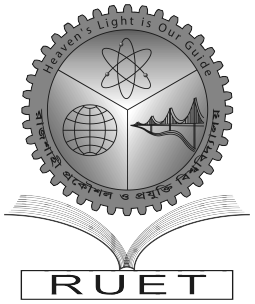
Heaven's Light is Our Guide



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Rajshahi University of Engineering & Technology, Bangladesh

Cold-Start Recommendation System Using K-Nearest Neighbor and Decision Tree in Movie Rating Prediction

Author

N. I. Md. Ashafuddula

Roll No. 123009

Department of Computer Science & Engineering

Rajshahi University of Engineering & Technology

Supervised by

Biprodip Pal

Assistant Professor

Department of Computer Science & Engineering

Rajshahi University of Engineering & Technology

ACKNOWLEDGEMENT

The real spirit of achieving goal is through the way of excellence and discipline. I would never be succeed to complete my task without the co-operation, encouragement and help provided to me by various personalities.

First of all, I render my gratitude to my Almighty Allah who bestowed self-confidence, ability and strength in me to complete this work preciously.

With deep sense of gratitude I express my sincere thanks to my esteemed and worthy Supervisor, **Biprodip Pal, Assistant Professor**, Department of Computer Science & Engineering, Rajshahi University of Engineering & Technology (RUET), Rajshahi, Bangladesh, for his valuable guidance in carrying out this work under his effective supervision, encouragement, enlightenment and co-operation. Most of the novel ideas and solutions found in this thesis are result of our numerous stimulating discussions. His feedback and editorial comments were also invaluable for writing of this thesis.

I want to express my gratitude thanks to **Dr. Md. Rabiul Islam, Professor, Head of the Department**, Computer Science & Engineering, Rajshahi University of Engineering & Technology (RUET), Rajshahi, Bangladesh, for his extending helps in various ways from his department. I also grateful to other teachers of the Department of Computer Science and Engineering, Rajshahi University of Engineering & Technology (RUET), Rajshahi, Bangladesh for their valuable and constructive suggestions, endless facilitation and inspiration.

Finally, yet importantly, I would like to express my heartfelt thanks to my beloved parents for their blessing, my family members for their help and wishes for the successful completion of this work.

December, 2017 **N. I. Md. Ashafuddula**

RUET, Rajshahi

Heaven's Light is Our Guide

Department of Computer Science & Engineering

Rajshahi University of Engineering & Technology, Bangladesh

CERTIFICATE

This is to certify that this thesis report entitled ***"Cold-Start Recommendation System Using K-Nearest Neighbor and Decision Tree in Movie Rating Prediction"*** *submitted by* ***N. I. Md. Ashafuddula, Roll 123009*** *in partial fulfillment of the requirement for the award of Bachelor of Science in Computer Science & Engineering from Rajshahi University of Engineering & Technology, Bangladesh is a record of the candidate own work carried out by him under my supervision. This thesis has not been submitted for the award of any other degree.*

Supervisor

_____

Biprodip Pal
Assistant Professor
Department of Computer Science & Engineering
Rajshahi University of Engineering & Technology
Rajshahi-6204

External Examiner

_____

Dr. Mir Md. Jahangir Kabir
Associate Professor
Department of Computer Science & Engineering
Rajshahi University of Engineering & Technology
Rajshahi-6204

ABSTRACT

Recommendation system (RS) is an information filtering system based on users different attributes and it's aim is to provide personalized recommendations to the users (e.g.movie, music, drama, song, books). To develop a recommender system Collaborative filtering (CF) and Content-based filtering (CB) are two most popular approaches.

The main challenge for a recommender systems is to provide the quality recommendation to the users in a cold-start situation. In this thesis, We consider one "cold-start" problem which is "Recommendation on existing items for new users" from Recommendation on existing items for new users, Recommendation on new items for existing users and Recommendation on new items for new users. Initially, the recommendation system, problems in a recommendation system, various algorithm to solve recommendation system in a cold-start situation are studied. Secondly, the problems in solving cold start problem with their proposed methodology are identified. The objectives of this thesis are as follows:

1. Classify new users based on their attributes
2. Recommend top rated item to the new users

To accomplish these objectives we have proposed a model where classification algorithms K Nearest Neighbor (K-NN) or Decision Tree (DT) combined with prediction mechanisms to provide the necessary means for retrieving recommendations. The proposed approach incorporates classifications methods in a CF systems while the use of demographic data helps to identify other users with similar behavior.

In this thesis, we build a movie rating prediction system based on K-NN and DT algorithm using K-means clustering algorithm for "cold-start" situation in a system. Then compare these two methods based on the MAE and RMSE value in a different number of cluster to evaluate better method in "cold-start" situation. Our experiment shows the performance of the proposed methodology through a large number of experiments. We have taken widely known movielens dataset provided by the grouplens organization. We reveal the advantages of the proposed solution by providing satisfactory numerical results in different experimental scenarios.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

Recommendation systems (RSs) technology currently used in many application domains. RSs can suggest items of interest to users based on their preferences. Such preferences could be retrieved either explicitly or implicitly. Generally recommendations are based on models built from item characteristics or users social environment. As an example, recommendations could be based on other users having similar characteristics (e.g., age, gender, occupation, country). The recommendation process is a complex process that combines information about users and the attributes of items.
This paper deals with solving cold-start problem by using K-Nearest Neighbor (KNN) and Decision Tree classification algorithm and comparison between these two methodologies.

1.2 Machine Learning

Machine learning is an essential part of artificial intelligence that is concerned with the design, analysis, implementation, and applications of programs that learn from experience [3]. In other words, Machine learning is the science of getting computers to act without being explicitly programmed. In the past decade, machine learning has given us self-driving cars, practical speech recognition, effective web search and vastly improved understanding of human genome. Machine learning is so pervasive today that we probably use it dozens of times a day without knowing it. Many researchers also think it is the best way to make progress towards human level AI. In this paper we will learn about the most effective machine learning techniques, and gain practice implementing them and getting them to work. Mainly there are three classes of learning:

i Supervised Learning

ii Unsupervised Learning

iii Reinforcement Learning

1.2.1 Supervised Learning

The learning where the algorithm generates a function that maps inputs to desire outputs. Supervised learning is the machine learning task of inferring a function from labeled training data. The training data consist of a set of training examples.

A supervised learning algorithm analyzes the training data and produces an inferred function, which can be used for mapping new examples. An optional scenario will allow for the algorithm to correctly determine the class labels for unseen instances. This requires the learning algorithm to generalize from the training data to unseen situations in a "reasonable" way. Here K-Nearest Neighbor (KNN) and Decision Tree (DT) classification algorithms are supervised learning.

1.2.2 Unsupervised Learning

Unsupervised learning is a type of machine learning algorithm used to draw inferences from datasets consisting of input data without labeled responses. The most common unsupervised learning method is cluster analysis, which is used for exploratory data analysis to find hidden patterns or grouping in data.
K-means clustering is an unsupervised method aims to create group of objects, or clusters, in such a way that objects in the same cluster are very similar and objects in different clusters are quite distinct.

1.2.3 Reinforcement Learning

Reinforcement Learning is a type of Machine Learning, and thereby also a branch of Artificial Intelligence. It allows machines and software agents to automatically determine the ideal behaviour within a specific context, in order to maximize its performance. Simple reward feedback is required for the agent to learn its behavior, this is known as the reinforcement signal. There are many different algorithms that tackle this issue. As a matter of fact, Reinforcement Learning is defined by a specific type of problem, and all its solutions are classed as Reinforcement Learning algorithms.

1.3 Data Mining

Data Mining is defined as extracting the information from the huge set of data. It discovers information within the data that queries and reports cannot effectively reveal. It allows user to analyze data from many different dimensions or angles, categorize it, and summarize the relationships identified.
Technically, data mining is the process of finding correlations or patterns among dozens of

fields in large relational databases.
Data mining process consists of three stages:

1. Initial exploration
2. Model building or pattern identification
3. Deployment (i.e., the application of the model to new data in order to generate predictions)

1.4 Motivation

Machine learning is a subfield of artificial intelligence that is concerned with the design analysis, implementation and applications of programs that learn from experience. Machine learning is classified as supervised learning and unsupervised learning.
Today, recommendation system is most popular to recommend items to the systems users, like youtube, facebook, amazon system. Now the main problem is cold-start problem when users or items are new in any system they don't have primary historical data to recommend users. This challenge motivated me to efficiently recommend items to new users.

1.5 Objectives of the thesis

- Reduce users features
- Finding number of possible cluster for users
- Cluster users based on their reduced features
- Classify new users based on their attributes
- Predict rating for users
- Efficient recommendation of the items with good rating

1.6 Organization of the thesis

Rest of the Thesis is organized as follows

Chapter 1: Introduction

This chapter introduces the basic concepts Machine Learning and it's types. It also discusses the objectives of this thesis work.

Chapter 2: Background

This chapter discusses the background of Machine learning, Recommendation Systems and some basic knowledge of our work.

Chapter 3: Literature Study

This chapter contains the literature study based on Clustering, Classification, KNN & DT Classification.

Chapter 4: Proposed Methodology

This chapter contains our proposed methodology to solve cold-start recommendation on movie rating prediction

Chapter 5: Result Analysis

This chapter deals result analysis.

Chapter 6: Conclusion

This chapter contains conclusion, limitation of K-NN, limitation of DT and future works.

1.7 Conclusion

In this chapter, the motivation, objectives and organization of the thesis are described. This chapter also contains some basic description about Machine Learning, Data Mining, Recommendation Systems (RSs), different type of RSs, Cold-start problem, Singular Value Decomposition (SVD), Elbow method and Motivation of this work.

Chapter 2

Literature Study

2.1 Introduction

This chapter consists of basic knowledge about recommendation system, types of recommendation system, main problem of recommendation system. This knowledge will help us to understand clearly about our proposed method and also before starting our proposed method we discussed about basic algorithms that we used in our proposed method such as clustering, classification the goal of clustering and classification, K-means algorithm, K-Nearest Neighbor (K-NN) algorithm & Decision Tree (DT) algorithm.

2.2 Recommendation System

Recommendation system is a information filtering system which can recommend based on different attributes. Example: youtube recommends us various items, Facebook recommends us new people, group etc.

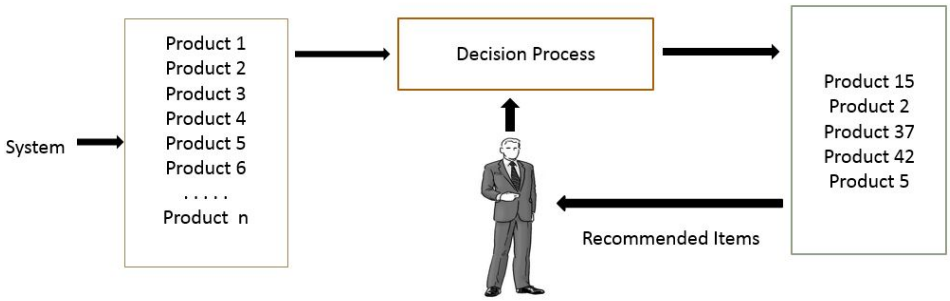


Figure 2.1: Recommendation process

Two different approaches are widely adopted to design recommendation system:

1. Content-based filtering and
2. Collaborative filtering

2.2.1 Content-based filtering

Content-based filtering generates a profile for user based on the content description on the items previously rated by the user. In this approach it is possible to recommend new items to the user which have not been rated by any users. But however, content-based filtering cannot provide recommendation to new users who does not have any historical ratings.

Content-based filtering often ask users to answer a questionnaire that explicitly states their preferences to generates initial profile of new user to provide new users recommendation. As a user consumes more items, her profile is updated and content features of items that she consumed will receive more weights. Content-based filtering one drawback is the recommended items are similar to the items previously consumed by the user. As example, if a user has seen only romance movies, then content-based filtering would recommend only romance movies. It often causes low satisfaction of recommendations due to lack of diversity for new or casual users who may reveal only small fraction of their interests. Another important limitation of content-based filtering is that its performance highly depends on the quality of feature generation and selection.

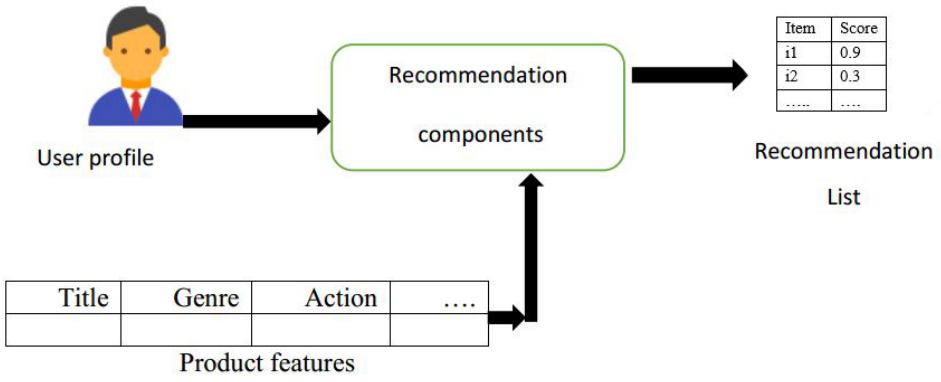


Figure 2.2: Content-based filtering process

2.2.2 Collaborative filtering

Collaborative filtering associates a user with a group of like-minded users, and then recommend items enjoyed by others user in the same group. This approach has few merits over content-based filtering approach. First, collaborative filtering does not require any feature generation and selection method and it can be applied to any domains if users rating (either explicit or implicit) are available. In other words collaborative filtering approach is content dependent. Second, collaborative filtering can provide "serendipitous finding", whereas content-based filtering cannot. As example, if most other romance movie fans love a comedy movie, even though a user has watched only romance movies, a comedy movie would be recommended to the user. Collaborative filtering captures this kind of relationship between items by analyzing user consumption history over the population. Content-based

filtering uses a profile of individual but does not exploit profiles of other users. Even though collaborative filtering performs better than content-based filtering when lots of user ratings are available, it suffers from cold-start problems where historical ratings on items or users are not available.

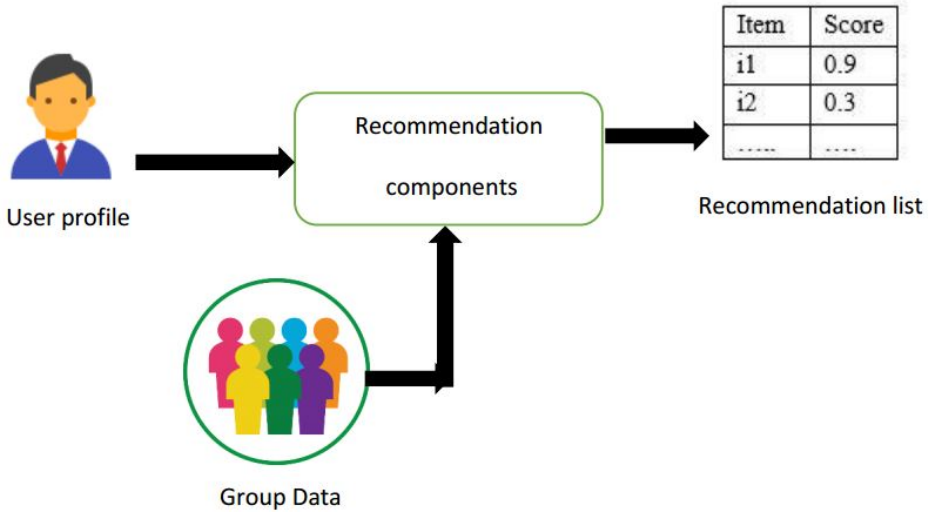


Figure 2.3: Collaborative filtering process

2.3 Cold-Start problem

The key challenge in any recommender systems including content-based and collaborative filtering is to provide recommendations at early stage when available data is extremely sparse. Since the col start problem is related to the sparsity of information (i.e., for users and items) available in the recommendation algorithm.

The problem is more severe at when the system newly launches and most of users and items are new. However, the problem never goes away completely, since new users and items are constantly coming in any healthy recommender system.

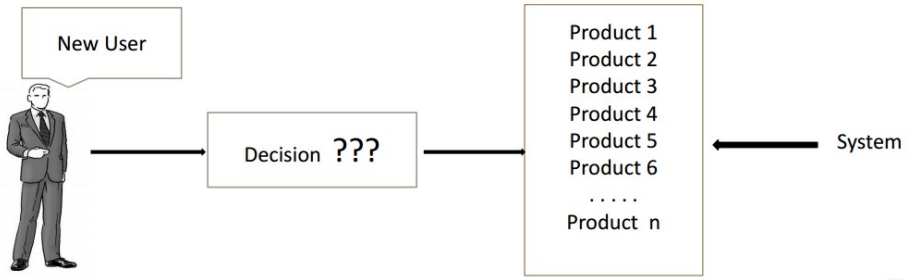


Figure 2.4: Cold-start problem

There are three types of cold start problems which can be occurred in a recommender system:

a) Recommendation for new users

b) Recommendation for new items

c) Recommendation for new items on new users

In this paper we focuses only on user side cold start problem.

We considers this situation where new user asks for recommendations and no data are available for her preferences. Such data are related to ratings for items. Ratings are very important here as they show the preferences of a specific user. Additionally, no historical data are present. We propose an algorithm which results the final outcome through three phases. The first phase is responsible to provide means for the classification of the new user in a specific group. For the classification, we adopt efficient techniques like the K-Nearest neighbor and the Decision Tree Algorithm. In the second phase, the algorithm utilizes an intelligent technique for finding the 'neighbors' of the new user. We examine important characteristics of the user and try to find other users inside the group that best match to her. In the third phase, the final outcome is calculated. This is done adopting prediction techniques for estimating the ratings of the new user.

2.4 Related Work

To build recommender systems two different approaches have been widely used: content-based filtering and collaborative filtering. Content-based filtering uses behavioral data about a user to recommend items similar to those consumed by the user in the past while collaborative filtering compares one user's behavior against a database of other users' behaviors in order to identify items that like-minded users are interested in.

The major difference between two approaches is that content-based filtering uses a single user information while collaborative filtering uses community information. Even though content-based filtering is efficient in filtering out unwanted information and generating recommendations for a user from massive information, it often suffers from lack of diversity on the recommendation. Content-based filtering requires a good feature generation and selection method while collaborative filtering only requires user ratings. Content-based filtering finds few if any coincidental discoveries while collaborative filtering systems enables

serendipitous discoveries by using historical user data. Hundreds of collaborative filtering algorithms have been proposed and studied, including K nearest neighbors [4, 5, 6], Bayesian network methods [7], classifier methods [8], clustering methods [9], probabilistic methods [10, 11], ensemble methods [12], and combination of KNN and SVD [13], Although collaborative filtering provides recommendations effectively where massive user ratings are available such as in the Netflix data set, it does not perform well where user rating data is extremely sparse. Several linear factor models have been proposed to attack the data sparsity. Singular Value Decomposition (SVD), Principal Component Analysis (PCA), or Maximum Margin Matrix Factorization (MMMF) has been used to reduce the dimensions of the user-item matrix and smoothing out noise [8, 14, 15]. However, those linear factor models do not solve the cold-start problems for new users or new items. Several hybrid methods, which often combine information filtering and collaborative filtering techniques, have been proposed. Fab [16] is the first hybrid recommender system, which builds user profiles based on content analysis and calculates user similarities based on user profiles. Basu et al. [17] generated three different features: collaborative features (i.e. users who like the movie X), con-tent features, and hybrid features (i.e. users who like comedy movies). Then, an inductive learning system, Ripper, is used to learn rules and rule-based prediction was generated for the recommendation. Claypool et al. [18] built an online news-paper recommender system, called Tango, that scored items based on collaborative filtering and content-based filtering separately. Then two scores are linearly combined: As users provide ratings, absolute errors of two scores are measured and weights of two algorithms are adjusted to minimize error. Good et al.[19] experimented with a number of types of filterbots, including Ripper-Bots, DGBots, Genre-Bots and MegaGenreBot. A filterbot is an automated agent that rates all or most items algorithmically. The filterbots are then treated as additional users in a collaborative filtering system. Park et al.[20] improved the scalability and performance of filterbots in cold-start situations by adding a few global bots instead of numerous personal bots and applying item-based instead of user-user collaborative filtering. Melville et al.[21] used content-based filtering to generate default ratings for unrated items to make a user-item matrix denser. Then traditional user-user collaborative filtering is performed using this denser matrix. Schein et al.[22] extended Hofmann's aspect model to combine item contents and user ratings under a single probabilistic frame work. Even though hybrid approaches potentially improve the quality of the cold-start recommen-

dation, the main focus of many hybrid methods is improving prediction accuracy over all users by using multiple data rather than directly attacking the cold-start problem for new users and items. Note that all above approaches only lessen the cold-start problem where a target user has rated at least few ratings but do not work for new user or new item recommendation. There are a few existing hybrid approaches which are able to make new user and new item recommendation. [23] proposed an online perceptron algorithm coupled with combinations of multiple kernel functions that unify collaborative and content-based filtering. The resulting algorithm is capable of providing recommendations for new users and new items, but the performance has not been studied yet. The computational complexity in the proposed kernel machine scales as a quadratic function of the number of observations, which limits its applications to large-scale data sets. Agarwal and Merugu [24] proposed a statistical method to model dyadic response as a function of available predictor information and unmeasured latent factors through a predictive discrete latent factor model. Even though the proposed approach can potentially solve the cold-start problems, its main focus is improving quality of recommendation in general cases and its performance in cold-start settings is not fully studied yet. Chu and Park [25] proposed a predictive bilinear regression model in "dynamic content environment" where the popularity of items changes temporally, lifetime of items is very short (i.e. few hours), and recommender systems are forced to recommend only new items. In the paper "Accuracy and Diversity in Cross-domain Recommendations for Cold-start Users with Positive-only Feedback" [26] they evaluates recommendation methods on a dataset with positive-only feedback in the movie and music domains, both in single and cross-domain scenario. They used two datasets (positive only Feedback) consisting of (a)Facebook likes on movies and (b)music artists. Problems in the paper was the results depends on the target domain and they did not analysis for more domains. In paper "Attack Resistant Collaborative Filtering" [27] they discuss about (a)Robust CF algorithm which is stable against moderate shilling attacks on large datasets (b) Leverages the accuracy of PCA-based attack detection. They used MovieLens dataset. The problem was (a)Shillings attacks are concentrated in a short period of time as opposed to real users (b) Need to devise more accurate ways of detecting the cutoff parameter to save flagged users from any potential impact. In paper "Learning Bidirectional Similarity for Collaborative Filtering" [28] the proposed a novel model learning user and item similarities simultaneously for collaborative filtering. They used Movielens, Eachmovie and

Netflix dataset. Their problem was to learn model in larger scale datasets. In paper "Social Collaborative Filtering for Cold-start Recommendations" they discuss about Cold-start recommendation task in an online retail setting for users who have not yet purchased any available items but who have granted access to limited side information, such as basic demographic data (gender, age, location) or social network information (Facebook friends or page likes). The dataset they used in this study comes from Kobo Inc. a major online ebook retailer with more than 20 million readers. It contains an anonymized dataset of ebook purchases and Facebook friends and page likes for a random subset of 30,000 Kobo users. Problem was they did not examine, if it can be extended with learning-based techniques such as collective matrix factorization.

2.5 Singular Value Decomposition

The singular-value decomposition (SVD) is a factorization of a real or complex matrix. It is the generalization of the eigen decomposition of a positive semi definite normal matrix (for example, a symmetric matrix with positive eigenvalues) to any matrix via an extension of the polar decomposition.
In this work I used SVD to reduce users feature.

$$A_{[m\times n]} = U_{[m\times r]}\Sigma_{[r\times r]}V^T_{[n\times r]} \tag{2.1}$$

2.6 Elbow Method

Using the elbow method to determine the optimal number of clusters for k-means clustering. K-means is a simple unsupervised machine learning algorithm that groups a dataset into a user-specified number (k) of clusters.

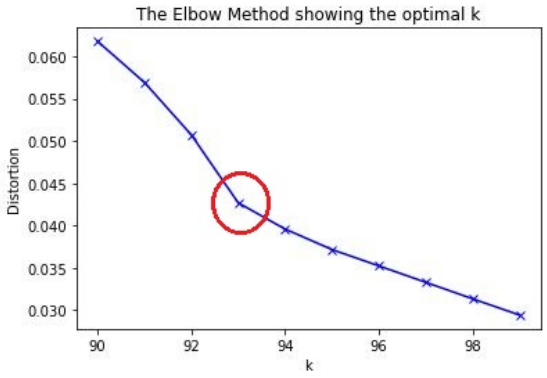


Figure 2.5: Elbow curve

2.7 Clustering

Clustering is the assignment of objects into subsets (called clusters) so that object in the same cluster are similar and dissimilar to objects of other groups. Clustering is a method of unsupervised learning, and a common technique for statistical data analysis used in many field. The goal of good document clustering schemes is to minimize intra-cluster distance distances between documents, while maximizing intra-cluster distances. Distance measurement is the heart of the document clustering.

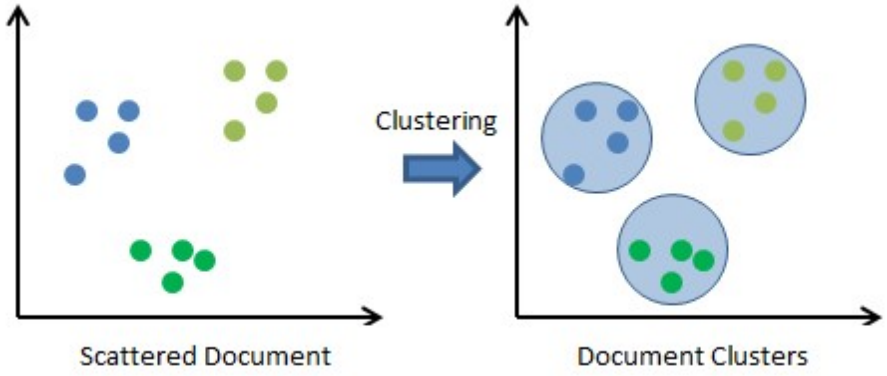


Figure 2.6: Example of clustering[1]

2.8 The goal of the clustering

The goal of clustering is to identify distinct groups in a dataset. The basic idea of model-based clustering is to approximate the data density by a mixture model, typically a mixture of Gaussians, and to estimate the parameters of the component densities, the mixing frac-

tions, and the number of components from the data. The number of distinct groups in the data is then taken to be the number of mixture components, and the observations are partitioned into clusters (estimates of the groups) using Bayes' rule. If the groups are well separated and look Gaussian, then the resulting clusters will indeed tend to be "distinct" in the most common sense of the word - contiguous, densely populated areas of feature space, separated by contiguous, relatively empty regions. If the groups are not Gaussian, however, this correspondence may break down; an isolated group with a non-elliptical distribution.

2.9 Classification

Classification is a technique to classify input data and label them from discrete set of possible values.

2.10 The goal of the classification

The goal of the classification is to take input data from dataset and predict category for those data from a discrete set of possible values. Example, Classifying emails as spam or not, giving a diagnosis for patient, given a set of symptoms, deciding if a user watching romance movie only or action movie or what type of category he falls.

2.11 Clustering vs. Classification

Classification can be defined as the task of assigning instances to pre-defined classes. As an example we can decide whether a particular patient record can be associated with a specific disease. On the other hand cluster is the task of grouping related data points together without labeling them. Example, grouping similar symptoms without knowing the symptoms indicate.

2.12 K-Means Clustering

The k-means clustering algorithm is known to be efficient in clustering large data sets. This clustering algorithm was developed by MacQueen, and is one of the simplest and the best known unsupervised learning algorithms that solve the well-known clustering problem. The K-Means algorithm aims to partition a set of objects, based on their attributes/features, into

k clusters, where k is a predefined or user-defined constant. The main idea is to define k centroids, one for each cluster. The centroid of a cluster is formed in such a way that it is closely related (in terms of similarity function; similarity can be measured by using different methods such as cosine similarity, Euclidean distance, Extended Jaccard) to all objects in that cluster.

2.13 K-Means Clustering Algorithm

Algorithm 1: The k-means clustering algorithm[1]

Step 1: Choose k number of clusters to be determined

Step 2: Choose k objects randomly as the initial cluster center

Step 3: Repeat

Step 4: Assign each object to their closest cluster

Step 5: Compute new clusters, i.e. Calculate mean points.

Step 6: Until

Step 7: No changes on cluster centers (i.e. Centroids do not change location any more) OR

Step 8: No object changes its cluster (We may define stopping criteria as well)

2.14 K-Nearest Neighbor (KNN) Classification

In pattern recognition, the k-nearest neighbor algorithm (K-NN) is a non-parametric method used for classification and regression. In both cases, the input consists of the k closest training examples in the feature space. In K-NN classification, the output is a class membership. The simple version of the K-nearest neighbor classifier algorithms is to predict the target label by finding the nearest neighbor class. The closest class will be identified using the distance measures like Euclidean distance.

2.15 K-Nearest Neighbor (KNN) Classification Algorithm

Algorithm 2: The K-NN Algorithm[29]

Step 1: Calculate "d(x, xi)", i =1, 2, ..., n; where d denotes the Euclidean distance between the points.

Step 2: Arrange the calculated n Euclidean distances in non-decreasing order

Step 3: Let k be a +ve integer, take the first k distances from this sorted list.

Step 4: Find those k-points corresponding to these k-distances.

Step 5: Let k_i denotes the number of points belonging to the i^{th} class among k points i.e. k ≥ 0

Step 6: If $k_i > k_j \; \forall \; i \neq$ j then put x in class i.

2.16 Decision Tree (DT) Classification

Decision Tree algorithm belongs to the family of supervised learning algorithms. Unlike other supervised learning algorithms, decision tree algorithm can be used for solving regression and classification problems too.

The general motive of using Decision Tree is to create a training model which can use to predict class or value of target variables by learning decision rules inferred from prior data (training data).

The understanding level of Decision Trees algorithm is so easy compared with other classification algorithms. The decision tree algorithm tries to solve the problem, by using tree representation. Each internal node of the tree corresponds to an attribute, and each leaf node corresponds to a class label.

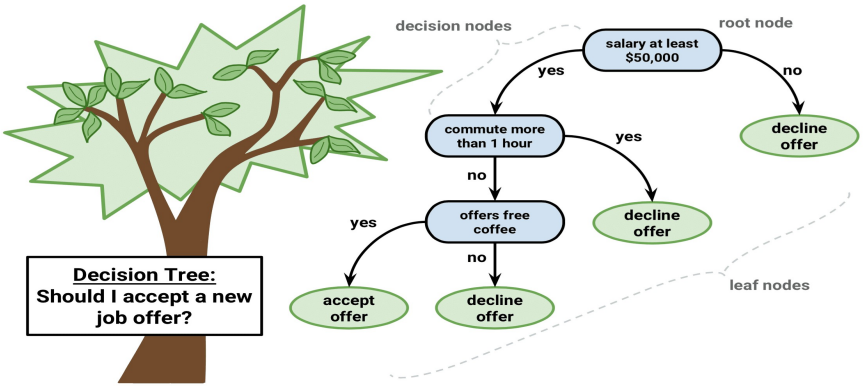


Figure 2.7: Decision Tree classifier[2]

2.17 Decision Tree (DT) Algorithm

Algorithm 3: The Decision Tree algorithm[30]

Step 1: Create a node N.

Step 2: If tuples in D are all of the same class, C then

Step 3: return N as a leaf node labeled with class C.

Step 4: If attribute_list is empty then

Step 5: return N as a leaf node labeled with the majority class in D.

Step 6: Apply Attribute_selection_method(D,attribute_list) to find ”best splitting_criterion”.

Step 7: Label node N with splitting_criterion.

Step 8: If splitting_attribute is discrete-valued and

Step 9: multiway splits allowed then

Step 10: attribute_list ← attribute_list – splitting_attribute.

Step 11: For each outcome j of splitting_criterion

Step 12: Let D_j be the set of data tuples in D satisfying the outcome j.

Step 13: If D_j is empty then

Step 14: attach a leaf labeled with the majority class in D to node N.

Step 15: else attach the node returned by generate_decision_tree (D_j, attribute_list) to

Step 16: node N.

Step 17: endfor

Step 18: return N.

2.18 Conclusion

In this chapter, we discussed basic knowledge such with singular valued decomposition (SVD), Elbow Method and related work of recommendation system & cold-start recommendation system then discussed about the basic knowledge about clustering, classification, K-NN & DT classifier. This chapter contains basic knowledge before starting our proposed approach.

Chapter 3

Proposed Methodology

3.1 Introduction

This chapter briefly discussed about our proposed methodology using previously gathered knowledge such as Elbow method, K-means clustering, K-NN and DT classification.

3.2 Proposed Methodology

In this section we propose an approach based on K-Nearest Neighbor (KNN) and Decision Tree (DT) classification. Our key idea is to build a predictive model for user/item pairs by leveraging all available information of users and items, which is particularly useful for cold-start recommendation including new user recommendation. The proposed model alleviates the user cold start problem of CF. The main operational aspect are depicted in Fig: 3.1

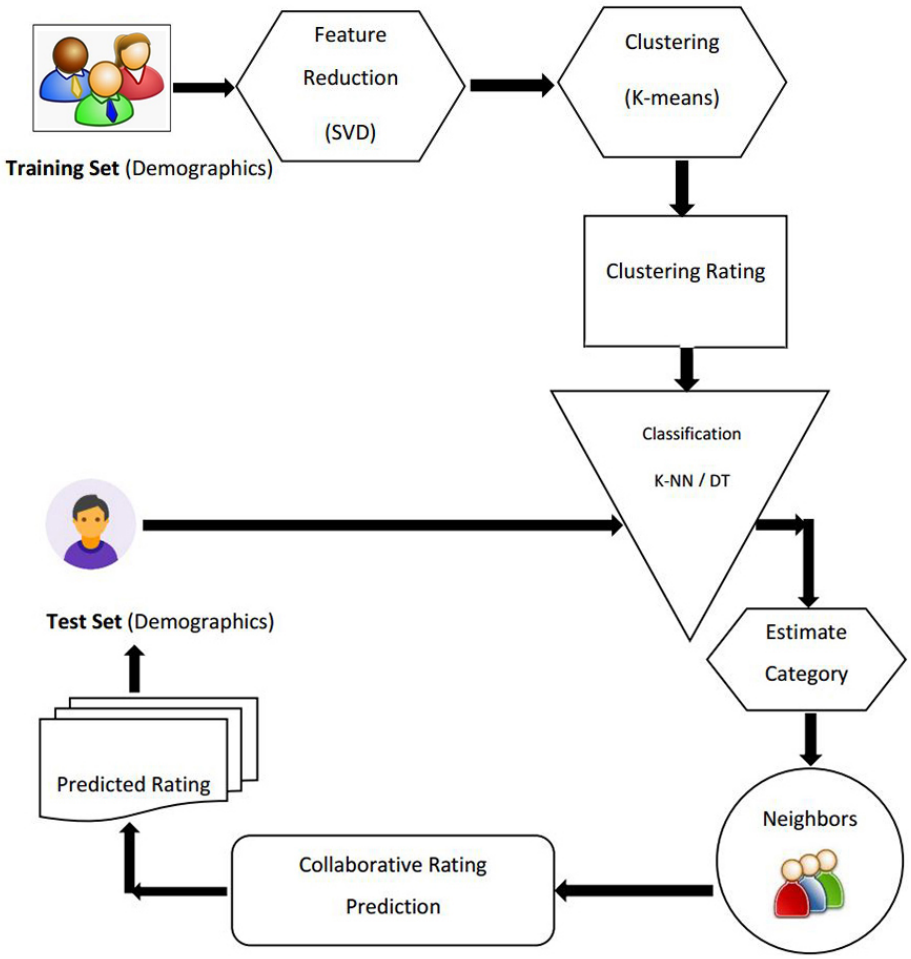


Figure 3.1: The architecture of proposed system

As an example, we can assume a system, where a group of users and items are exists. Suppose, after some times, a group of new registered users are identified & system has to perform item recommendation for these new users. Where the system has only their demographic data such as age, gender, country, etc.

In our proposed method, system first reduces the dimension of existing users using SVD. Then using their demographic data apply elbow method to find number of possible cluster or group. After identifying number of cluster apply K-Means cluster on existing users data. Now system classifies the existing users using DT or KNN approach.

For new users system perform DT or KNN classification and classify them to Estimate their corresponding category. After finding their category, from the category the group of neighbors likely items are collected. Using those likely items whose ratings are 3 or above 3 (rating in scale of 0 to 5) will be recommended to the new users.

The process of predicting item rating for a new user involves three phases. Let the set of current users in the system be, $U = \{u_1, u_2, \ldots\ldots, u_m\}$ and $N = \{n_1, n_2, \ldots\ldots, n_n\}$ be the set of the new users. Moreover, a set, $I = \{i_1, i_2, \ldots.., i_k\}$ of items is available. At first, we build a model based on demographic data, $D = \{d_1, d_2, \ldots.., d_i\}$ (D is defined by developers) and users preferences. We name this step 'Classification'. People with a common background are much likely to have similar preferences. The classification component implements a model on the basis of a training set which is the reduction of current users feature and contains instances of the whole data store. Instances include variables related to D. Then, we use the generated model to map a new observation in the appropriate category C. C belongs in the set of categories, $C = \{c_1, c_2, \ldots.., c_b\}$ In the Fig: 3.1, we show two key factors: (a) the prediction variable V, and (b) the estimated category C. For each new observation, $O_j, j = \{1, 2, \ldots...n\}$ We set a class attribute that represents V. Value of this class are possible categories U, $c_j \subseteq C$ for each new O_j . One of these categories c_j is the output of the output of the model and the corresponding category C for every $n \in N$. The neighbors in NG are users that belong to the same category as the model predicts.

Second, after the selection of NG, we calculate the similarity between $n \in N$ and each of the neighbours $u_j \in NG$, j = 1,2,.... $|NG|$ through demographic data. We name this as step 'Similarity Estimation'. In this phase we incorporate a similarity function that combine similarity from $d_j \in D$, j = 1,2,3,.... $|D|$.

Finally, we make predictions combining the similarity measure and neighbors ratings. We

name this step as 'Collaborative Prediction'. This component implements a function that makes a prediction for an item $i \in I$. the prediction is derived by a weighted average of each u_j ratings. More specially, we combine the similarity weights calculated in the previous phase with the ratings of neighbors for the possible recommended item.

3.3 Dataset

In this work we used Movielens ml-100k dataset[31]. Which is stable benchmark dataset with 100,000 ratings (1-5) from 943 users on 1682 movies and released in 4/1998.

- Training set with 3/4 users data
- Test set with 1/4 users data
- 5 User's Attributes

Table 3.1: Users attribute in dataset

User Attributes	Movie Features
Gender	2
Age	7
Occupation	21
Constant Feature	1
Location	0

Table 3.2: Relationship of user id, movie id with ratings in dataset

User Id	Movie Id	Rating
1	201	4
3	111	3.5
91	102	5
20	210	3
23	91	4

3.4 User classification and Neighbor finding

Through the classification algorithms, we are able to produce category C, based on the data related to the set U. In order to have the final C, we applied multi-class classifiers which gives us opportunity to have multiple class in the results. To do this, at first we reduced the current existing users (demography) features using Singular Value Decomposition (SVD) in economy size Equation 2.1. Then using the reduced feature we have trained the classification algorithms. We use here in K Nearest Neighbor (KNN) Fig: 3.2, and Decision Tree (DT) Fig: 3.3. To achieve the number of class for a specific group we use clustering algorithm then applied classification algorithm. Here we use Algorithm: 1, K-Means clustering.

For predicting new Category C_j of a new user, we used this trained classification algorithm KNN and DT which gives us related classification or users category, In the KNN algorithm we use Euclidean distance.

In general, the distance between points x and y in a Euclidean space R^n is given by Equation Eq. 3.1

$$d = \sqrt{\sum_{i=1}^{n} |X - Y|^2} \tag{3.1}$$

Depending on value of matrix $\Sigma_{[r\times r]}$ (strength / diagonal matrix) choose 2 features from current users features.

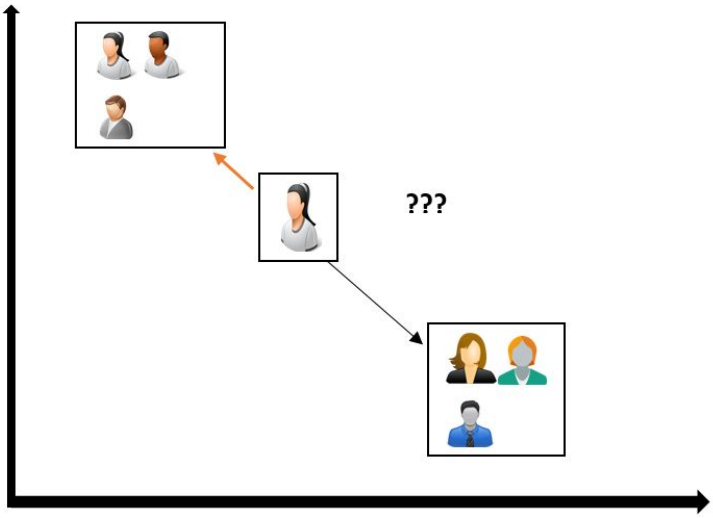


Figure 3.2: KNN Neighbor finding

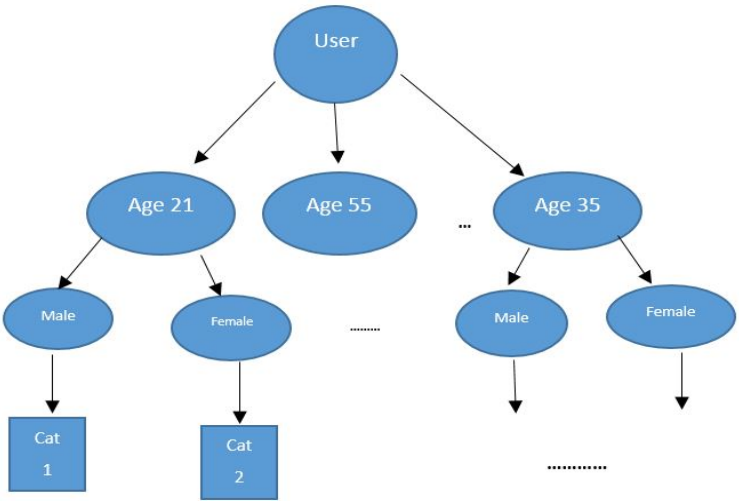


Figure 3.3: Decision Tree (DT)

3.5 Rating prediction

The final phase is to produce rating predictions for new users. For each user $n_j \in N$, the model should provide predicted ratings for every new user. That the items seen by the same category by each current user which has highest average rating and rating should be done by at least 5 user in the category.

Therefore the following equation for rating prediction.

$$r_{n_j,i_b} = \sum_{i=1}^{K} R_c / \sum_{i=1}^{K} N \tag{3.2}$$

For $n_j \in C$, where K = number of user in a class who are rated item i_b and R_c is the rating of item i_b. r_u, i_b is the rating of the user u for item i_b. Based on the above approach, we aim to enhance rating that are made by users having large similarity degree with every new user. This is as expected as users having in common a lot of characteristics probably they will have similar item preferences.

3.6 Conclusion

In this chapter, we briefly discussed about our proposed two methodologies, dataset and finally rating prediction for new user in the system. The main challenge in K-NN was to decide the value of K and it was slower than DT approach. In the K-NN approach we consider K = 5.

Chapter 4

Result Analysis

4.1 Introduction

This chapter contains the performance report of our proposed model. We define certain performance metrics and then present our results. Our aim is to quantify the performance of the proposed model concerning the prediction accuracy and compare the MAE & RMSE results obtained by using different classification algorithms (K-NN & DT).

4.2 Metrics Evaluation

Widely used metrics for prediction accuracy. The first metrics is the Means Absolute Error (MAE). MAE is defined in Eq. 4.1. In this equation, p_u, i defines the prediction for user u and for item I while r_u, i symbolizes the actual rating. Finally K is the symbolize number of items under evaluation.

$$MAE = 1/k \sum_{u,i} |p_u, i - r_u, i| \tag{4.1}$$

Another important metric is the Root Mean Square Error (RMSE) defined as Eq. 4.2

$$RMSE = \sqrt{1/k \sum_{u,i} (p_u, i - r_u, i)^2} \tag{4.2}$$

Both metrics are widely used in evaluating recommender systems with respect to prediction accuracy.

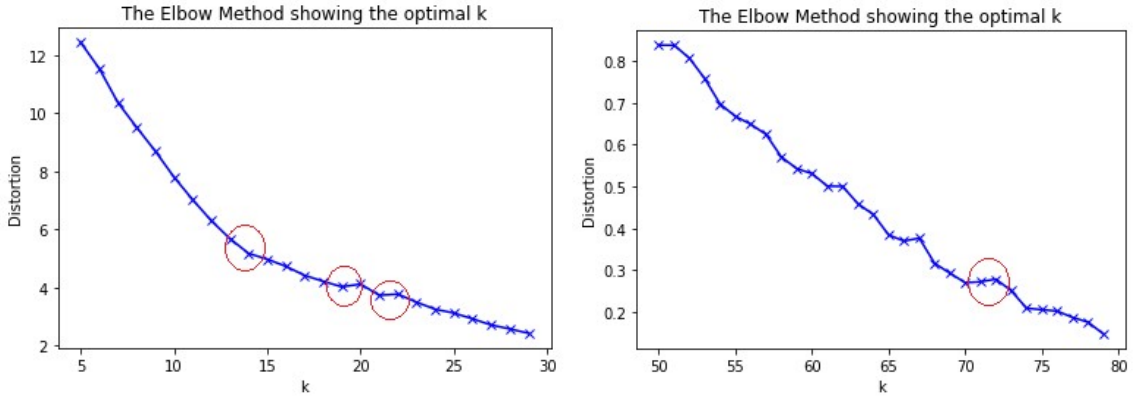
4.3 Result Evolution

We run a number of experiments for a specific dataset. The dataset is retrieved by the grouplens research team[31]. Grouplens provides the Movielens dataset containing 1000000 ratings for 1682 movies by 943 users. From the set of users we choose a number of user who are registered consider as current / existing users others are considered as new users in the system. Through this approach, we try to find out how the system behaves for different numbers of registered users.

4.3.1 RMSE of rating prediction obtained by KNN and DT algorithm

From Elbow method we choose cluster 13,19,23,71. Though curve shows better elbow at 14 we choose lower number 13 as cluster number fig 4.1a. And for fig 4.1b, we try to choose k for which system gives us better performance so though 65 and 67 shows better

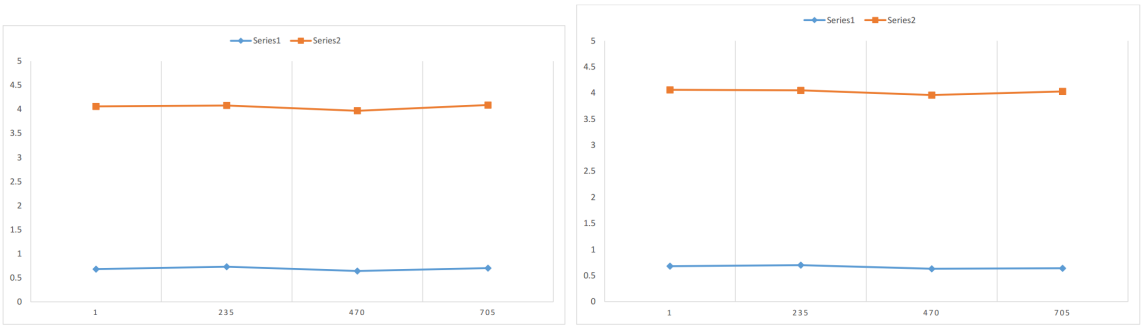
elbow we used 71. Thus we can compare the results behavior of MAE & RMSE for lower number of cluster & with higher number of cluster.



(a) for k (5 - 30) (b) for k (50 - 80)

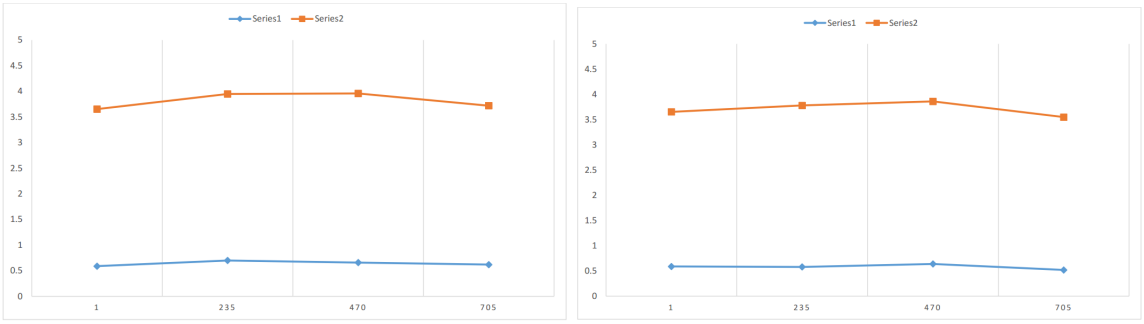
Figure 4.1: Identification number of cluster from Elbow curve

For different cluster number we predict rating and measure MAE, maximum MAE, RMSE & maximum RMSE then plot them, thus we know the system performance for K-NN & DT approaches.



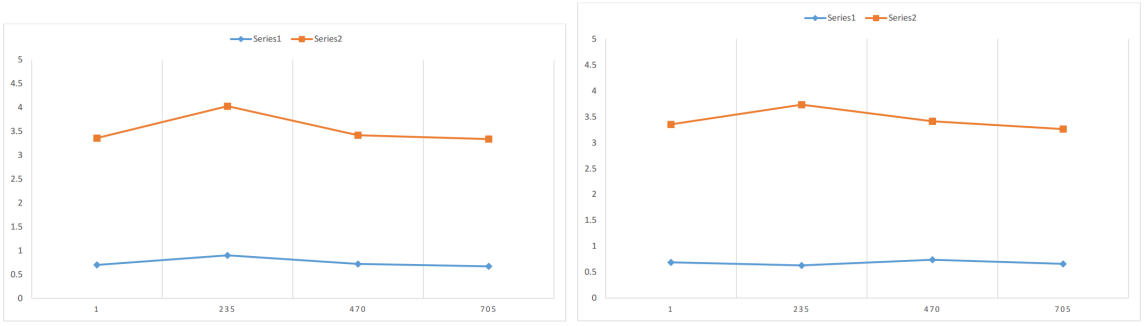
(a) Users (KNN approach) (b) Users (DT approach)

Figure 4.2: MAE Result scenario, Error vs Users for 13 category



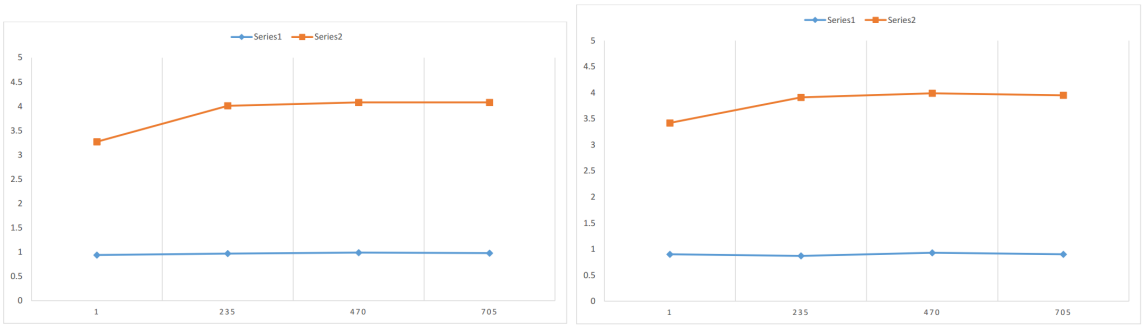
(a) Users (KNN approach) (b) Users (DT approach)

Figure 4.3: MAE Result scenario, Error vs Users for 19 category



(a) Users (KNN approach) (b) Users (DT approach)

Figure 4.4: MAE Result scenario, Error vs Users for 23 category



(a) Users (KNN approach) (b) Users (DT approach)

Figure 4.5: MAE Result scenario, Error vs Users for 71 category

Ratings are between 1 (minimum value) and 5 (maximum value). All ratings are integer values. For each user, we have taken the identification number and her demographic data $D = \{d1, d2, d3, d4, d5\} = \{age, occupation, gender, constant\,feature, country\}$ as primarily no other data from new users are available in the system.

4.4 Comparison between K-NN and DT approach

We conclude comparison between K-NN and DT approaches mainly depending on average of MAE in K-NN and DT approaches and also calculate RMSE.

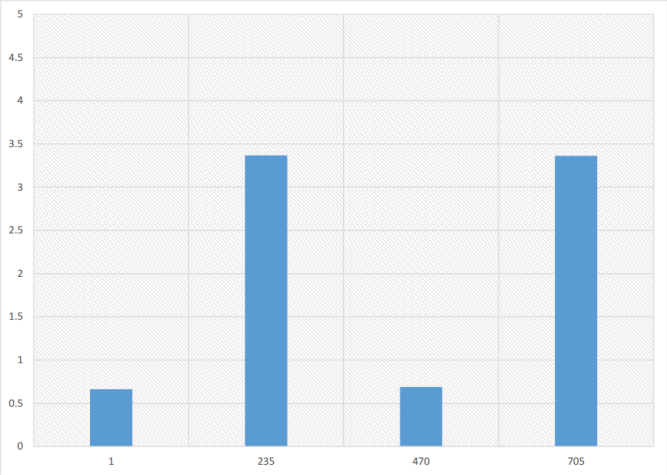


Figure 4.6: MAE Comparison between DT and KNN for category 13

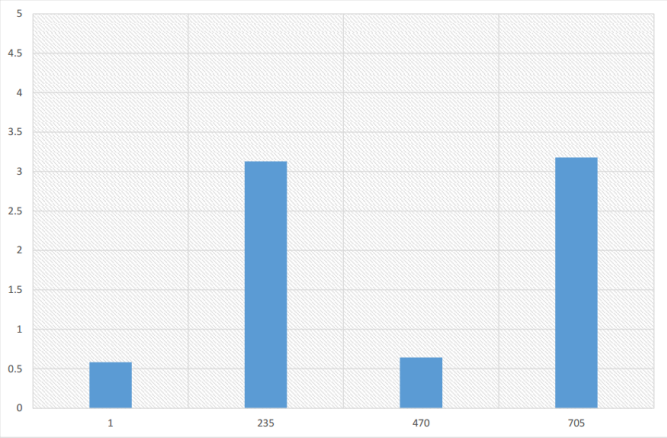


Figure 4.7: MAE Comparison between DT and KNN for category 19

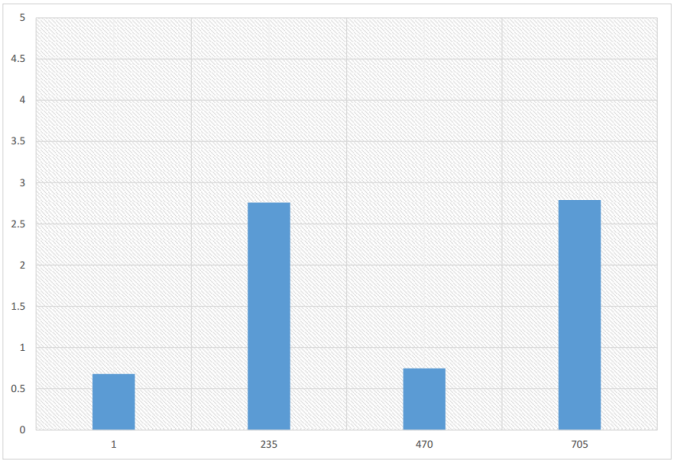


Figure 4.8: MAE Comparison between DT and KNN for category 23

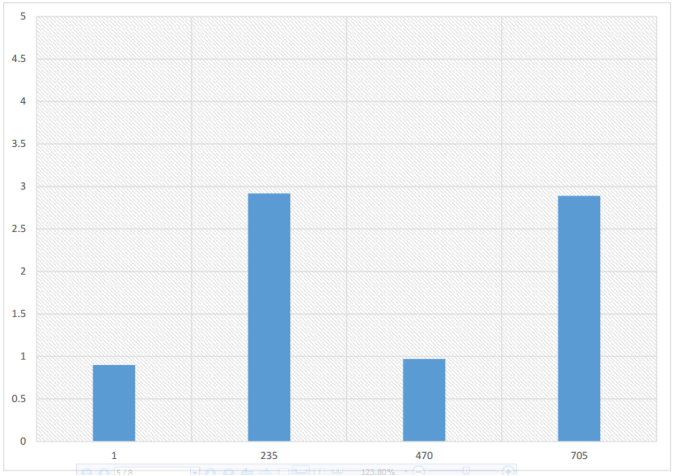


Figure 4.9: MAE Comparison between DT and KNN for category 71

Comparing KNN and DT for different category from the bar graph we can see that for 19 cluster both shows better MAE. The final comparison between these two approaches for MAE & RMSE are shown in fig: 4.11 & fig: 4.10 respectively.

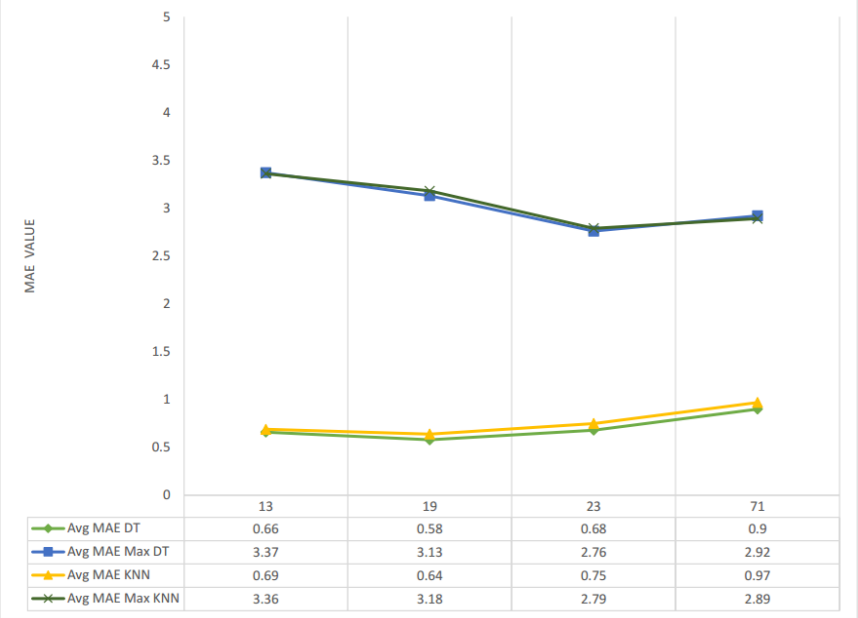


Figure 4.10: Final MAE comparison between KNN and DT

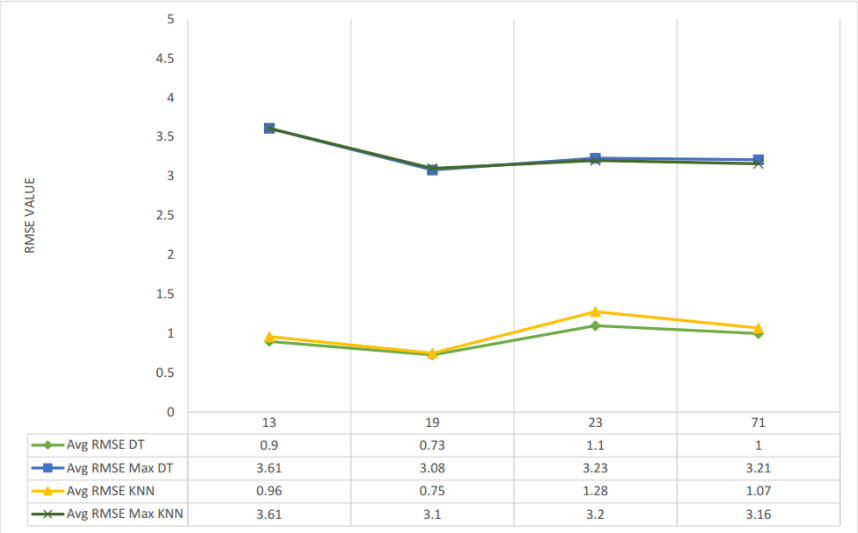


Figure 4.11: MAE Final MAE comparison between KNN and DT

4.5 Resultant discussion

We followed the following result scenario. In which in the result graph KNN and DT for different category we compare the two technique with 4 validation user data. $|u| = users$, and the corresponding average MAE & RMSE value. The MAE & RMSE value scaled between 0 to 5 and we showed that in the above graph which defines the max MAE & RMSE value. We calculate MAE from following Eq. 4.1 and then calculate RMSE from

following Eq. 4.3.

$$RMSE = \sqrt{1/n \sum_{i=1}^{n} (y_i - Y_i)^2} \tag{4.3}$$

$where, y_i = predicted rating, Y_i = actual rating and n = number of user.$

After the result scenario we conclude the comparative result between KNN and DT for different user's category level. Then we finally reached in the final comparison between DT and KNN. After that we calculate standard deviation for different number of user's to see how spread the data around the mean, where RMSE is used to measure distance between prediction and actual rating for items. We calculate standard deviation using Eq. 4.4.

$$\sigma = \sqrt{1/n \sum_{i=1}^{n} (a_i - A_i)^2} \tag{4.4}$$

where, a_i = average RMSE for a category, Ai = RMSE for a category for a number of user and n = number of user.

Table 4.1: MAE Result table for DT

Category No.	Average MAE	Max MAE	Standard Deviation
13	0.66	3.36	±0.03
19	0.58	3.13	±0.04
23	0.68	2.76	±0.04
71	0.90	2.92	±0.02

Table 4.2: MAE Result table for KNN

Category No.	Average MAE	Max MAE	Standard Deviation
13	0.69	3.36	±0.03
19	0.64	3.17	±0.04
23	0.75	2.79	±0.09
71	0.97	2.89	±0.02

Table 4.3: RMSE Result table for DT

Category No.	Average RMSE	Max RMSE	Standard Deviation
13	0.90	3.61	±0.12
19	0.73	3.08	±0.076
23	1.10	3.22	±0.036
71	0.99	3.21	±0.17

Table 4.4: RMSE Result table for KNN

Category No.	Average RMSE	Max RMSE	Standard Deviation
13	0.96	3.61	±0.24
19	0.75	3.10	±0.08
23	1.28	3.20	±0.07
71	1.08	3.16	±0.174

In order to verify the effectiveness of our proposed algorithm, comparative experiments among the proposal method, Social network sub-community and ontology decision mode (SSODM)[32], MODE method and Average method. $1/4$ users are taken as new users, respectively the other methods for experimental testing to compare MAE values.
Visible, our proposed method using DT and using K-NN both shows comparatively smallest MAE values.

Table 4.5: MAE results of various Algorithms

Method	MAE value
Proposed method using DT	0.5825
Proposed method using K-NN	0.6425
SSODM	0.7378
Average	0.7978
MODE	0.8014

4.6 Conclusion

This chapter consists of all about result analysis of our work. Here we summarize the result as, for 19 cluster system gives us comparable better performance than 13, 23, 71 cluster. Using 19 cluster we got the MAE result 0.58 from DT approach and 0.64 from K-NN approach, RMSE 0.73 from DT approach and 0.75 from K-NN approach.

Chapter 5

Conclusion

5.1 Conclusion

In this paper, we present a method to the new user cold start problem for RSs applying Collaborative Filtering (CF). The proposed system adopts a three-phase approach in order to provide predictions for new user. We followed a mechanism that takes into consideration their demographic data and based on similarity techniques finds the user's 'neighbors'. We defined as the 'neighbors' are which have the similar characteristics with new user. The idea is that people with similar background and characteristics having more possibilities to have similar preferences. Therefore each new user is classified into a group according a rating prediction mechanism is responsible to result ratings for them. The final prediction of rating calculated by taking average rating for a particular movie within the group in which a few of current user are rated that movie. Our experimental shows the performance of proposed DT and KNN technique. We choose the dataset provided by Grouplens research team. The proposed DT performs better than KNN. When a large amount of current user present in a system the RMSE value is lower than the KNN comparatively.

5.2 Limitations of proposed methodology

- Though there were more than four elbow curve we have tested only for four.
- Comparatively DT is faster than K-NN.
- We predict movie rating with accounting number of user who rated that movie are 5.

5.3 Future Works

In future, we will look forward to solve cold-start recommendation system for large scaled dataset using same methodologies and using different approaches then compare them to find an optimal approach in cold-start recommendation system.

References

[1] S. Kunwar, "Code project." https://www.codeproject.com/Articles/439890/Text-Documents-Clustering-using-K-Means-Algorithm, 2013.

[2] R. Saxena, "Data science, machine learning." http://dataaspirant.com/2017/01/30/how-decision-tree-algorithm-works/, 2017.

[3] S. Günter and H. Bunke, "Validation indices for graph clustering," *Pattern Recognition Letters*, vol. 24, no. 8, pp. 1107–1113, 2003.

[4] U. Shardanand and P. Maes, "Social information filtering: algorithms for automating "word of mouth"," in *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 210–217, ACM Press/Addison-Wesley Publishing Co., 1995.

[5] J. L. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl, "An algorithmic framework for performing collaborative filtering," in *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 230–237, ACM, 1999.

[6] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in *Proceedings of the 10th international conference on World Wide Web*, pp. 285–295, ACM, 2001.

[7] J. S. Breese, D. Heckerman, and C. Kadie, "Empirical analysis of predictive algorithms for collaborative filtering," in *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*, pp. 43–52, Morgan Kaufmann Publishers Inc., 1998.

[8] D. Billsus and M. J. Pazzani, "Learning collaborative information filters.," in *Icml*, vol. 98, pp. 46–54, 1998.

[9] G.-R. Xue, C. Lin, Q. Yang, W. Xi, H.-J. Zeng, Y. Yu, and Z. Chen, "Scalable collaborative filtering using cluster-based smoothing," in *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 114–121, ACM, 2005.

[10] T. Hofmann and J. Puzicha, "Latent class models for collaborative filtering," in *IJCAI*, vol. 99, 1999.

[11] D. M. Pennock, E. Horvitz, S. Lawrence, and C. L. Giles, "Collaborative filtering by personality diagnosis: A hybrid memory-and model-based approach," in *Proceedings of the Sixteenth conference on Uncertainty in artificial intelligence*, pp. 473–480, Morgan Kaufmann Publishers Inc., 2000.

[12] D. DeCoste, "Collaborative prediction using ensembles of maximum margin matrix factorizations," in *Proceedings of the 23rd international conference on Machine learning*, pp. 249–256, ACM, 2006.

[13] R. Bell, Y. Koren, and C. Volinsky, "Modeling relationships at multiple scales to improve accuracy of large recommender systems," in *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 95–104, ACM, 2007.

[14] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Application of dimensionality reduction in recommender system-a case study," tech. rep., Minnesota Univ Minneapolis Dept of Computer Science, 2000.

[15] K. Goldberg, T. Roeder, D. Gupta, and C. Perkins, "Eigentaste: A constant time collaborative filtering algorithm," *Information Retrieval*, vol. 4, no. 2, pp. 133–151, 2001.

[16] M. Balabanović and Y. Shoham, "Fab: content-based, collaborative recommendation," *Communications of the ACM*, vol. 40, no. 3, pp. 66–72, 1997.

[17] C. Basu, H. Hirsh, W. Cohen, *et al.*, "Recommendation as classification: Using social and content-based information in recommendation," in *Aaai/iaai*, pp. 714–720, 1998.

[18] T. Miranda, M. Claypool, A. Gokhale, T. Mir, P. Murnikov, D. Netes, and M. Sartin, "Combining content-based and collaborative filters in an online newspaper," in *In Proceedings of ACM SIGIR Workshop on Recommender Systems*, Citeseer, 1999.

[19] N. Good, J. B. Schafer, J. A. Konstan, A. Borchers, B. Sarwar, J. Herlocker, J. Riedl, *et al.*, "Combining collaborative filtering with personal agents for better recommendations," in *AAAI/IAAI*, pp. 439–446, 1999.

[20] S.-T. Park, D. Pennock, O. Madani, N. Good, and D. DeCoste, "Naïve filterbots for robust cold-start recommendations," in *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 699–705, ACM, 2006.

[21] P. Melville, R. J. Mooney, and R. Nagarajan, "Content-boosted collaborative filtering for improved recommendations," in *Aaai/iaai*, pp. 187–192, 2002.

[22] A. I. Schein, A. Popescul, L. H. Ungar, and D. M. Pennock, "Methods and metrics for cold-start recommendations," in *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 253–260, ACM, 2002.

[23] J. Basilico and T. Hofmann, "A joint framework for collaborative and content filtering," in *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 550–551, ACM, 2004.

[24] D. Agarwal and S. Merugu, "Predictive discrete latent factor models for large scale dyadic data," in *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 26–35, ACM, 2007.

[25] W. Chu and S.-T. Park, "Personalized recommendation on dynamic content using predictive bilinear models," in *Proceedings of the 18th international conference on World wide web*, pp. 691–700, ACM, 2009.

[26] I. Fernández-Tobías, P. Tomeo, I. Cantador, T. Di Noia, and E. Di Sciascio, "Accuracy and diversity in cross-domain recommendations for cold-start users with positive-only feedback," in *Proceedings of the 10th ACM Conference on Recommender Systems*, pp. 119–122, ACM, 2016.

[27] B. Mehta and W. Nejdl, "Attack resistant collaborative filtering," in *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 75–82, ACM, 2008.

[28] B. Cao, J.-T. Sun, J. Wu, Q. Yang, and Z. Chen, "Learning bidirectional similarity for collaborative filtering," *Machine Learning and Knowledge Discovery in Databases*, pp. 178–194, 2008.

[29] R. Saxena, "Data science, machine learning." `http://dataaspirant.com/2016/12/23/k-nearest-neighbor-classifier-intro/`, 2016.

[30] J. Adler, "Data mining." `http://www.csun.edu/~twang/595DM/`, 2009.

[31] grouplens, "Movielens 100k dataset." `https://grouplens.org/datasets/movielens/100k/`, 1998.

[32] M. Chen, C. Yang, J. Chen, and P. Yi, "A method to solve cold-start problem in recommendation system based on social network sub-community and ontology decision model," in *Proceedings of the 3rd International Conference on Multimedia Technology (ICMT 2013)*, pp. 159–166, 2013.