Factorization (PMF) approach to rating and Bayesian Personalized Ranking (BPR) has been proposed in the [23]' study. The proposed model showed efficient results in explicit and implicit feedback data. In the study conducted in [24], a solution to the cold start problem has been suggested based on prediction with user profile classification algorithms from social media data. Decision Trees and Random Forest algorithms have been used in the classification. The study at [25] proposes a unified learning-based approach to address the cold start problem. A double of Deep Q Learning approaches based on the trust levels of the consultants are proposed to apply the composite learning specifically to the cold start problem and to obtain the confidence scores for potential consultants, then select the best candidates. The study in [26] proposes a hybrid model combining content-based filtering and Latent Dirichlet Allocation (LDA) based models. The proposed model solves the cold start problem since the words correspond to user or item characteristics and provide predicted ratings for new users or items through their hidden size.

3. Experimental setup

3.1. Proposed model

This study proposes a solution to the cold start problem by taking a tiny amount of data input from the user. The user has been requested to specify only the social media account username. By using this social media information, historical data belonging to the user is reached, and a special profile belonging to the user is created with the implicit information obtained from these historical data. Then, by using this profile, it is aimed to provide the most accurate estimates to the user by filtering the appropriate and inappropriate movies for the user from the list of movies we have. In order to do this, assistance is taken from Boosting algorithms with good performance in classification. In this section, the stages used to create a recommender system and the methods used in each process are stated in detail. Inputs of the the recommender system are some specialties that have been made meaningful from the users' social media data. These features are evaluated as profiles representing the personalities of the users. A classification problem is defined by using user profiles as inputs and recommendation results of the movies in the movie list as the prediction output. To determine the related movie list, we select the N movies that are rated the most by users with whom we have user profiles. In the results of this model, recommendable films with a high ratio for the user are collected and suggested to the user. The general structure of the proposed system is shown in Figure 1. Various classification studies have been conducted about recommender systems and cold start problems [32]. The Letterboxd site keeps users' rankings on a scale of 1-5. In this gap, the values include not only integer values such as 1, 2, 3, but also decimal values such as 2.5, 3.5 In order to convert the problem into a classification problem, evaluations above 3.5 can be positively recommendable, while assessments below 3.5 are tagged as negative and inadvisable. It is aimed to recommend more than one movie to users, not just a result in the figure that a movie about users can be recommended or not. It has a negative or positive label that may or may not be recommended for only one film by using each user profile. For this reason, N classification models have been created to represent N quantity films in each user profile. In this dataset, there are reviews of various users for several movies. Here, the first top 20 films with the most user ratings were used to produce the model. The literature assumes that users evaluate at least 20 films in standard data sets.

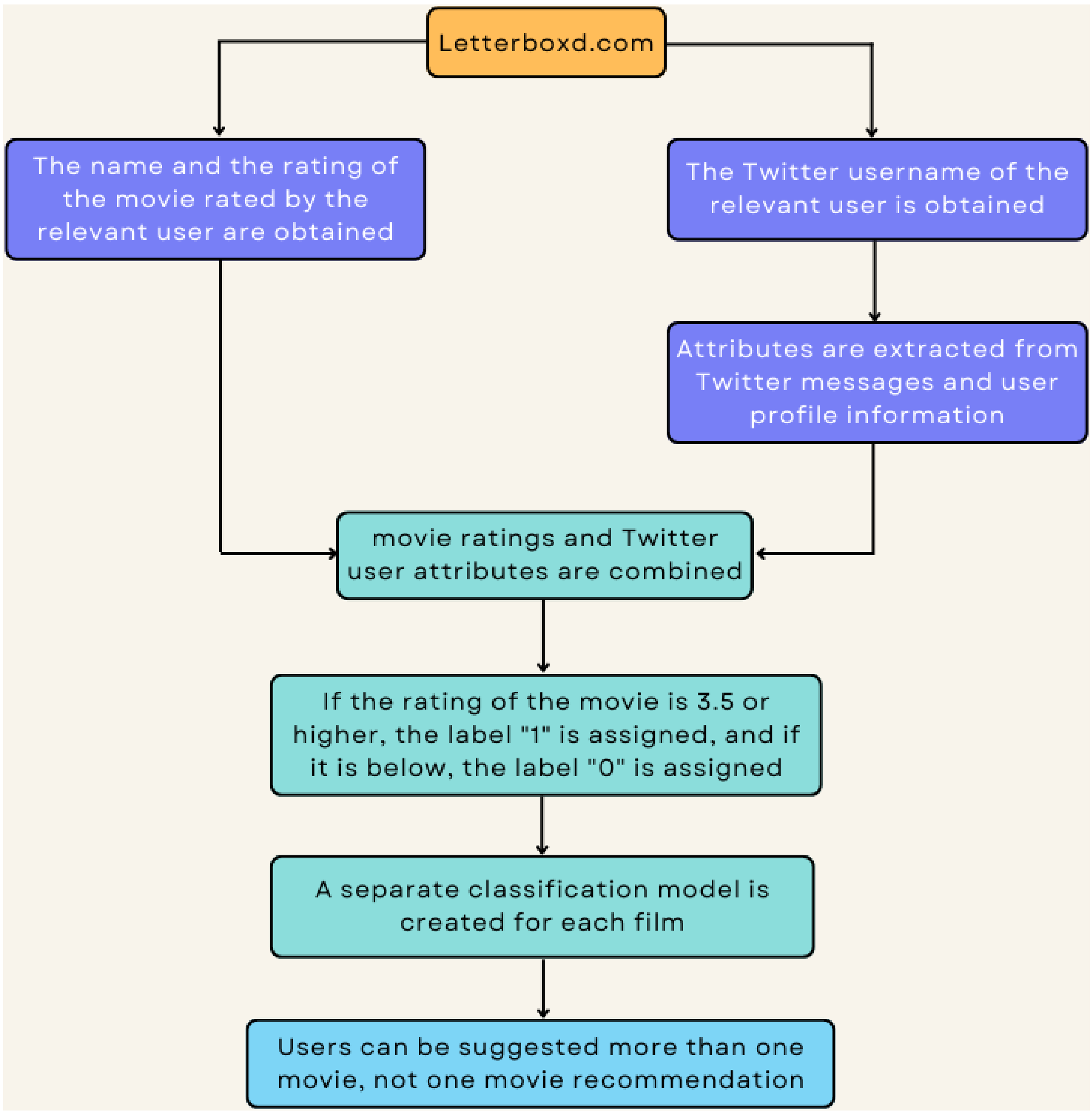

Figure 1. The general structure of the proposed model.

3.2. Dataset

Today, many websites have made reach to their data legal for scientific studies. These access permissions are given to people who supply various conditions or request access to data. However, some websites still need to support these access technologies in their infrastructure. It needs to collect data on websites into special databases to meet data needs. At this point, web scraping technology comes into play [15]. In this study, Letterboxd, the website where movie ratings are taken, was preferred, and a web service integration prepared for outside world users to access the data could not be found. For this reason, web scraping techniques have been used on this site to collect and process the data. With this technique, users' profiles on the website, Twitter user names, various movies, and ratings have been accessed.

In this study we have decided to use a new data set, entirely outside the previously used data sets. We have reviewed many online movie rating sites as we will use social media data in the cold start problem. We have seen that many of them do not have information about their social media accounts on the user profile screens. The Letterboxd site played an important role in our preference, apart from being very popular among movie users and having social media addresses in its user profiles. In short, we used the Letterboxd site to get links to movies and users' social media accounts, and Twitter to get social media information due to the ability to access detailed information about users. We have developed a Java-based script web scraping tool to collect data on the Letterboxd site. By using this tool, user names, Twitter usernames, and movie ratings were collected from the relevant site. Approximately 10 out of every 100 users on the Letterboxd website were seen to have social media information on their profile. For this reason, a small function has been added to the data scraping tool during the data collection process to collect profiles containing only social media information. By using our web scraping tool, a total of 1,344,930 movie ratings were collected, including 1059 users and an average of 1270 for each user. To store this data, we preferred Cassandra, which is NoSQL-based and can perform fast and flexible transactions. The format of the dataset collected from the Letterboxd site and a few sample records can be seen in Table 1.

Table 1. The format of the data collected by web scraping from the Letterboxd.

id	**film**	**rating**	**username**	**twitterusername**
79bc11435fb1	Deadly Force	3.0	jacobknight	blairpac
4dc568b37367	Aladdin	2.0	rstrahs	beforesunsct
70cf6fa85a21	Audition	3.5	annan	dregmobile
2928a5da3d80	Cops & Robbersons	2.0	bigdaddywarbuxx	davidlsims
85527421b67d	Igby Goes Down	4.5	emilybabyy	ccbaxter08

The id value in the data demonstrates the user's identity information on Twitter. By using this credential, the user's account name is first found in the Twitter profile data. Then the Letterboxd user who has this account name is accessed. All movie evaluations of the relevant user are attained within the evaluation dataset together with the reached Letterboxd user. The assessment for the related film is taken from this evaluation data and combined with the user's profile. Using social media data for external or implicit information would be beneficial, mainly because the film industry is an important agenda on social media platforms. We have observed that using Twitter as a social media platform is very convenient. Because Twitter is one of the largest social media platforms and is actively used by users worldwide. Also, collecting the desired data using Twitter's web services directly allows you to access the data reasonably quickly and practically compared to web scraping technologies. For these reasons, the profile and tweet data of users were gathered by accessing Twitter web services using the Twitter usernames of the users in Letterboxd. Twitter is an excellent platform for accessing much information about the behavior of users. It offers information about users, their related areas, hobbies, and behavior. By contacting the Twitter web services directly, we collected each user's profile information and the most recent 3000 tweets from the same user. A total of 2,185,776 tweet data were collected, including 1059 user profiles and an average of 2064 tweets for each user. This data is stored in the free open-source Cassandra, as in the Letterboxd data. After collecting the data mentioned in the data collection section, these data should be turned into meaningful. Making sense of this data and inferring various characteristics is very critical for the success of the recommender system. In order to process more than 1 million user tweets, an architectural tool where we can perform distributed, and parallel transactions were preferred. The frequency analysis of the tweet data belonging to the users on this parallel processing tool has been evaluated using statistical methods such as cumulative total and maximum value findings. After this evaluation, various user features thought to be meaningful for the recommender system has been extracted [27]. Then, the user profile creation process is completed by combining the various features (such as the number of followers, the number of following, and the number of likes) in the Twitter user profile.

3.3. Feature extraction

We have processed the user profile and tweet data we receive from Twitter to create each user's Twitter profile. The data arising after this processing defines the properties we will use during the modeling stage. Features such as the Twitter registration date of users, the number of followers, and the number of the following can be accessed directly from the Twitter user table. We processed tweet data to extract features that have some special meanings. For example, the day users use to tweet or the time they use it, the total number of interactions they receive from tweets they post, whether they have frequent retweeting users, etc. These extracted user properties that represent each user's profile have been transposed to a separate Cassandra table. The features that emerge using all Twitter data and are the inputs of our model are shown in Table 2.

Table 2. List of obtained attributes

Feature ID	Feature Name	Explanation
F1	The date of creation of the Twitter account	Numerical value
F2	The total number of users' favorites/likes	Numerical value
F3	Total number of followers	Numerical value
F4	Total number of the followings	Numerical value
F5	Total number of hashtags used	Numerical value
F6	A feature that indicates that the user has retweeted more	True or false
F7	Number of lists of the user	Numerical value
F8	The most frequently preferred day for the user to tweet	Numerical value
F9	The time that the user uses most often to tweet	Numerical value
F10	The source from which the user sends their tweets	Categorical value
F11	The total number of tweets of the user	Numerical value
F12	The overall number of interactions that the user's tweets have received	Numerical value
F13	The total count of comments the user has received	Numerical value
F14	A property that indicates whether there is an approved account or not	True or false
F15	A feature that specifies when the user is using Twitter day or night	True or false
F16	Using Twitter on weekdays or weekends	True or false

3.4. Classification algorithms

3.4.1. AdaBoost

AdaBoost creates a model based on training data. Then it forms a new model to correct the errors in this model. Models continue until the items in the data set are perfectly predicted or until the maximum number of models is reached. It is one of the first boosting algorithms to achieve successful binary classification results. A weak classifier is trained, and all the example data samples are given equal weight. A weight is calculated for the classifier, with more accurate classifiers being given a higher weight and less accurate a lower weight. The weight is calculated based on the classifier's error rate, which is the number of misclassifications in the training set, divided by total training set size. This output weight per model is known as the "alpha." Each classifier will have a weight calculated based on the classifier's error rate. For each iteration, the alpha of the classifier is calculated, with the lower the error rate the higher the alpha.

$$\alpha_t = \frac{1}{2} ln(\frac{1-\epsilon_t}{\epsilon_t}) \tag{1}$$

After weak classifier is trained, we update the weight of each training example with following formula.

$$D_{t+1}(i) = \frac{D_t(i) exp(-\alpha_t y_i h_t(x_i))}{Z_t} \tag{2}$$

D_t is weight at the previous level. We normalize the weights by dividing each of them by the sum of all the weights, Z_t. y_i is y par of training example y coordinate for simplicity. Once all of the iterations have been completed, all of the weak learners are combined with their weights to form a strong classifier, as expressed in the below equation:

$$H(x) = sign(\sum_{t=1}^{T} \alpha_t h_t(x)) \tag{3}$$

The final classifier is therefore built up of "T" weak classifiers, h_t is the output of the weak classifier, with the weight applied to the classifier. Therefore, the final output is a combination of all the classifiers [33].

3.4.2. Random forest

The Random Forest algorithm is one of the supervised classification algorithms. It is used in both regression and classification problems. The algorithm aims to raise the classification value by generating multiple decision trees during the grading process. The Random Forest algorithm chooses the highest score among many decision trees that work independently. The main difference between the Decision Trees algorithm and the Random Forest algorithm is that finding the root node and dividing the nodes is random.
Feature importance is calculated as the decrease in node impurity weighted by the probability of reaching that node. The node probability can be calculated by the number of samples that reach the node, divided by the total number of samples. The higher the value, the more important the feature. The final feature importance, at the Random Forest level, is it's average over all the trees. The sum of the feature's importance value on each trees is calculated and divided by the total number of trees:

$$RFf_i = \frac{\sum_{j \epsilon all trees} normfi_{ij}}{T} \tag{4}$$

RFf_i is the importance of the feature i calculated from all trees in the Random Forest model. $normfi_{ij}$ is the normalized feature importance for i in tree j. T is total number of trees [34].

3.4.3. Gradient boosting

A combination of weak forecasting models typically creates a model of Decision Trees. We may find minimal error values by using gradient descent and updating our estimates according to the learning rate. The intuition behind the Gradient Boosting algorithm is to use the patterns in residuals repeatedly and to strengthen and improve a model with poor predictions. When we reach a stage where there are no patterns on which the remnants can be modeled, modeling can stop the residues. The purpose of boosting is to sequentially apply the weak classification algorithm to repeatedly modified versions of the data, thereby producing a sequence of weak classifiers.

$$F_0(x) = argmin \sum_{i=1}^{n} L(y_i, \gamma) \tag{5}$$

The first step is creating an initial constant value prediction F_0. L is the loss function and it is squared loss in our regression case.

$$r_{im} = -\left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}\right]_{F(x)=F_{m-1}(x)} \qquad for\ i = 1, ..., n \tag{6}$$

We are calculating residuals r_{im} by taking a derivative of the loss function with respect to the previous prediction F_{m-1} and multiplying it by -1. As you see in the subscript index, the r_{im} is computed for each single sample i.

$$F_m(x) = F_{m-1}(x) + \gamma_m h_m(x) \tag{7}$$

We always fit a base learner to the gradient of the loss function write the model F_{m-1}. The term boosting refers to the fact that a high bias model, which performs bad on the dataset, is boosted to finally become a reasonable classifier and possibly a strong classifier [35].

3.4.4. XGBoost

Extreme Gradient Boosting (XGBoost) algorithm, decision trees are created in sequential form. It is a decision tree-based algorithm that uses gradient boosting. Weights play an important role in XGBoost. Weights are assigned to all the independent variables, then fed into the decision tree, which predicts results. The weight of variables predicted wrong by the tree increases, and these variables are then fed to the second decision tree. These predictors then ensemble to give a strong model [28].

$$\widetilde{L}^{(t)} = \sum_{i=1}^{n} \left[g_i f_t(X_i) \tfrac{1}{2} h_i {f_t}^2(X_i)\right] + \Omega(f_t) \tag{8}$$

The above is a sum of simple quadratic functions of one variable and can be minimized by using known techniques, so our next goal is to find a learner that minimizes the loss function at iteration t.

3.4.5. LightGBM

LightGBM is a histogram-based algorithm. It reduces the calculation cost by discrete variables that have a continuous value. The training time of decision trees is directly proportional to the calculation performed and, thereby the number of divisions. With this method, the training period is shortened, and resource usage is reduced. With the leaf-oriented strategy, the model has less error rate and learns faster. However, the leaf-oriented growth strategy brings about excessive learning of the model in cases where the number of data is low. Therefore, the algorithm is more convenient for use in extensive data.

Gradient One-Sided Sampling utilizes every instance with a larger gradient and does the task of random sampling on the various instances with the small gradients. The training dataset is given by notating O for each particular decision tree node. The variance gain of j or the dividing measure at the point d for the node is given by [36]:

$$\widetilde{V}_j(d) = \frac{1}{n}\left(\frac{\left(\Sigma_{x_i \in A_l} g_i + \frac{1-a}{b}\Sigma_{x_i \in B_l} g_i\right)^2}{n_l^j(d)} + \frac{\left(\Sigma_{x_i \in A_r} g_i + \frac{1-a}{b}\Sigma_{x_i \in B_r} g_i\right)^2}{n_r^j(d)}\right) \tag{9}$$

3.4.6. CatBoost

Catboost (Category and Boosting) gets its name from the words "Category" and "Boosting." It is a gradient augmentation algorithm that uses decision trees. Since Catboost produces a symmetrical tree, it produces fast results during the training stage, significantly reducing training time. CatBoost uses a more efficient strategy that reduces overfitting and allows to use of the whole dataset for training. We perform a random permutation of the dataset, and for each example, we compute the average label value for the example with the same category value placed before the given one in the permutation [29].

$$\frac{\sum_{j=1}^{p-1}\left[x_{\sigma_j,k}=x_{\sigma_p,k}\right]Y_{\sigma_j}+a.P}{\sum_{j=1}^{p-1}\left[x_{\sigma_j,k}=x_{\sigma_p,k}\right]+a} \tag{10}$$

Let $\sigma=\left(\sigma_1,...,\sigma_n\right)$ be the permutation, then $x_{\sigma_p,k}$ is substituted with where we also add a prior value P and a parameter a $>$ 0, which is the weight of the prior.

3.5. Evaluation metrics

We will use various indicators to measure the accuracy of our estimates. These indicators, Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE), are markers that are widely used today in the evaluation of recommender systems [30]. Precision, Recall, and F1 measurement are the most common gauges used to evaluate boosting algorithms.

- Precision: It determines the ratio of recommended movies to movies with a really high rating.

$$precision=\frac{True\ Positive}{True\ Positive+False\ Positive} \tag{11}$$

- Recall: It is a metric that shows how many processes we need to predict as Positive are positive.

$$recall=\frac{True\ Positive}{True\ Positive+False\ Negative} \tag{12}$$

- F1-Score: It shows the harmonic average between certitude and sensitivity. Extreme situations should not be ignored. Consequently, F1-Score is an essential evaluation criterion.

$$F1=2\cdot\frac{precision.recall}{precision+recall} \tag{13}$$

- Root Mean Squared Error: It is the standard deviation of the remoteness between the estimated value and the actual value.

$$RMSE=\sqrt{\frac{\sum\left(y_i-y_p\right)^2}{n}} \tag{14}$$

$y_i=actual\ value,\ y_p=predicted\ value,\ n=number\ of\ observations$

- Mean Absolute Error: Mean Absolute Error measures the difference between two continuous variables. When the average absolute error is low, it means that the recommender system has successfully predicted the user's assessments.

$$MAE=\frac{|y_i-y_p|}{n} \tag{15}$$

- Hit Ratio: We use the hit ratio to evaluate the first top 20, in other words, if a user evaluates one of the top 20 that we recommend, it is considered a "hit."

$$HR@20 = \frac{hits}{hits + misses} \tag{16}$$

4. Experimental results

Afterward, models were developed for the classification problem with the help of the previously mentioned Boosting algorithms. 5-fold cross-validation was used in the training and testing phases. When developing different models, 80% of datasets are allocated to training, and the remaining 20% are distinguished to evaluate forecasts. AdaBoost, Random Forest, Gradient Boosting, XGBoost, LightGBM, and CatBoost classifiers with using have developed a separate model for 20 films. After calculating and expressing all these values separately on a movie basis, these 6 evaluation measures of the movies are taken, and their overall performance is obtained. The measurement results of these models developed for each film are shown in Table 3.

Table 3. The measurement values of boosting algorithms for 20 films.

Model	Precision	Recall	F1 Score	RMSE	MAE	HR
AdaBoost	0.7148	0.7263	0.7472	0.5754	0.3381	0.1727
Random Forest	0.7488	0.7622	0.7634	0.5896	0.3272	0.1984
Gradient Boosting	0.7552	0.7685	0.7885	0.5561	0.3109	0.2365
XGBoost	**0.8129**	0.8246	0.8566	0.4410	0.2331	0.4249
LightGBM	0.8079	0.8128	0.8407	0.4612	0.2289	**0.4568**
CatBoost	0.8106	**0.8379**	**0.8693**	**0.4234**	**0.2151**	0.4392

Considering the measurement values, all 6 classifiers give close results. CatBoost outperformed LightGBM and XGBoost. The availability of more than one categorical data in user profiles has improved the CatBoost algorithm's performance in the recommender system. Another point to mention is LightGBM's success in recommending movies that are genuinely (Hit Ratio) advised for the user. Also, XGBoost's precision is better than other algorithms. The CatBoost algorithm has successfully made correct predictions compared to other Boosting algorithms. Low levels of root mean square error and mean absolute error demonstrate the success of the proposed system. CatBoost has performed more successfully at the point of error than other classifiers. It has been observed that the CatBoost classifier is more resolute than other classifiers. AdaBoost, Random Forest and Gradient Boosting algorithms are insufficient compared to other algorithms. Compared to other studies in the literature, this study is quite efficient in terms of both error and success rates. Comparative results are shown in Table 4.

In the literature, it is seen that measurements are generally made with Precision, F1-Score, RMSE and MAE evaluation metrics. In our study, a more detailed measurement is made by adding Hit Ratio and Recall metrics to these metrics. The hit Ratio value is the most important criterion that shows the success of a recommendation system. Here, it means giving the right suggestions to the relevant person or even giving the right suggestion list. In this study, the list value of the Hit Ratio is 20. Generally, lists of 5, 10, and 20 are created in the literature. The value of 20 in our study is quite high compared to other studies. In this study, looking at the Hit Ratio value constitutes the original side of this study. When the MAE values were examined, our study yielded close results with [16] and [20]. Considering the RMSE values, our study lagged [14], [20] and [25]. When the Precision values were examined, our study [19] and [25] gave similar results. Our study was quite successful according to F1-Score values. In short, it is seen that Hit Ratio, Recall and F1-Score values in this study achieved better results than other studies in the literature. In this project, a different solution for the cold start problem was proposed and the recommendation process was improved according to the experimental results.

Table 4. Comparison of measurement results with other studies

Author(s)	Years	Precision	F1-Score	RMSE	MAE
Our model	2022	0.81	0.86	0.42	0.21
Leung et al. [8]	2008	0.25	0.17	-	-
Zhong et al. [9]	2022	-	-	0.91	0.77
Kim et al. [10]	2011	-	-	-	0.75
Zhang et al. [11]	2022	-	-	1.01	0.79
Ahn [12]	2008	-	-	-	0.77
Bobadilla et al. [13]	2011	0.45	-	-	0.74
Tian et al. [14]	2019	-	-	0.14	-
Martins et al. [15]	2015	0.58	-	-	-
Pereira et al. [16]	2015	-	-	-	0.18
Nguyen et al. [17]	2017	-	-	-	0.85
Viktoratos et al. [18]	2018	-	-	-	-
Xing et al. [19]	2021	0.8	-	-	-
Lika et al. [20]	2014	-	-	0.15	0.15
Natarajan et al. [21]	2020	-	-	0.80	0.60
Silva et al. [22]	2019	-	0.23	-	-
Feng et al. [23]	2021	0.17	-	-	-
Herce-Zelaya et al. [24]	2020	-	-	-	0.30
Wahab et al. [25]	2022	0.95	-	0.1	-
Kawai et al. [26]	2022	-	0.11	-	1.52

5. Conclusion and future works

When the measured results are examined, it can be said that the recommender system developed to advise users during a cold start is quite efficient. In addition, our work on privacy, one of the problems in today's recommender systems related to user data, is promising. It has been shown that a significant improvement has been made in solving the problem by obtaining only one social media account information from the user without encountering long forms or various questions for the cold start problem. Our study shows that implicit knowledge in the solution of cold start problems about recommender systems plays an important role in improving the success of the systems. Another important aspect of the study is that it does not take any help from the absolute rating data and advises directly with external information. Since the model mentioned in the study uses only contextual data from Twitter; it will not be correct to compare this study with a cutting-edge technology recommender system that does not contain a solution to the cold start problem. The rating data used in the study were merely utilized to build and validate the model. For this reason, it would not be fair to compare it with a recommender system that utilizes users' past evaluations. The proposed model is easily useable because it is a classification problem, can quickly generate recommendations, and does not require excessive memory. Also, owing to its flexibility, it can be effortlessly used in many areas where recommender systems are used today.

In the future, natural language processing techniques can be added to the recommender system using many user tweets or product comments. By doing more work on the features, the system's performance can be improved by adding new features. The proposed model may cause problems when users are new to the system, and their external information cannot be accessed. To solve this problem, can be improved with add various external information that can be used about the user to the proposed system. In addition, the accounts of the user's friends on the user's Twitter account can also be examined and solutions can be sought for the cold start problem using methods such as user similarity.

References

[1] Rawat B, Samriya JK, Pandey N, Wariyal SC. A comprehensive study on recommendation systems their issues and future research direction in e-learning domain. Materials Today: Proceedings. 2020. https://doi.org/10.1016/j.matpr.2020.09.796

[2] Pu P, Chen L, Hu R. A user-centric evaluation framework for recommender systems. In: Proceedings of the fifth ACM conference on Recommender systems; 2011. pp. 157-164.

[3] Hwangbo H, Kim YS, Cha KJ. Recommendation system development for fashion retail e-commerce. Electronic Commerce Research and Applications 2018; 28: 94-101. https://doi.org/10.1016/j.elerap.2018.01.012

[4] Gunawan A. A, Suhartono D. Music recommender system based on genre using convolutional recurrent neural networks. Procedia Computer Science 2019; 157: 99-109. https://doi.org/10.1016/j.procs.2019.08.146

[5] Chen MH, Teng CH, Chang PC. Applying artificial immune systems to collaborative filtering for movie recommendation. Advanced Engineering Informatics 2015; 29 (4): 830-839. https://doi.org/10.1016/j.aei.2015.04.005

[6] Abbasi-Moud Z, Vahdat-Nejad H, Sadri J. Tourism recommendation system based on semantic clustering and sentiment analysis. Expert Systems with Applications 2021; 167:114324. https://doi.org/10.1016/j.eswa.2020.114324

[7] Tian Y, Zheng B, Wang Y, Zhang Y, Wu Q. College library personalized recommendation system based on hybrid recommendation algorithm. Procedia CIRP 2019; 83: 490-494. https://doi.org/10.1016/j.procir.2019.04.126

[8] Leung CWK, Chan SCF, Chung FL. An empirical study of a cross-level association rule mining approach to cold-start recommendations. Knowledge-Based Systems 2008; 21 (7): 515-529. https://doi.org/10.1016/j.knosys.2008.03.012

[9] Zhong D, Yang G, Fan J, Tian B, Zhang, Y. A service recommendation system based on rough multidimensional matrix in cloud-based environment. Computer Standards Interfaces 2022; 82: 103632. https://doi.org/10.1016/j.csi.2022.103632

[10] Kim HN, El-Saddik A, Jo GS. Collaborative error-reflected models for cold-start recommender systems. Decision Support Systems 2011; 51 (3): 519-531. https://doi.org/10.1016/j.dss.2011.02.015

[11] Zhang J, Ma C, Zhong C, Zhao P, Mu X. Combining feature importance and neighbor node interactions for cold start recommendation. Engineering Applications of Artificial Intelligence 2022; 112:104864. https://doi.org/10.1016/j.engappai.2022.104864

[12] Ahn HJ. A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem. Information Sciences 2008; 178 (1): 37-51. https://doi.org/10.1016/j.ins.2007.07.024

[13] Bobadilla J, Ortega F, Hernando A, Bernal J. A collaborative filtering approach to mitigate the new user cold start problem. Knowledge-Based Systems 2012; 26:225-238. https://doi.org/10.1016/j.knosys.2011.07.021

[14] Tian G, Wang Q, Wang J, He K, Zhao W et al. Leveraging contextual information for cold-start Web service recommendation. Concurrency and Computation. Practice and Experience 2019; 31 (17): e5195. https://doi.org/10.1002/cpe.5195

[15] Martins EF, Belém FM, Almeida JM, Gonçalves MA. On cold start for associative tag recommendation. Journal of the Association for Information Science and Technology 2016; 67 (1): 83-105. https://doi.org/10.1002/asi.23353

[16] Pereira ALV, Hruschka ER. Simultaneous co-clustering and learning to address the cold start problem in recommender systems. Knowledge-Based Systems 2015; 82: 11-19. https://doi.org/10.1016/j.knosys.2015.02.016

[17] Nguyen VD, Sriboonchitta S, Huynh VN. Using community preference for overcoming sparsity and cold-start problems in collaborative filtering system offering soft ratings. Electronic Commerce Research and Applications 2017; 26: 101-108. https://doi.org/10.1016/j.elerap.2017.10.002

[18] Viktoratos I, Tsadiras A, Bassiliades N. Combining community-based knowledge with association rule mining to alleviate the cold start problem in context-aware recommender systems. Expert Systems with Applications 2018; 101: 78-90. https://doi.org/10.1016/j.eswa.2018.01.044

[19] Xing Y, Wang F, Zeng A, Ying F. Solving the cold-start problem in scientific credit allocation. Journal of Informetrics 2021; 15 (3): 101157. https://doi.org/10.1016/j.joi.2021.101157

[20] Lika B, Kolomvatsos K, Hadjiefthymiades S. Facing the cold start problem in recommender systems. Expert Systems with Applications 2014; 41 (4): 2065-2073. https://doi.org/10.1016/j.eswa.2013.09.005

[21] Natarajan S, Vairavasundaram S, Natarajan S, Gandomi AH. Resolving data sparsity and cold start problem in collaborative filtering recommender system using linked open data. Expert Systems with Applications 2020; 149: 113248. https://doi.org/10.1016/j.eswa.2020.113248

[22] Silva N, Carvalho D, Pereira AC, Mourão F, Rocha L. The pure cold-start problem: a deep study about how to conquer first-time users in recommendations domains. Information Systems 2019; 80: 1-12. https://doi.org/10.1016/j.is.2018.09.001

[23] Feng J, Xia Z, Feng X, Peng J. RBPR: a hybrid model for the new user cold start problem in recommender systems. Knowledge-Based Systems 2021; 214: 106732. https://doi.org/10.1016/j.knosys.2020.106732

[24] Herce-Zelaya J, Porcel C, Bernabé-Moreno J, Tejeda-Lorente A, Herrera-Viedma E. New technique to alleviate the cold start problem in recommender systems using information from social media and random decision forests. Information Sciences 2020; 536: 156-170. https://doi.org/10.1016/j.ins.2020.05.071

[25] Wahab OA, Rjoub G, Bentahar J, Cohen R. Federated against the cold: a trust-based federated learning approach to counter the cold start problem in recommendation systems. Information Sciences 2022; 601: 189-206. https://doi.org/10.1016/j.ins.2022.04.027

[26] Kawai M, Sato H, Shiohama T. Topic model-based recommender systems and their applications to cold-start problems. Expert Systems with Applications 2022; 202: 117129. https://doi.org/10.1016/j.eswa.2022.117129

[27] Mika P. Social networks and the semantic web. PhD, Vrije Universiteit, Amsterdam, Nederland, 2006.

[28] Chen T, Guestrin C. Xgboost: a scalable tree boosting system. In Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining; San Francisco California USA; 2016. pp. 785-794.

[29] Dorogush AV, Ershov V, Gulin A. CatBoost: gradient boosting with categorical features support. Preprint arXiv 2018; 1810.11363. https://doi.org/10.48550/arXiv.1810.11363

[30] Chen M, Liu P. Performance evaluation of recommender systems. International Journal of Performability Engineering 2017; 13 (8): 1246. https://doi.org/10.23940/ijpe.17.08.p7.12461256

[31] Zhang Y, Jiao JR. An associative classification-based recommendation system for personalization in B2C e-commerce applications. Expert Systems with Applications 2007; 33 (2): 357-367. https://doi.org/10.1016/j.eswa.2006.05.005

[32] Jiang Y, Shang J, Liu Y. Maximizing customer satisfaction through an online recommendation system: a novel associative classification model. Decision Support Systems 2010; 48 (3): 470-479. https://doi.org/10.1016/j.dss.2009.06.006

[33] Zhang Y, Ni M, Zhang C, Liang S, Fang S, et al. Research and application of AdaBoost algorithm based on SVM. In: IEEE 2019 8th Joint International Information Technology and Artificial Intelligence Conference; Chongqing, China 2019. pp. 662-666.

[34] Biau G, Scornet, E. A random forest guided tour. Test 2016; 25 (2): 197-227. https://doi.org/10.1007/s11749-016-0481-7

[35] Gao K, Chen H, Zhang X, Ren X, Chen J, et al. A novel material removal prediction method based on acoustic sensing and ensemble XGBoost learning algorithm for robotic belt grinding of Inconel 718. The International Journal of Advanced Manufacturing Technology 2019; 105 (1): 217-232. https://doi.org/10.1007/s00170-019-04170-7

[36] Ke G, Meng Q, Finley T, Wang T, Chen W, et al. Lightgbm: A highly efficient gradient boosting decision tree. In: 31st Conference on Neural Information Processing Systems; CA, USA 2017.