Computers, Materials & Continua
DOI:10.32604/cmc.2022.019805
Article

Tech Science Press

Ensemble Learning Based Collaborative Filtering with Instance Selection and Enhanced Clustering

G. Parthasarathy[1,*] and S. Sathiya Devi[2]

[1]Anna University, Chennai, 600025, India
[2]University College of Engineering, BIT Campus, Anna University, Tiruchirappalli, 620024, India
*Corresponding Author: G. Parthasarathy. Email: parthasarathee.g@gmail.com
Received: 26 April 2021; Accepted: 24 June 2021

Abstract: Recommender system is a tool to suggest items to the users from the extensive history of the user's feedback. Though, it is an emerging research area concerning academics and industries, where it suffers from sparsity, scalability, and cold start problems. This paper addresses sparsity, and scalability problems of model-based collaborative recommender system based on ensemble learning approach and enhanced clustering algorithm for movie recommendations. In this paper, an effective movie recommendation system is proposed by Classification and Regression Tree (CART) algorithm, enhanced Balanced Iterative Reducing and Clustering using Hierarchies (BIRCH) algorithm and truncation method. In this research paper, a new hyper parameters tuning is added in BIRCH algorithm to enhance the cluster formation process, where the proposed algorithm is named as enhanced BIRCH. The proposed model yields quality movie recommendation to the new user using Gradient boost classification with broad coverage. In this paper, the proposed model is tested on Movielens dataset, and the performance is evaluated by means of Mean Absolute Error (MAE), precision, recall and f-measure. The experimental results showed the superiority of proposed model in movie recommendation compared to the existing models. The proposed model obtained 0.52 and 0.57 MAE value on Movielens 100k and 1M datasets. Further, the proposed model obtained 0.83 of precision, 0.86 of recall and 0.86 of f-measure on Movielens 100k dataset, which are effective compared to the existing models in movie recommendation.

Keywords: Clustering; ensemble learning; feature selection; gradient boost tree; instance selection; truncation parameter

1 Introduction

The exponential increase of data in the digital universe has encouraged efficient information filtering and personalization technology. Recommender System (RS) is a popular technique to perform both information filtering and personalization to the end-user from the huge information space. Nowadays, RS is an integral part of every e-commerce application such as Amazon, Twitter, Netflix,

LinkedIn, etc., to provide more relevant and personalized suggestions. So, Recommender Systems (RSs) are the systems that provides recommendations based on user's past behavior. Tapestry is the oldest recommendation system that filters the mail, which is interested in the user [1]. The RS collects information explicitly or implicitly to make recommendations. The rating information (i.e., like/dislike, a discrete rating) given by the users on products is called explicit, and the information collected from users' behavior (i.e., feedback, browsing behavior) is called implicit [2]. RS is broadly divided into three types namely (i) Collaborative Filtering (CF) (ii) Content-Based Filtering (CBF) and (iii) Hybrid CF recommendation systems that finds relevant items by finding the users having similar interests [3]. Content-Based Filtering (CBF) suggests items that are similar in features and the user has already chosen in the past [4]. In the hybrid approach, two or more methods are combined to gain better results. Vekariya et al. [5] mentioned the hybrid system types given by Robert Burke: weighted, switching, mixed technique, feature combination cascade, feature augmentation, and meta-level. Hence, there are three approaches in RS in that CF is successful in research and practice.

There are two types of CF approaches namely (i) Memory-based and (ii) Model-based approach. The memory-based approach uses the entire instance of the database, which results in scalability. The model-based approach tries to reduce a massive dataset into a model and performs recommendation task. Model-Based CF (MBCF) reacts to the user's request instantly with reduced computation. There are five primary approaches in MBCF such as classification, clustering, latent model, Markov Decision Process (MDP), and Matrix Factorization (MF) [6]. Generally, MBCF is preferred over memory-based, because it requires less computational cost and performs recommendations without domain knowledge. Hence, model-based CF is highly preferable and it suffers from three main issues like (i) scalability, (ii) sparsity and (iii) cold start. Sparsity refers that most of the users does not have enough rating to find the similar user. Cold start refers to the challenge in producing specific recommendations for the new or cold users, who rated an inadequate number of items. Scalability reduces the efficiency of handling a vast amount of data, when recommending to the user. Apart from these issues, building a user profile for new users by CF is difficult [7]. The main objective of this research paper is to develop a new CF approach by combining the user and item related feature to provide a solution to scalability and sparsity issue. The proposed system performs a mixture of clustering and ensemble-based classification using feature combination for a recommendation. The main contribution of this paper is summarized as follows:

- Proposed a new feature, and instance selection method; hierarchical enhanced BIRCH based clustering algorithm to overcome data sparsity.
- Incorporating CART based feature, and truncation parameters for normal distribution based instance selection.
- Developed an ensemble based Gradient Boosting Tree (GBT) recommendation model which improves recommendation accuracy, and also addresses the scalability issue.

The rest of the paper is organized as follows: Section 2 presents the related work review. A detailed description of the proposed approach is given in Section 3. Section 4 provides the experimental result on the benchmark datasets. Finally, the conclusion of the work is presented in Section 5.

2 Literature Review

This section reviews the existing collaborative recommendation approaches in movie recommendation. CF is a technique, which automatically predicts the unknown ratings of the product (or) user's interest by analyzing the known ratings of it (or) compiling preferences of similar users. The CF is used to develop a personalized recommendation on many e-commerce applications on the web. The

main process of CF is to identify the similar users for guiding the active user. In memory-based CF, instance-based methods are employed to determine similar users, but it suffers from poor scalability for a vast database. On the other hand, model-based CF approach is commonly used in offline dataset for prediction and recommendation. The model-based CF approach is small, which occupies less memory and work faster. Identifying a group of similar users is a challenging task in both memory and model-based approaches. Generally, a group of similar users is generated using clustering algorithms.

Ju et al. [8] developed a collaborative model based on k-means clustering and artificial bee colony algorithm. The developed algorithm was used to address the local optima problem of k-means clustering, and the similarity measure to perform clustering presented by Rongfei et al. [9]. The adjusted DBSCAN algorithm was utilized to develop a cluster that improves the accuracy of movie recommendation for the users, who having many available ratings. Das et al. [10] has presented a K-d trees and quad trees based hierarchical clustering method for movie recommendation. The developed method addresses the scalability issue and maintain acceptable recommendation accuracy. Experimental result proved that the computation time was effectively reduced on movielens-100K, movielens-1M, book-crossing, and trip advisor datasets.

Mohammad pour et al. [11] introduced a CF method based on hybrid meta-heuristic clustering for movie recommendation. The developed method merges a genetic algorithm and a gravitational emulation bounded clustering search. Here, the cluster was efficient, but suffers from higher computational cost. In addition, a Modified Cuckoo Search (MCS) algorithm and a Modified Fuzzy C Means (MFCM) approach was developed by Selvi et al. [12]. In this method, the number of iterations and error rates were reduced by MFCM. The recommendation accuracy and the efficiency of clustering was improved by MCS algorithm.

Generally, the cluster's discrimination ability and the cluster's performance depends on dimensionality reduction and it was performed in two ways (i) Feature selection, and (ii) Instance selection. Cataltepe et al. [13] has developed a new feature selection method for Turkish movie recommendation system. The developed method practices user behavior, various kinds of content features, and other users' message to predict the movie ratings. The developed method improves the recommendation's accuracy, notably for users who have viewed a deficient number of movies. The k-means clustering has been described along with a backward feature selection method to improve movie recommendation by Ramezani et al. [14]. The feature selection process eliminates irrelevant feature and makes the real similarity between the users. Further, an information-theoretic approach was developed by Yu et al. [15] for movie recommendation. The developed approach used description rationality, and the power to measure an instance's pertinence regarding a target notion. The empirical evaluation result showed that the developed method significantly reduces the neighborhood size and increases the CF process's speed.

Yu et al. [16] developed a system for feature and instance selection based on mutual information and Bayes theorem. This literature showed that the feature weighting and instance selection based on the pertinence analysis improves the collaborative filtering in light of accuracy. The integration of texture and visual features used by Pahuja et al. [17] were effective in movie recommendations. The feature sets have different level of significance in different scenarios and identified based on the business requirement. Further, the class based collaborative filtering algorithm was described by Zeng et al. [18] which adapts the user frequency threshold methodology for instance selection. The threshold selection improves the speed of computation, recommendation accuracy, and alleviates the cold start problem.

The CF's accuracy depends on the classification model, and the Extreme Gradient Boosting (XGBoost) algorithm-based recommendation system was described in Xu et al. [19]. Shao et al. [20] has introduced a Heterogeneous Information Boosting (HIBoosting) model based on Gradient Boosting Decision Tree (GBDT) algorithm. The developed model blends independent data in information networks to provide users with more helpful recommendation assistance.

From the above mentioned literatures, it is recognized that the model-based CF approach addresses the sparsity, and scalability issues better with feature reduction, clustering, and machine learning-based approaches. Still, the prediction accuracy and addition of new data incrementally becomes questionable. This research paper proposed a gradient boosting decision tree based CF approach with instance selection and enhanced clustering for an effective movie recommendation. Hence, the proposed model overcomes the sparsity and scalability issues and improves the accuracy of prediction and movie recommendation.

3 Proposed Methodology

The proposed collaborative movie recommendation approach with combined features and probabilistic based instance selection is described in this section. Generally, the RS suffers from three main issues such as sparsity, scalability, and cold start, irrespective of different implementation approaches. These issues affect the performance of RS. Hence this paper proposes an approach for model-based collaborative RS to solve the sparsity and scalability issues. The sparsity occurs due to the sparseness of the user-item matrix. The proposed approach considers both the ratings and content-based features of the data set and uses feature selection to overcome the sparsity problem. The later issue is addressed by enhanced clustering and instance selection. This approach addresses the scalability issue and improves the recommendation's accuracy when combined with an ensemble method with a limited computational cost. The proposed collaborative RS approach is shown in Fig. 1. The proposed approach consists of seven stages such as: (i) Preprocessing, (ii) Feature Selection (iii) Instance Selection (iv) Clustering (v) Model Creation (vi) Prediction and (vii) Recommendation. Each stage is described in detail in the forthcoming subsections.

3.1 Preprocessing

Preprocessing is a technique that cleans, integrates, and fills the missing values in the collected dataset to avoid the result's inconsistencies. The proposed approach considers both user ratings as well as content-based features for a recommendation. Since, these features are of different data types while integrating, there must be inconsistencies, which affects the prediction's performance. Hence, the proposed approach applies label encoding while combining it to make them the same (or) similar data type [21]. The missing values are filled with the mean value for the corresponding features in the input dataset.

3.2 Feature Selection

Feature selection selects the most influencing features from the available dataset to avoid computational complexity while training and testing that improves the recommendation model's generalization. In the proposed approach, feature selection is utilized to reduce the sparsity of the integrated feature dataset. The proposed method uses the correlation-based mutual information measure to identify the entire feature set's significant features. It considers feature importance, and used to choose the features based on the relative rank of features from a tree. In the proposed approach, feature importance is implemented based on the Classification and Regression Tree (CART) algorithm. Since the target

variable in the proposed approach is categorical values, CART uses the Gini index as an impurity measure to find the splits in the tree. Gini index is a measure of inequality practiced in the irregular pattern of data. The Gini index always results in a quantity between 0 and 1, where 0 resembles perfect equality, and 1 replies to perfect inequality. The minimum value 0 occurs when all the data at a feature (node) belongs to one target category. The Gini index at a feature (node) t is defined in Eq. (1).

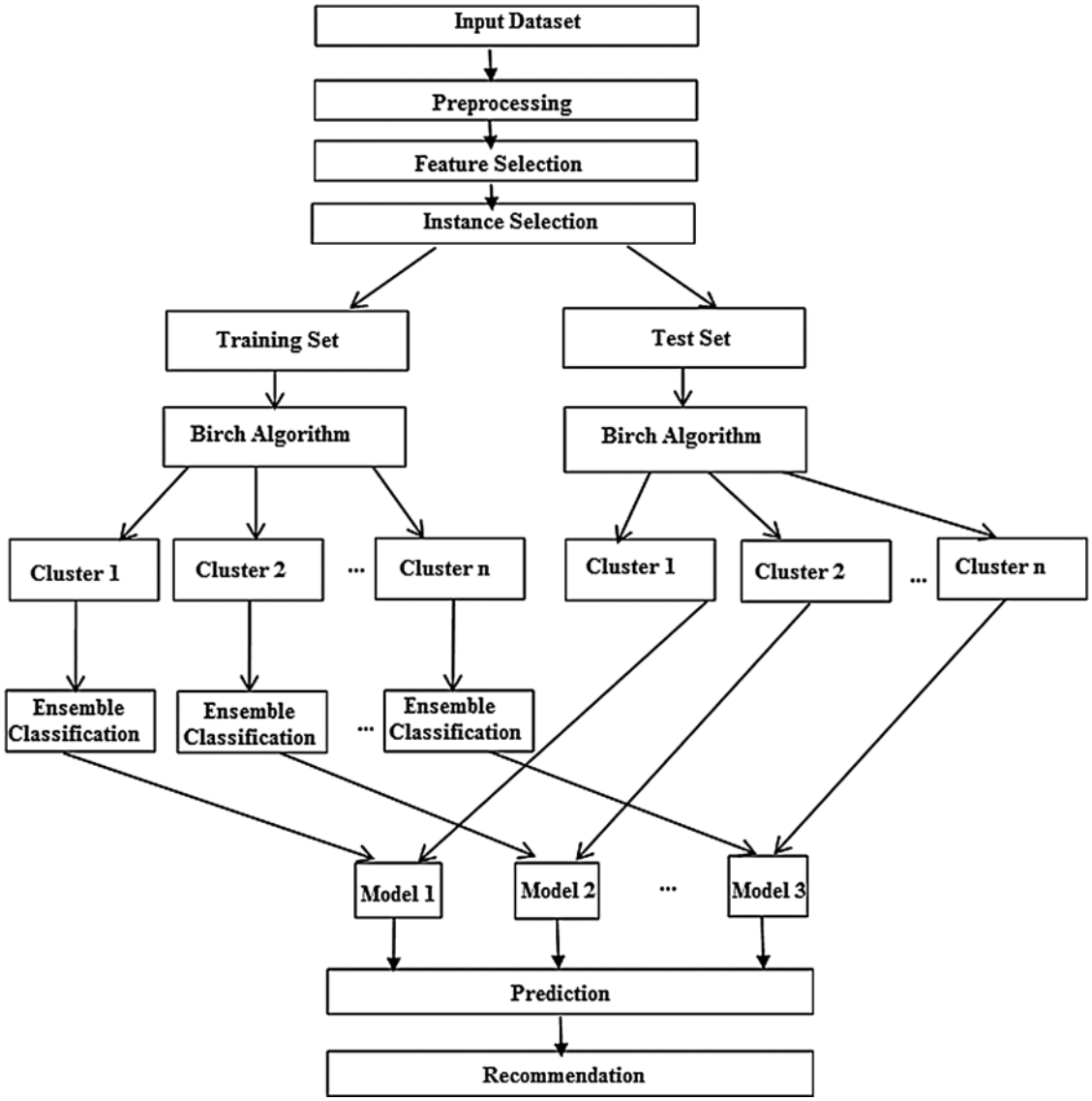

Figure 1: Proposed collaborative recommendation approach

$$Gini(t) = \sum_{j \neq i} p\left(\frac{j}{t}\right) p\left(\frac{i}{t}\right) \tag{1}$$

where i and j are the kinds of the target value, and p is indicated as probability. Eq. (1) can be rewritten as represented in Eq. (2).

$$Gini(t) = 1 - \sum_j p2\left(\frac{j}{t}\right) \tag{2}$$

where $p\left(\frac{i}{t}\right)$ indicates the proportion of target category j present in feature (node) t. The Gini criterion for the split's' at a feature t is defined in Eq. (3).

$$Gini_{split}(s,t) = Gini(t) - p_L.Gini(t_L) - p_R.Gini(t_R) \tag{3}$$

where $p_L$and p_R are the proportion of instances in t sent to the left child and right child features (nodes) respectively and s $\in$ S refers to a particular generic split among all possible set of splits S. The steps involved in CART algorithm is given below:

Step 1: Starts from the root node $t = 1$, search for a split $s*$ among all potential candidate's s that gives the high decrease in impurity. Then split node $1(t = 1)$ into two nodes $t = 2$ and $t = 3$, using split$s*$.

Step 2: Repeat the method in each of $t = 2$ and $t = 3$, then extend the tree breeding process till at most insignificant one of the tree growing rules is met. From the constructed tree, the feature importance is calculated by using Eq. (4).

$$fi_j = (GI)_j * (n)_j - (GI_l) * (n)_l - (GI)_r * (n)_r \tag{4}$$

where fi_j is the importance of feature j, GI_j is the Gini impurity value of node j, n_j is the number of instances falls in the root node, GI_l is the Gini impurity value of a left child node, n n_l is the number of instances fall in the left node, GI_r is the Gini impurity of the right child node, n_r is the number of instances fall in the right node. The normalized feature importance is calculated by dividing the feature importance by the sum of feature importance of all features, and it will be represented in percentage. The main advantages of feature selection are reducing over fitting, improves accuracy, and reduces training time. The features are selected using the feature importance scores, where almost 19 relevant features are selected from 31 features.

3.3 Instance Selection

In most of the Collaborative RS, the predictions are based on users' preferences similar to the active user. Though a similar user's search is significant in collaborative RS, the entire scan of the dataset leads to non-scalability issues and poor prediction performance when more users and items are added into the dataset. Hence the proposed approach adopts an instance selection strategy to filter the relevant users than searching for the entire data set. The proposed method incorporates instance selection using Probability Density Function (PDF) of a normal distribution, shown in Eq. (5). It achieves the instance selection by computing the truncation parameter (α) from the selected features.

$$f(x) = \left\{ \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x-\mu)^2}{2\sigma^2}} , where \ -\infty < x < \infty, \mu > 0, \sigma > 0 \right. \tag{5}$$

Most of the real-world datasets fall under the normal distribution density function. This distribution's empirical rule states that all the samples fall within three standard deviations of the mean. In that, 68% of the sample fall inside the first standard deviation from the mean, 95% fall inside two standard deviations, and 99.7% fall inside three standard deviations [22]. The mean value of the target value is

calculated using Eq. (6).

$$\mu = \frac{\sum x_i}{n} \tag{6}$$

Next, the standard deviation is calculated using the mean value obtained from Eq. (3). It is the average of the difference between a sample and mean value using Eq. (7).

$$\sigma = \sqrt{\frac{\sum (x_i - \mu)^2}{n}} \tag{7}$$

The normal distribution curve is plotted, and the truncation parameter is found from likely, very likely, and almost certainly values. The selected instances increase the mean value of the distribution, which means that the most reviewed items are selected. The instances are selected from the truncation algorithm (95% of the relevant instances), and output is given to the Enhanced BIRCH algorithm.

3.4 Enhanced BIRCH

The relevant users are identified in the previous subsection, where these users are partitioned into small groups based on the clustering algorithm. The clustering process in RS solves the scalability issue and increases recommendation accuracy with limited computational cost. In this scenario, clustering is performed based on BIRCH algorithm. It is one of the best hierarchical clustering algorithm for high dimensional data, but it suffers from the issue of initial and number of cluster assignment. So, the hyper parameters tuning is added to enhance the BIRCH algorithm for efficient cluster formation process. In the clustering approach, the number of clusters is to be given as input data. This optimal value of the number of groups (K) is decided using different methods. The Elbow method is one of the standard methods to choose the optimal number. The *K*value is calculated by using the inertia score, which is the sum of samples' squared distances to their closest cluster center. The average internal sum of squares (Wk) is the average distance between points inside a cluster, and it is mathematically expressed in Eq. (8).

$$W_k = \sum_{r=1}^{k} \frac{1}{n_r} D_r \tag{8}$$

where k is number of clusters, n_r is number of points in cluster$r$, D_r is sum of distances between each point in a cluster expressed in Eq. (9), and d indicates distance.

$$D_r = \sum_{i=1}^{n_r - 1} \sum_{j=1}^{n_r} ||d_i - d_j|| \tag{9}$$

The hyper parameters are needed to determine the number of clusters and to make the clustering computation faster. We have performed hyperparameters tuning in branching factor, compute labels, the number of clusters, and threshold value to enhance the algorithm. It fully utilizes accessible memory to infer the best conceivable sub-clusters to limit computational costs [23]. The cluster centroid is the mean of all the points in the dataset and it is expressed in Eq. (10). Where, x is the point in the dataset and n is the number of points.

$$\overline{x} = \sum \frac{xi}{n} \tag{10}$$

The root node is formed using the number of points, Linear Sum of points (LS), and Sum of the Squared of the points (SS). The radius R is calculated using Eq. (11) to create the leaf node.

$$R = \sqrt{\frac{\sum (x_i - \overline{x})^2}{n}} \tag{11}$$

The radius is compared with the threshold value (T), which is set initially. Based on the comparison, the next point is placed in the leaf node or the existing node. The number of a leaf node is restricted by using the value L. At the end of the first phase, the CF tree is built using the above steps. The second phase of Birch architecture is done using the above created CF tree in the agglomerative hierarchical clustering technique discussed earlier in this section. The next section discusses the model creation of the proposed model.

3.5 Model Creation

Ensemble methods plays a significant part in machine learning, GBT (Gradient Boost Tree) algorithm is one among them. A series of weak learners (decision trees) are ensemble by using a boosting technique. GBT produces additive models by sequentially implementing a base learner to current residuals by least-squares at each stage. GBT classification model performance is increased by tuning the hyperparameters, maximum depth, minimum sample split, learning rate, loss, number of estimators, and maximum features. Pseudo-residuals are the slope of the loss function being diminished, concerning the model estimations at all training data points estimated at the current step [24–26].

3.6 Prediction and Recommendation

The significance of RS mostly relies on the accurate prediction algorithm whose purpose is to approximate the value of the unseen data. According to this value, the system recommends to the user. The proposed approach utilizes the ensemble regression algorithm for effective prediction. The ensemble methodology combines a set of models, each of which performs a similar job to obtain a more reliable composite global model, more accurate and reliable. The proposed approach considers the Gradient Boost regression model for efficient model creation and prediction. This model adopts balanced and conditional recommendations. In gradient boost regression, a series of weak learners (decision trees) are constructed, boosting the classification performance by combining the respective learner. Gradient boosting constructs additive classification models by sequentially applying a simple parameterized function (base learner) to current pseudo residuals by least-squares at every iteration. Hence, the performance of the gradient boosting regression highly depends on parameter tuning. The proposed approach uses the Grid Search method to tune the hyper parameter of the model. A grid search is used for parameter tuning to build and evaluate a model for the different parameters of an algorithm defined in a network. The parameter to be tuned are: (i) Maximum depth (ii) Minimum Sample Split (iii). Learning rate (iv). Loss (v). Number of Estimators and (vi). Maximum features. The Grid Search performs search candidate sampling with k-fold cross-validation to tune the hyper parameters. The pseudo-residuals are the gradient of the loss function being reduced, concerning the model estimations at all training data points evaluated at the prevailing step. The performance of the model is discussed in section IV.

3.7 Performance Measure

The performance of the proposed approach is evaluated against the known measure for prediction and recommendations and is given below: For prediction, MAE is used and represented as the difference between the predicted rating of user u on item $i(p_{u,i})$ and the actual rating of user u on item $i(r_{u,i})$ and is represented in Eq. (12). For recommendation, precision and recall, and f-measures are used.

$$MAE = \frac{1}{N}\sum_{u,i}|p_{u,i} - r_{u,i}| \tag{12}$$

Precision is defined as the percentage of recommended items that are relevant to the user and expressed in Eq. (13). The recall is the ratio of correct recommendations relevant to the query to the total number of appropriate recommendations and is shown in Eq. (14). In addition, f-measure is the harmonic mean of precision and recall, and it is indicated in Eq. (15).

$$\text{Pr}\,ecision = \frac{Number\ of\ correct\ recommendations\ relevent\ to\ the\ total\ query}{number\ of\ recommendations} \tag{13}$$

$$recall = \frac{number\ of\ correct\ recommendation}{total\ number\ of\ relevant\ recommendation} \tag{14}$$

$$F\ measure = \frac{2}{\frac{1}{\text{Pr}\,ecision} + \frac{1}{recall}} \tag{15}$$

4 Experiment and Results

In this section, experiments of the proposed model is carried on Movielens 100k and 1M datasets [27,28]. Tabs. 1 and 2 lists two standard movie recommendation datasets, which are used as the benchmark. The number of users varies from 943 to 6040, and the number of items varies from 1682 to 3952. The number of ratings ranges from 1,00,000 to 1,000,209 and the density of rating ranges from 4.19% to 6.30%. In Movielens 100k and 1M datasets, the rating levels are whole star rating from 1 to 5, and each user has at least have 20 movie ratings. Especially, Tab. 2 lists the proportion of each rating level, mean (μ), and standard deviation (σ) among the various statistical measures.

Dataset link: https://grouplens.org/datasets/movielens/

Table 1: Datasets description

Dataset	u	i	r	Density
Movielens 100K	943	1,682	1,00,000	6.30%
Movielens 1M	6,040	3,952	1,000,209	4.19%

4.1 Experiment

The experiment is carried out on the windows platform using the python programming language. All the item and user features are combined with user preference of a movie. These features are combination of different formats like numbers and strings. Label Encoder applies to these features to make the features as a single data type. Almost 31 features are integrated using preprocessing

technique. Among 31 features, 19 features are chosen using feature selection in the Movielens 100k and 1M data sets.

Table 2: Basic statistical data

Dataset	Proportion of r in %					μ	σ
	1	2	3	4	5		
Movielens 100K	6.11	11.37	27.15	34.17	21.20	3.52	1.12
Movielens 1M	5.62	10.75	26.11	34.89	22.63	3.58	1.18

The first step truncation algorithm calculates the mean and standard deviation of the selected features by Eqs. (6) and (7) in Section 3.3. Next, the probability density function of a normal distribution is drawn using these features. The ranges of likely, very likely, and most likely values are found by using the mean and standard deviation values. The truncation parameter range 2σ is fixed based on the number of samples and the increase in the density function's peak value. In the selected parameter value, 95% of instances are selected, and also the peak value is increased by introducing the truncation parameter. The Truncation algorithm is fitted to the data set for determining the instances, which is shown in Figs. 2a & 2b. The curve in Fig. 2a shows the mean value of 3.52, and the density peak value is 0.36, and Fig. 2b shows the peak value that is increased to 0.52. It indicates that most of the data samples fall under our truncation parameter range, determining the more similar samples and less deviated samples.

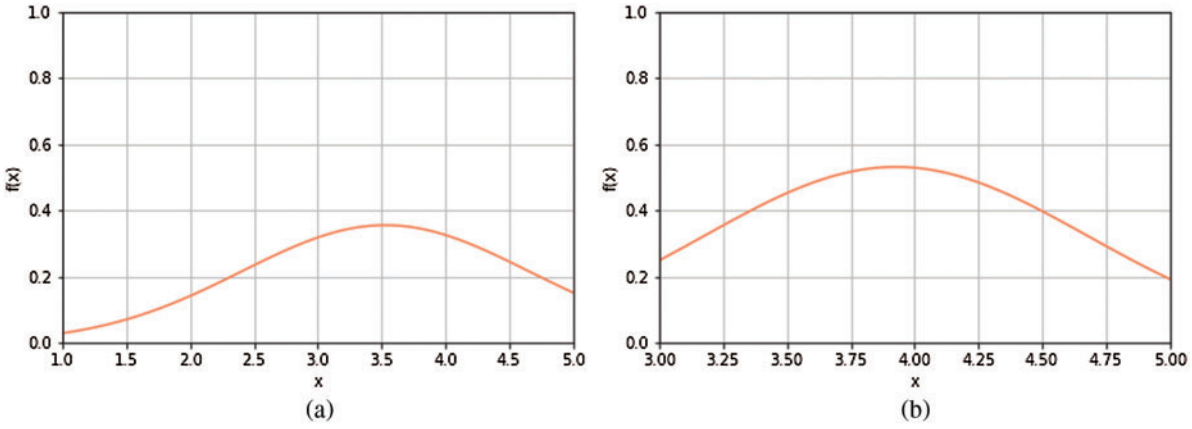

Figure 2: (a) Samples before applying truncation algorithm (b) Samples after applying truncation algorithm

The dataset selected by the truncation algorithm is divided into training for preparation of model and testing for the experiment in the ratio of 80 and 20 using a ten-fold cross-validation technique. Before using the clustering technique, the number of clusters to be decided using elbow method. In this technique, the number of clusters from 2 to 10 is assigned. The curve is plotted between the number of clusters and the inertia score, which is the sum of samples' squared distances in the closest cluster center. The number of clusters is chosen, where the point after which the inertia has started decreasing

linearly. Tab. 3 presents an inertia score for K values that varies from 2 to 9. The parameters used for clustering technique are shown in Tab. 4. The enhanced birch algorithm expressed in Section 3.4 gives three different clusters C1, C2, and C3.

Table 3: Elbow curve method using inertia score

K-value	Inertia score
2	80933.62
3	71617.69
4	71481.11
5	70501.14
6	69986.27
7	68117.22
8	67212.9
9	65963.53

Table 4: Hyperparameters list for BIRCH clustering algorithm

Parameter	Tuned value
Branching-factor	50
Compute-Labels	True
N-clusters	3
Threshold	0.5

The Grid search obtains the best parameters, which methodically build and estimate a model for each mixture of algorithm parameters specified in a grid. Hyper parameters are tuned using Grid search method for the gradient boost classification algorithm and it is listed in Tab. 5. The gradient boost classification tree models are created based on the clusters obtained from the enhanced birch algorithm, and it is named as M1, M2, and M3.

Table 5: Hyper parameters list for GBT algorithm

Parameter	Tuned value
Max-depth	4
Min-Sample-Split	8
Learning Rate	0.1
Loss	Deviance
n-estimators	100
Max-features	Log2

4.2 Results

The enhanced BIRCH clustering algorithm is used to test the samples, and the results are predicted and classified into the corresponding clusters such as C1, C2 and C3. The test samples are given to the related models such as M1, M2, and M3 and the prediction values are found by using MAE. Where, these values are recorded for the proposed models and tabulated in Tab. 6, which contains both input datasets.

Table 6: MAE values of the proposed model

Model	MAE(ml 100K)	MAE(ml 1M)
M1	0.5380	0.5641
M2	0.5450	0.5587
M3	0.5110	0.5927

Among these three models, model M3 shows better results and yields 0.52 as average. The experiment is performed without applying the proposed model, and the MAE values are tabulated in Tab. 7, which proves that the proposed model reduces the error value.

Table 7: MAE values after feature and instance selection

Model	MAE(ml 100K)	MAE(ml 1M)
M1	0.7690	0.7002
M2	0.7939	0.6133
M3	0.7157	0.6786

After finding the active user cluster, and the recommendations are made by removing the watched movies from the list using a top n recommendation algorithm as mentioned in Sections 3.5 and 3.6. The model is validated through the recommendation measures such as precision, recall and f-measure, which are explained in the Section 3.7. The recommended measures for each model are calculated and tabulated. Tab. 8 shows the recommender measures of the proposed model in movie lens 100k and 1M dataset.

Table 8: Recommendation measures of the proposed model

Input	Model	Precision	Recall	F-measure
Movielens 100K	M1	0.7770	0.8173	0.7966
	M2	0.9040	0.9123	0.9081
	M3	0.8250	0.8650	0.8442
Movielens 1M	M1	0.6787	0.8166	0.6440
	M2	0.8181	0.8333	0.6473
	M3	0.8888	0.9235	0.7081

4.3 Discussion

In this section, the MAE value of the proposed model is compared with the existing recommendation algorithms. Tab. 9 represents that the Mohammad pour et al. [11] achieved the minimum MAE value of 0.6610 and 0.8220 on Movielens 100k, and 1M datasets. On the other hand, the proposed model delivers 0.52 and 0.5718 MAE value on Movielens 100k and 1M datasets. The simulation results showed that the proposed model reduces 68% of average MAE before the truncation algorithm, which indicates that the proposed model produces less error, and high accuracy compared to the existing algorithm, and it is represented in Fig. 3. The Fig. 3 shows the comparison of MAE values with the existing models. It is determined that the proposed model gives better results than the existing model in both Movielens100k and 1M datasets and proved that the model gives consistent results. The proposed model gives minimum error values, due to feature selection and instance selection by a truncation algorithm.

Table 9: MAE value compared with an existing model

Model	Movielens 100k	Movielens 1M
Mohammad pour et al. [11]	0.6610	0.8220
Proposed Model	**0.5200**	**0.5718**

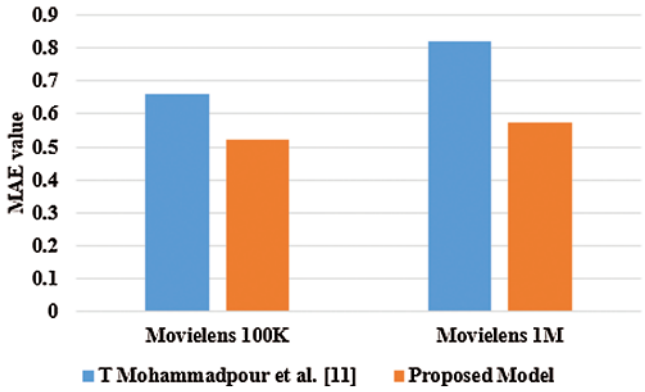

Figure 3: Graphical comparison of proposed and existing model in terms of MAE

In Tab. 10, the performance comparison is carried out between the proposed model and the existing recommendation system developed by Selvi et al. [12]. It is determined that the proposed model delivers better results than the existing model in light of precision, recall and f-measure on Movielens 100k dataset. In this section, the proposed model obtained 0.8350 of precision, 0.8640 of recall and 0.8672 of f-measure on Movielens 100k dataset, which are better compared to existing model that is graphically represented in the Fig. 4. The data sparsity is reduced using an enhanced BIRCH clustering through the deployed feature selection and instance selection algorithms. Scalability issue is addressed by implementing the truncation algorithm based on feature importance. The low MAE value and the high precision, recall, and f-measure values showed that the proposed GBT recommendation model performed well in movie recommendation.

Table 10: Recommendation measures comparison with existing model in terms of precision, recall, and f-measure

Model	Dataset	Precision	Recall	F-measure
Selvi et al. [12]	Movielens 100k	0.7300	0.7190	0.8439
Proposed Model	**Movielens 100k**	**0.8350**	**0.8640**	**0.8672**
Proposed Model	**Movielens 1M**	**0.7952**	**0.8578**	**0.6664**

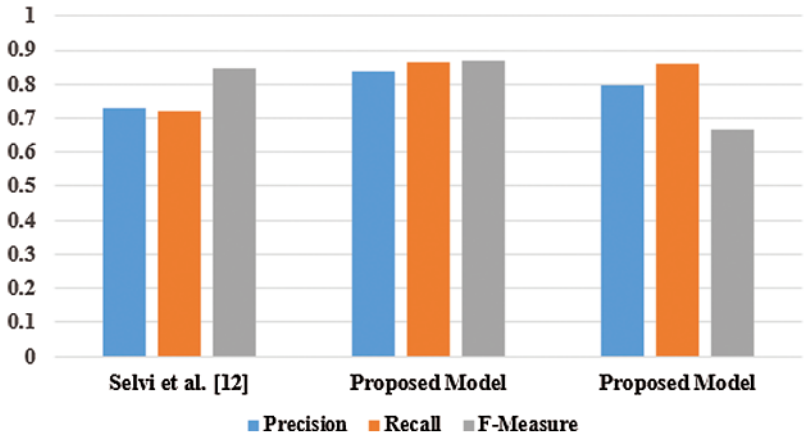

Figure 4: Graphical comparison of proposed and existing model in terms of precision, recall and f-measure

In Tab. 11, the proposed model is compared with two existing recommendation models, which are developed by Fu et al. [29] and Zhang et al. [30]. The existing models obtained 0.8300 and 0.9460 RMSE value on movielens 100k and movielens 1M datasets. Related to the existing models, the proposed model obtained batter RMSE value of 0.4392 and 0.4500 on movielens 100k and movielens 1M datasets.

Table 11: RMSE value compared with the existing models

Model	Movielens 100k	Movielens 1M
Fu et al. [29]	-	0.8300
Zhang et al. [30]	0.9460	-
Proposed Model	**0.4392**	**0.4500**

5 Conclusion

An ensemble collaborative recommendation model with a truncation algorithm is proposed for movie recommendation in this research. The proposed model is validated on two real-world datasets; Movielens 100k and 1M datasets. In the proposed model, feature selection using the significance of the feature plays an important role, truncation algorithm influences the ensemble model performance consistently, and the ensemble learning in collaborative filtering produces better results than the

existing models by means of recall, precision and f-measure. The prediction and recommendation performance measures showed that the proposed model is outperformed the existing methods in movie recommendation. The personalized recommender performance measure showed that the proposed model provides top recommendations to the active users. In the future work, we planned to design a recommendation model for big data environment, which is a complicated, engaging and challenging. It involves the recent tools and techniques to handle a massive amount of data.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] D. Goldberg, D. Nichols and D. Terry, "Using collaborative filtering to weave an information tapestry," *ACM Communications*, vol. 5 , pp. 61–70, 1992.

[2] J. Bobadilla, F. Ortega, A. Hernando and A. Guiterrez, "Recommender systems survey," *Journal of Knowledge Based Systems*, vol. 46 , pp. 109–132, 2013.

[3] S. Khusrom, Z. Ali and I. Ullah, "Recommender systems: issues, challenges, and research opportunities," in *Proc. Information Science and Applications (ICISA)*, vol. 376 , pp. 1179–1189, 2016.

[4] P. Thorat, M. Goudar and S. Barve, "Survey on collaborative filtering, content-based filtering, and hybrid recommendation system," *International Journal of Computer Applications*, vol. 110 , no. 4 , pp. 31–36, 2017.

[5] V. Vekariya and G. R. Kulkarni, "Hybrid recommender systems: survey and experiments," in *Proc. Int. Conf. on Digital Information and Communication Technology and Its Applications (DICTAP)*, vol. 12, no. 4 , pp. 331–70, 2012.

[6] M. Phung, N. Dung and L. Nguyen, "A model based approach for collaborative filtering," in *Proc. 6th Int. Conf. on Information Technology for Education*, Ho Chi Minh city, Vietnam, pp. 217–228, 2010.

[7] F. O. Isinkaye, Y. O. Folajimi and B. A. Ojokoh, "Recommendation systems: Principles, methods, and evaluation," *Egyptian Informatics Journal*, vol. 16 , no. 3 , pp. 261–273, 2015.

[8] C. Ju and C. Xu, "A new collaborative recommendation approach based on users clustering using artificial bee colony algorithm," *The Scientific World Journal*, vol. 2013 , pp. 1–9, 2013.

[9] J. Rongfei, J. Maozhong and L. Chao, "A new clustering method for collaborative filtering," in *Proc. Int. Conf. on Networking and Information Technology*, Manila, Philippines, pp. 488–492, 2010.

[10] J. Das, S. Majumder, P. Gupta and K. Mali, "Collaborative recommendations using hierarchical clustering based on k-d trees and quadtrees," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 27 , no. 04 , pp. 637–668, 2019.

[11] T. Mohammad pour, A. M. Bidgoli, R. Enayatifar and H. H. S. Javadi, "Efficient clustering in collaborative filtering recommender system: Hybrid method based on genetic algorithm and gravitational emulation local search algorithm," *Genomics*, vol. 111 , no. 6 , pp. 1902–12, 2019.

[12] C. Selvi and E. Sivasankar, "A novel optimization algorithm for recommender system using modified fuzzy c-means clustering approach," *Soft Computer*, vol. 23 , pp. 1901–1916, 2019.

[13] Z. Cataltepe, M. Uluyagmur and E. Tayfur, "Feature selection for movie recommendation," *Turkish Journal of Electrical Engineering & Computer Sciences*, vol. 24 , pp. 833–848, 2016.

[14] M. Ramezani, P. Moradi and F. A. Tab, "Improve performance of collaborative filtering systems using backward feature selection," in *Proc. 5th Conf. on Information and Knowledge Technology*, Shiraz, Iran, pp. 225–230, 2013.

[15] K. Yu, X. Xu, M. Ester and H. P. Kriege, "Selecting relevant instances for efficient and accurate collaborative filtering," in *Proc. ACM 10th Int. Conf. on Information and Knowledge Management (CIKM'01)*, Virtual Event , QLD , Australia, pp. 239–246, 2001.

[16] K. Yu, X. Xu, M. Ester and H. P. Kriegel, "Feature weighting and instance selection for collaborative filtering: An information-theoretic approach," *Knowledge and Information Systems*, vol. 5 , no. 2 , pp. 201–224, 2003.
[17] N. Pahuja and S. Chaudhari, "Uncovering significance of feature-selection in recommender system models," in *Proc. Global Conf. for Advancement in Technology (GCAT)*, Bangalore, India, pp. 1–5, 2019.
[18] C. Zeng, C. X. Xing, L. Z. Zhou and X. H. Zheng, "Similarity measures and instance selection for collaborative filtering," *International Journal of Electronic Commerce*, vol. 8 , no. 4 , pp. 115–129, 2004.
[19] A. L. Xu, B. J. Liu and C. Y. Gu, "A recommendation system based on eXtreme gradient boosting classifier," in *Proc. 10th Int. Conf. on Modelling, Identification and Control (ICMIC)*, Guiyang, China, pp. 1–5, 2018.
[20] Y. Shao and C. Wang, "HIBoosting: A recommender system based on a gradient boosting machine," *IEEE Access*, vol. 7 , pp. 171013–171022, 2019.
[21] S. Sathiya Devi and G. Parthasarathy, "Feature engineering based approach for prediction of movie ratings," *International Journal of Information Engineering and Electronic Business (IJIEEB)*, vol. 11 , no. 6 , pp. 24–31, 2019.
[22] Y. Yang and Y. Chen, "A normal distribution model for diffuse radiation versus incidence angle," *Journal of Solar Energy*, vol. 186 , pp. 60–71, 2019.
[23] M. A. Sakib, "An improved approximation algorithm for hierarchical clustering," *Pattern Recognition Letters*, vol. 104 , pp. 23–8, 2018.
[24] A. Israeli, L. Rokach and A. Shabtai, "Constraint learning based gradient boosting trees," *Expert Systems with Applications*, vol. 128 , pp. 287–300, 2019.
[25] B. Lorbeer, A. Kosareva, B. Deva, D. Softić and P. Ruppel *et al.*, "Variations on the clustering algorithm BIRCH," *Big Data Research*, vol. 11 , pp. 44–53, 2018.
[26] T. N. Nguyen, V. V. Le, S. I. Chu, B. H. Liu and Y. C. Hsu, "Secure localization algorithms against localization attacks in wireless sensor networks," *Wireless Personal Communications*, pp. 1–26, 2021.
[27] F. Maxwell Harper and A. Joseph, "Konstan the movielens datasets: History and context," *ACM Transactions on Interactive Intelligent Systems*, vol. 5 , no. 4 , pp. 19, 2019.
[28] C. H. Nguyen, T. L. Pham, T. N. Nguyen, C. H. Ho and T. A. Nguyen, "The linguistic summarization and the interpretability, scalability of fuzzy representations of multilevel semantic structures of word-domains," *Microprocessors and Microsystems*, vol. 81, pp. 103641, 2021.
[29] M. Fu, H. Qu, Z. Yi, L. Lu and Y. Liu, "A novel deep learning-based collaborative filtering model for recommendation system," *IEEE Transactions on Cybernetics*, vol. 49 , no. 3 , pp. 1084–1096, 2018.
[30] J. Zhang, Y. Wang, Z. Yuan and Q. Jin, "Personalized real-time movie recommendation system: Practical prototype and evaluation," *Tsinghua Science and Technology*, vol. 25 , no. 2 , pp. 180–191, 2019.