Movie Recommendation System Using Collaborative Filtering

Sadman Sakib

ID: 2016-3-60-046

Sadia Shahreen

ID: 2017-1-60-071

Rafi Md. Hasib

ID: 2016-3-60-061

Md. Fahim Sabab Siyam

ID: 2017-1-60-142

A thesis submitted in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science and Engineering

Department of Computer Science and Engineering

East West University

Dhaka- 1212, Bangladesh

January 2021

We, hereby, declare that the work presented in this thesis is the outcome of the investigation performed by me under the supervision Surajit Das Barman, Senior Lecturer, Department of Computer Science and Engineering, East West University. I also declare that "**Movie Recommendation System Using Collaborative Filtering**" is being submitted elsewhere for the award of any degree or diploma.

Countersigned

. . . .

(Surajit Das Barman)

Senior Lecturer, Dept. of CSE

East west University

Supervisor

Signatures of Author

. . . .

Sadman Sakib (2016-3-60-046)

. . . .

Sadia Shahreen (2017-1-60-071)

. . . .

Rafi Md. Hasib (2016-3-60-061)

. . . .

Md. Fahim Sabab Siyam (2017-1-60-046)

. . . .

ABSTRACT:

System recommend is an out of recent apps that help to recommend items or products based on users need because, user need change from person to person, we can't generalize recommender system [1]. Movie introduces system also one same problem because individuals have different expectations. Overcoming this situation effective referrers rescue proposal this working to fix this situation a sentiment recommend frame is recommend out of this work. One of the most famous methodologies is collaborating filter and find number first similar users and then find users used yes viewed and gave good reviews out of which yes-no viewed by private users, recommended movies for users have advantage big. This system knows the user's interest and recommends the elements of particular interest to the user [2]. For this purpose, the selection of an appropriate measure of similarity is the key to the success of the recommendation system.

ACKNOWLEDGMENT

For the better understanding, a sample Acknowledgment is given below.

As it is true for everyone, we have also arrived at this point of achieving a goal in our life through various interactions with and help from other people. However, written words are often elusive and harbor diverse interpretations even in one's mother language. Therefore, we would not like to make efforts to find best words to express my thankfulness other than simply listing those people who have contributed to this thesis itself in an essential way. This work was carried out in the Department of Computer Science and Engineering at East West University, Bangladesh.

First of all, we would like to express my deepest gratitude to the almighty for His blessings on us. Next, our special thanks go to our supervisor, "Surajit Das Barman", who gave us this opportunity, initiated us into the field of **"Movie Recommendation System Using Collaborative Filtering"**, and without whom this work would not have been possible. His encouragements, visionaries and thoughtful comments and suggestions, unforgettable support at every stage of our BSc study were simply appreciating and essential. His ability to muddle us enough to finally answer our own question correctly is something valuable what We have learned, and We would try to emulate, if ever We get the opportunity.

I would like to thank "Sadman Sakib", "Rafi Md. Hasib" and "Md. Fahim Sabab Siyam" for their excellent collaboration during performance evaluation studies and overall support.

Last but not the least, we would like to thank our parents for their unending support, encouragement and prayers.

There are numerous other people too who have shown me their constant support and friendship in various ways, directly or indirectly related to our academic life. We will remember them in our heart and hope to find a more appropriate place to acknowledge them in the future.

Sadman Sakib	Rafi Md. Hasib
ID: 2016-3-60-046	ID: 2016-3-60-061
Sadia Shahreen	Md. Fahim Sabab Siyam
ID: 2017-1-60-071	ID: 2017-1-60-142

Chapter 1

1.1 Introduction

The advancement of the internet and e-commerce makes our life a lot of convenient as billions of needed merchandises are searchable online. Meanwhile, we should face the matter of knowledge overload in daily life. beneath the circumstances, it's a lot of tougher for North American nation to dig out relevant object that we actually wish than ever before [3]. several analyzers have done many research on the advice system, creating progress concerning this explicit issue. however, data meagerness has perpetually been a vital reason for the recommendation low accuracy. to create full use of existing information, researchers have projected more and more glorious algorithms, reminiscent of neighborhood-based CF (Collaborative Filtering) and model-based CF.

Essentially, the aforesaid cooperative filtering and diffusion-based strategies are supported similarities [4]. In collaborative filtering, the foremost ordinarily used index is trigonometric function similarity [5]. However, it powerfully tends to suggest fashionable objects, leading to correct however less-diverse recommendations [6]. Neighborhood-based CF algorithms are additional classified into 2 categories: user- primarily based CF [7] and item- based CF [8]. and also, the bedrock of them is interlinked. For instance, user-based CF considers two users to be similar once their neighbors are similar. Obviously, choice of nearest neighbor is significant. Correspondingly, selecting the appropriate similarity are useful for the development of the suggested accuracy and also the pertinency to the recommended rule [9]. The investigator takes into consideration the score data that best reflects the user's preferences. They came up with similarity measures reminiscent of trigonometric function similarity [10], Pearson correlation [11], adjusted cosine [12]. Et al suggests such as Salton similarity [11] and Jaccard similarity, that is considered the quantity of things of self and their neighbors. A vertex similarity index CosRA is proposed, which mixes each benefit of cosine index and resource-allocation index [3]. This fusion enhances the overall performance of personalized recommendation.

1.2 Aim

Our aim is to find a more appropriate algorithm which can give us more accuracy practically and mathematically whenever we try to find movie suggestion from any movie recommendation system.

1.3 Motivation

In general, user behavior is a complex process and is not constant for each user, so it is too difficult to determine the homogeneity between users by means of unique correlation determination. Some methods of determining correlation will not work well in all cases. Different methods work well in different situations. In addition, the performance of traditional correlation or similarity methods is not suitable for the case where the data set is sparse and it is not possible to predict recommendations for users or items that start to cool. Since most real applications deal with sparse data sets, this problem needs to be handled efficiently. So, here a new fast and efficient element-based CF method has been proposed with better prediction accuracy to overcome the problem of existence of cold-start elements. The novelty of the proposed method on the one hand is the direct asymmetric similarity between the elements according to the element type on the basis of the asymmetric characteristic of the trust relationship. Second, the relative similarity relationship between the elements is calculated based on the transitive property of the confidence values of the elements. Again, determining the similarity between items using the item's gender is a novel approach in this area, which has not been discussed before. When a new item enters the system, the item can be of any category. In addition, an article that is barely reviewed must be in at least one category. Since gender information is available in the past, similarity between items can be determined using gender help, which is key to alleviating the problem of cold start entries. However, the category-based item similarity method might consider a new item similar to the highest-ranked item in the system because of its presence in more popular categories. However, since one of the items is new and another may be very popular in the system and contain user rating information, the degree of similarity between them will not be uniform. By considering the respective item's rating data, an asymmetric similarity relationship is introduced into the similarity of items based on gender. In addition, an improvement in the prediction method has been proposed to increase the performance accuracy of the recommendation system [13].

1.4 Research Objectives

There are many ways to explore this recommendation system, where improvements in this industry can be found. The objective of this report is to explore different types of recommendation systems for movies. To be able to find a way to be more efficient. As we all know, there is no limit to being more efficient.

• Reduce sparse issues,

• Get more filtered data,

• Be more efficient,

• Reduce time complexity,

• Solve over-specialization.

1.5 Paper Contribution

The main objective of this paper is to design a new collaborative item-based filtering model by calculating the similarity of the items based on the gender of the items and also improve the prediction accuracy of the recommendation. The contributions of this paper are classified into four categories:

• A unique asymmetric similarity method has been proposed that considers the gender of the items with the confidence between them determining the similarity relationship. direct asymmetry of items.

• Another new method of similarity was determined by determining the correlation of items based on a bridging confidence relationship between them.

• A prediction algorithm is proposed to increase the accuracy of the recommendation.

• Detailed testing is performed to prove our claim that the proposed method is better than the existing methods.

CHAPTER 2

Literature Review

Ramni Harbir Singh et al. showed how to model a movie recommendation system by incorporating content-based filtering into the recommender system [14]. The KNN algorithm is used in conjunction with the cosine similarity principle to produce a model that is more accurate than other distance measurements while having relatively low operational complexity.

M. K. Kharita, A. Kumar, and P. Singh projected the Item Based cooperative Filtering technique in that the moving picture Recommendation system is to give a prediction of the various things in which a user would have an interest supported their preferences. item-based collaborative filtering technique is wont to provide recommendations of items [15]. In Item-based collaborative filtering in movie recommendation cos similarity matrix are generated which contains the expected ratings of the movies. From these predicted ratings prime rated movies are designated and counseled to the actual user. In this Technique, they achieved 79.92 accuracy.

Harper, et. al. [16] referenced the insights regarding the Movie Lens Dataset in their examination paper. This dataset is broadly utilized particularly for film proposal reason. There are various adaptations of dataset accessible like Movie Lens 100K/1M/10M/20M/25M/1B Dataset. The dataset comprises of elements like client id, thing id/film id, rating, timestamp, film title, IMDb URL, delivery date, and so on alongside the film sort data.

Item based CF initially concocted Amazon's thing proposal and a short time later sanctioned by other help-based sites like YouTube [17]. In thing-based CF, the likeness of each pair of things is characterized in light of the evaluations that are given by the clients and afterward recommends another arrangement of things to an objective client that isn't appraised at this point by the objective client however associated with the objective client's appraised things. For Cosine-based Closeness, Pearson relationship-based Likeness, Changed Cosine Comparability, and so forth, similitude measures are utilized to distinguish the level of connections between things. Numerous scientists have proposed various ways to deal with develop a superior adaptation of thing-based CF. Li et al. have proposed a security saving thing-based CF by assessing clients' protection of evaluations from others. They prescribe an unsynchronized convention called Unseen’s to accomplish secure multi-party calculation. From that point onward, they have changed two famous comparability calculation techniques without influencing RS' execution and proposed Private Cosine and private individual relationship strategy with the assurance of clients' protection. Dakhel et al. have characterized another technique to register the level of relationship between things by adjusting the conventional cosine comparability strategy named as Thing Deviated Connection (IAC). Then, at that point, the uneven relationship is utilized as extra data in network factorization to fuse with thing-based CF. In any case, there actually exist a few issues and the greater part of them are sparsity, adaptability, and cold start.

Various diverse model-based methodologies have been fostered that utilization thing to-thing similitudes just as affiliation rules. Shardanand and Maes [18] fostered a thing-based expectation calculation inside the setting of the Ringo music suggestion framework, alluded to as craftsman, that decides if a client will like a specific craftsman by registering its similitude to the specialists that the client has preferred/hated before. This comparability was figured utilizing the Pearson connection work. Sarwar et al. [8] further read up this worldview for processing forecasts and they assessed different techniques for registering the closeness just as ways to deal with limit the arrangement of thing to-thing similitudes that should be thought of. The creators revealed significant upgrades in execution over the client-based calculation. Mobasher et al. [18] introduced a calculation for prescribing extra website pages to be visited by a client in view of affiliation rules. In this methodology, the authentic data about clients and their web-access designs were mined utilizing a regular itemset revelation calculation and were utilized to create a bunch of high certainty affiliation rules. The suggestions were figured as the association of the resulting of the guidelines that were upheld by the pages visited by the client. Lin et al. [19] utilized a comparable methodology however they fostered a calculation that is ensured to observe affiliation rules for every one of the things in the information base. At long last, inside the setting of utilizing affiliation rules to determine top-N proposals, Demiriz [20] concentrated on the issue of how to weight the various standards that are upheld by the dynamic client. He introduced a strategy that registers the closeness between a standard and the dynamic client's bin as the result of the certainty of the standard and the Euclidean distance between things in the forerunner of the affiliation rule and the things in the client's bin. He contrasted this methodology both and the thing-based plan depicted in Section 4 (in view of our starter work introduced in Karypis [4] and the reliance network-based calculation [21]. His tests showed that the proposed affiliation rule-based plan is better than reliance networks yet substandard compared to the thing-based plans.

CHAPTER 3

3.1 Recommendation System

The recommendation system is more common and increases the cost of production for many service providers. Today, the world is an overcrowded, so recommendations are needed to recommend any product or service. However, the recommender system reduces transaction costs and improves the quality and decision-making process for users [22], [23]. It is applied in various related fields such as information retrieval or human-computer interaction (HCI). It collects huge amount of information about user preferences for some items like online shopping products, movies, taxis, TV, travel, restaurants, etc. It stores information in different ways positive or negative. It attracted user reviews of watched movies, browsed locations and purchased products. When compares the need to buy products, service providers (travel and restaurants), there is a big problem with the design of the movie recommendation system, as other recommendation systems require the service Fast calculation and processing of from service providers and product distributors. To recommend movies, first collects user ratings and then recommends the highest rated list to target users. In addition, users can check other users' reviews before watching movies. Various recommendation schemes were presented, including collaborative filtering, content-based recommendation system, and combined recommendation system. However, some problems arise with user-posted reviews.

3.1.1 Drawback of Recommendation System

Since the recommendation system is used for many e-commerce sites, it is very easy to use the user's personal data. So, there will always be insecurity issues. Data can easily be distributed to multiple parties. For which third parties can easily predict user preferences or choices. Based on this data, they will target customers and enrich their business [24].

3.2 Item Based Filtering

Collaborative filtering between repeats or repeats or items is a form of recommender system collaborative filtering based on the similarities between items calculated using people's ratings for those items. Item-item Collaborative Filtering was invented and used by Amazon.com in 1998.

First presented at the 2001 Science Council. The article model solves these problems on systems with more users than articles. Item The item model uses a valuation distribution by item, not by user. If you have more users than articles, each article tends to get a higher rating per user, so the average article rating of doesn't change immediately. This makes the distribution of points in the model more stable and eliminates the need to recreate the model frequently. When a user evaluates using an article, similar articles within that article are selected from existing system templates and added to the user's recommendations.

Item’s rating is predicted based on how similar items have been rated by that user. The ratings are predicted using the user’s own ratings on neighboring (closely related) items. Neighborhoods are defined by similarities among items. (Columns of the ratings matrix) Adjusted Cosine similarity provides superior results.

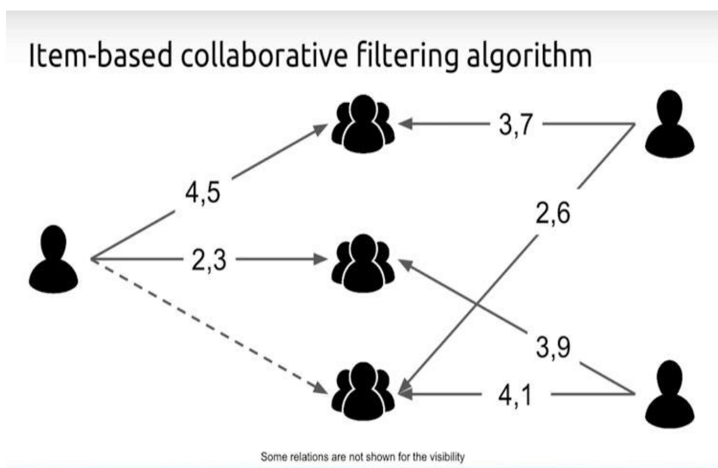

Fig 1. Work process of Item-Based Collaborative system

Pearson Method

Pearson's correlation coefficient is defined as the product of the covariances of two variables divided by the product of their standard deviations (divided by the product of their standard deviations). The definition includes a "product moment," which is the mean (the first moment about the origin) of the product of the mean-adjusted random variables; this is reflected in the name, which includes the modifier product-moment as a part of the definition.

It is necessary to first identify the covariance between the two variables in issue in order to compute their Pearson product-moment correlation. Following that, the standard deviation of each variable must be calculated. The correlation coefficient is computed by dividing the covariance by the product of the standard deviations of the two variables in the data set.

Formula,

$$r = \frac{\sum_{i=1}^{n} \quad (x_i - \underline{x})\,(y_i - \underline{y})}{\sqrt{\sum_{i=1}^{n} \quad (x_i - \underline{x})^2}\sqrt{\sum_{i=1}^{n} \quad (y_i - \underline{y})^2}}$$

r = correlation coefficient

x_i = values of the x-variable in a sample

$\underline{x}$ = mean of the values of the x-variable

y_i = values of the y-variable in a sample

$\underline{y}$ = mean of the values of the y-variable

Cosine Similarity

In the case of two items, the cosine similarity between them is measu$Type\ equation\ here.$red by the angle of cosine between them. It compares files on a scale that has been standardized. This may be accomplished by comparing the placements of the dot product across two different IDs. The attitude between v1 and v2 is shown in the figure above. The greater the similarity of the vectors' attitudes, the greater the dissimilarity between them. If the attitudes between two vectors were minor, he felt that the vectors may be the same; but, if the attitudes between two vectors were great, he believed that the vectors were extremely distinct since they were all different. By using this similarity matric a recommendation system can recommend the next most similar item to the user. This the cosine similarity formula. Where in this, A and B are

$$similarity\ (A\ ,B) = \frac{A\ .B}{\|A\|\ \times\ \|B\|} = \frac{\sum_{i=1}^{n}\quad A_i\ \times\ B_i}{\sqrt{\sum_{i=1}^{n}\quad {A_i}^2}\ \times\ \sqrt{\sum_{i=1}^{n}\quad {B_I}^2}}$$

The lists of ratings of the similar movies browsed by the two users. Which computes similarity as the normalized dot product of input samples A and B.

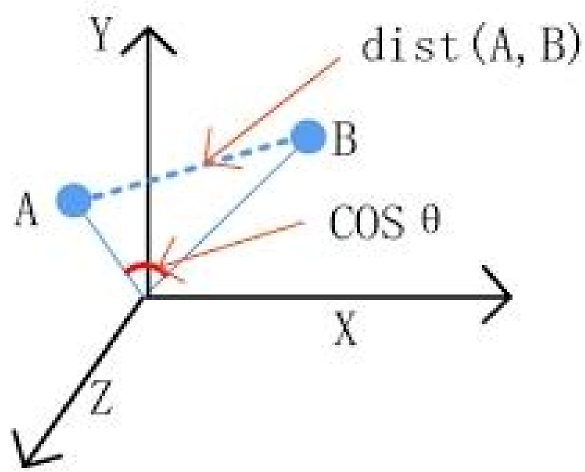

Fig 2. Cosine Similarity

From the diagram the angle theta between two movies B and A. The theta ranges from 0-1 and it shows the relation of two movies that their similarity is good or bad. So, if the angle between two vectors is small or theta value is near 1, they are same each other and if the angle between the two vectors is large or theta value is near 0 then the movies are least similar. Only those movies which movies value will be near 0 only those will be recommended according to the user.

3.2.1 Advantage of Item based filtering

Since the anticipated ratings are based on the ratings provided by the user herself, the predicted ratings tend to be much more consistent with the other ratings provided by this particular user.

It is typical practice to employ item-based similarity when providing high-volume services. • The rationale for this is because the neighborhood of an item changes considerably more slowly than the neighborhood of a user.

• It is much simpler to communicate the advice to the consumers when using this method.

• Item-based recommenders outperform their user-based counterparts by a significant margin.

• The major benefit of the item-based technique is that it predicts with more accuracy than other methods.

• Item-based approaches in recommender systems are also more resistant to shilling assaults than other methods.

3.2.2 Limitations of Item based filtering

There are several limits to item-based filtering that may be seen as well.

• Users' past experiences with objects-based approaches may lead to the recommendation of items that are evident or that are not original to them.

• LSA has a negative overall influence on the item-based suggestions, according to the findings.

3.3 Collaborative Filtering System

Collaborative filtering is a technique that can filter out items that users might like based on the reactions of similar users. This feature works by searching a large group of people and finding a smaller group of users with similar interests to a particular user. It looks at the items they like and combines them to create a ranked list of recommendations. There are many ways to determine which users are alike and combine their selections to create a list of recommendations. Collaborative filtering works across the interactions that customers have with gadgets. These interactions can assist locate styles that the information approximately the gadgets or customers itself can`t. As example,

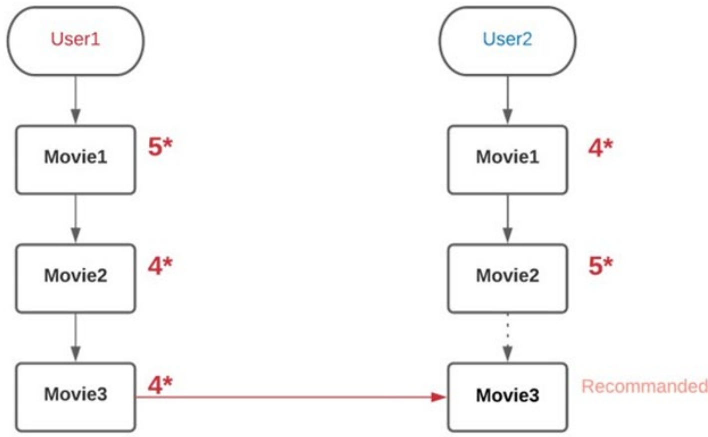

Fig 3. Collaborative Filtering System

We have two users. User 1 has watched Movie1 and rated 5 out of 5. After that the user watched movie2 and rated 4 out of 5. On the other-hand User 2 has also watched both Movie 1 and Movie2 rated 4 and 5. Now, User1 Has watched Movie3 and rated 4 out of 5. That means the user felt interested and liked the movie so much. As the User 2 has also got same type of interest. So, the probability of liking this movie for User 2 is high. So, this movie 3 will be recommended towards User 2 as both users have almost same mentality and user experience.

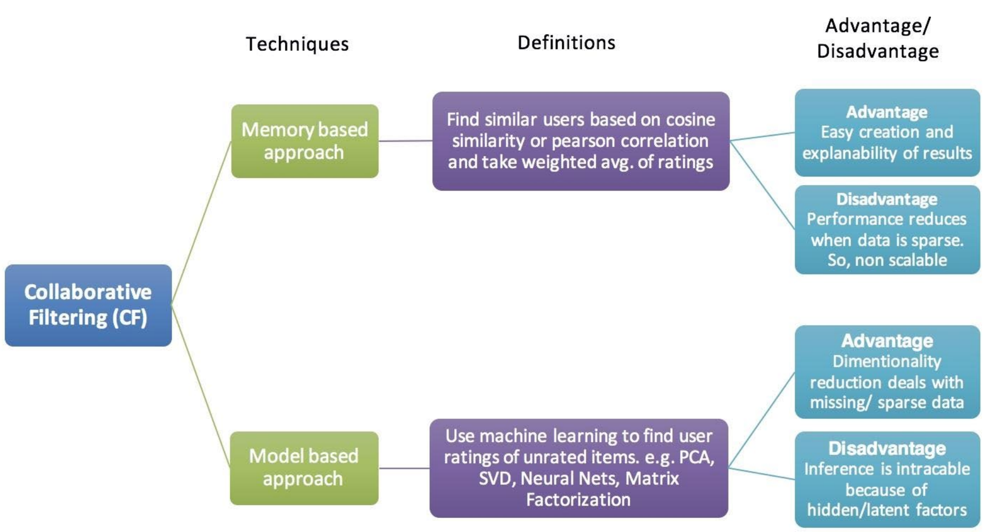

Fig 4. Classification of Collaborative Filtering (CF)

3.3.1 Memory Based

Memory-based collaborative filtering technology can be divided into two main parts: user item filtering and item filtering. The User Articles filter gets a specific user, searches for users who are similar to that user based on rating similarity, and recommends articles that these similar users like. In contrast, item filtering retrieves an item, searches for users who already like the item, and searches for other items that those users or similar users also like. He receives the item and creates other items based on the referral. The main difference between the memory-based approach and the model-based approach (waiting, described in the next section) is that you do not use gradient descent (or other optimization algorithm) to learn the parameters. The user or the next entry is calculated using only the cosine similarity or Pearson's correlation coefficient, based solely on arithmetic operations. A common measure of distance is cosine similarity. You can think of a metric as a geometry by thinking of a row (column) of a particular user (item) in the score matrix as a vector [25]. In user-based collaborative filtering, similarity between two users is measured as a sine and cosine of the angle between the vectors of the two users. For users u and u, the cosine similarities are:

$$sim\ (u\ ,u') = cos\ cos\ (\theta) = \frac{r_u \cdot r_{u'}}{\|r_u\|\ \|r_{u'}\|}$$

$$= \sum_i \frac{r_{ui}\ r_{u'i}}{\sqrt{\sum_i \quad r_{ui}^2}\sqrt{\sum_i \quad r_{u'i}^2}}$$

3.3.2 Model Based

Following this approach, CF models are developed using machine learning algorithms to predict user ratings for unrated items. To my knowledge, the algorithms of this approach can be divided into 3 sub-categories. Model-driven collaborative filtering generates recommendations by analyzing user ratings and building models from them. It can use both explicit and hidden data, such as user ratings, and can also observe habits such as music played, apps downloaded, websites visited or books read. . .. Model-driven algorithms start by building a model of a user's behavior and thus making predictions about their ratings. The data from the scoring matrix is used to estimate the model's parameters. Compared to other approaches, such as memory-based methods, these methods have the advantage of being able to display more items to more users. Even when working with not too large molds, we would say they have a wide coverage.

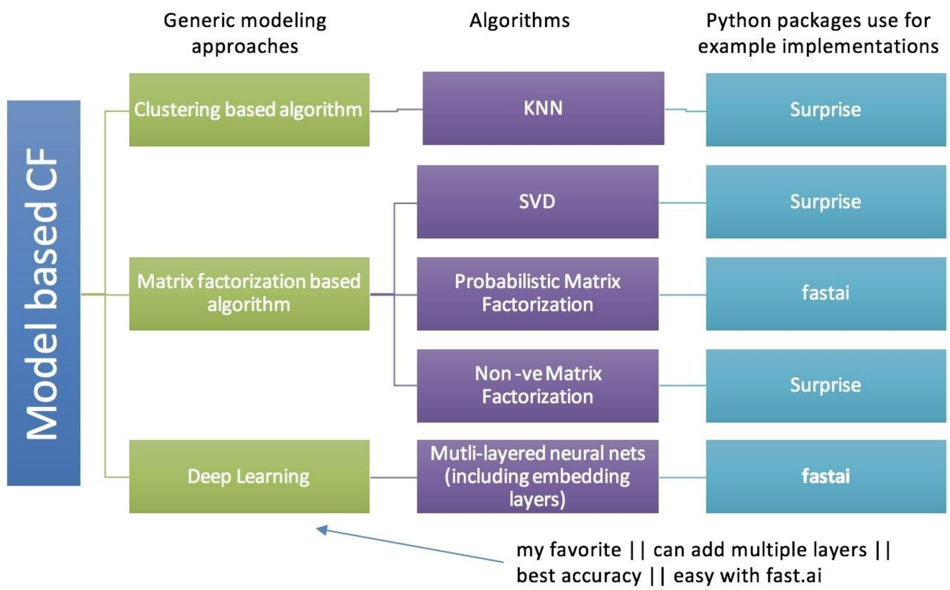

Fig 5. Classification of Model Based CF

3.3.3 Random Forest

With the Random Forest algorithm, you can learn about new things without having to worry about what you're doing. When it comes to classification and regression analysis, the random forest approach comes in handy. Given a data sample, it constructs a large number of decision trees, then integrates their outputs to anticipate the result and picks the best one among them. Using a Random Forest technique in a group setting is the most effective way to improve results. The random forest method's operational mechanism, as well as the forest that it is capable of generating, is an ensemble of decision trees, and it is widely used to train the bagging algorithm, which is a decision tree-based algorithm that learns from previous decisions. In order to improve the overall result of the experiment, the approach of bagging learning is used. Bagging learning is a technique that brings together several learning models. Most importantly, this technique may be used to both classification and regression scenarios, which is a considerable benefit. Preventing overfitting may be accomplished by taking the results of several different decision trees and averaging or combining them. The benefit of using a decision tree is that it can cope with a larger number of data points than a single decision tree. A multi-decision tree exhibits less volatility than a single decision tree when compared to a single decision tree. In addition to being very accurate, it provides a great degree of adaptability. When data is presented without being scaled, a high level of accuracy is maintained in the findings. Even if a major portion of the data is missing, it is still feasible to attain high levels of precision and accuracy. It has a number of drawbacks, the most significant of which is that it is very difficult to understand. When compared to decision trees, the design of decision trees is far more difficult and time-consuming. Greater computer resources are urgently required to meet the demands of the current workload. With a high number of decision trees, the process gets more complex to manage.

3.3.4 Linear Regression

Linear regression is one of the simplest and widely used Machine Learning techniques available. An example of this would be the use of regression analysis in predictive analysis. For continuous/real or quantitative variables such as sales, salary, age, product price, and so on, linear regression may be used to create predictions.

The linear regression technique demonstrates a linear connection between a dependent (y) variable and one or more independent (y) variables, which is why it is referred to as linear regression in this context. Because linear regression reveals a linear connection, it may be used to determine how the value of the dependent variable changes as a function of the value of the independent variable, or vice versa.

The linear regression model generates a slanted straight line that represents the connection between the variables in the data set. Take a look at the picture below:

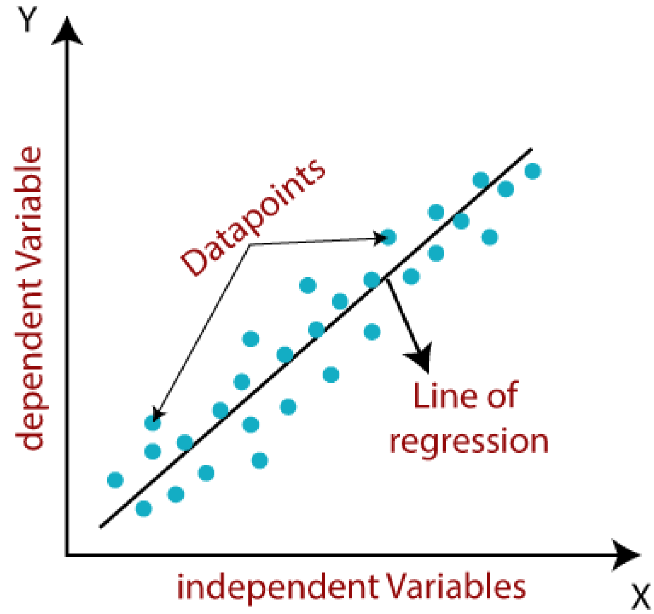

Fig 6. Linear Correlation between X and Y

Mathematically, we can represent a linear regression as:

y= $a_0 + a_1x + \varepsilon$

Here,

Y= Dependent Variable (Target Variable)

x= Independent Variable (predictor Variable)

a_0= intercept of the line (Gives an additional degree of freedom)

a_1= Linear regression coefficient (scale factor to each input value).

ε = random error

The values for x and y variables are training datasets for Linear Regression model representation.

3.3.5 Support Vector Machine (SVM)

Support Vector Machine, often known as SVM, is one of the most widely used supervised learning algorithmic programs, and it is used for both classification and regression problems. However, it is mostly utilized in Machine Learning to solve issues with classification. With the SVM method, the purpose is to create the simplest line or call boundary that can be used to divide n-dimensional space into categories, so that we can easily insert a new datum into its appropriate class in the future. A hyperplane is a boundary that represents the optimal choice boundary.

SVM chooses the intense points/vectors that facilitate in making the hyperplane. These extreme cases are referred to as support vectors, and therefore algorithmic program is termed as Support Vector Machine. Think about the below diagram within which there are 2 completely different classes that are classified employing a call boundary or hyperplane:

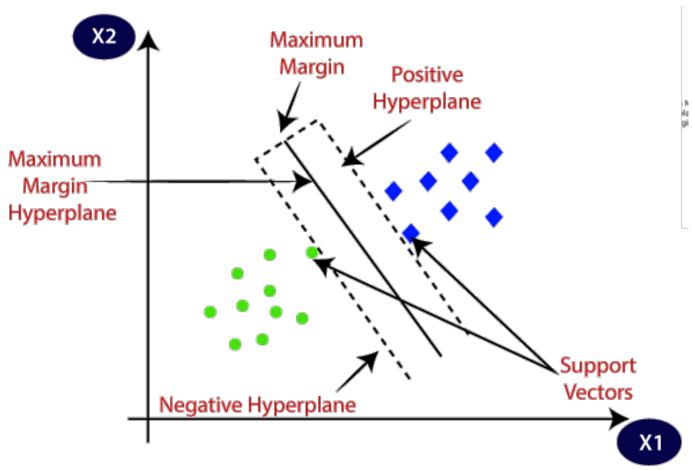

Fig 7. Explanation of SVM

3.3.6 K-Nearest Neighbor (K-NN)

Nearest Neighbor (also known as K Nearest Neighbor) is one of the most straightforward machine learning algorithms, based on supervised learning approaches. The KNN algorithm makes the assumption that the new case/new data is comparable to the existing cases and assigns the new case to the category that most closely matches the available categories in the KNN algorithm's database. This algorithm saves all of the data that is available and classifies a new data point based on how similar it is to the existing data. This implies that when new information is received, it may be quickly identified as beneficial. The KNN algorithm and how to utilize it. It may be used for both classification and regression issues, however it is more often employed for classification problems. It is important to note that KNN is a non-parametric method, which means it does not make any assumptions about the underlying data. It is sometimes referred to as a lazy learning algorithm since it does not learn from an immediate training set; instead, it stores the data set and performs an action on the dataset at classification time. During the learning phase, this algorithm just retains the data set, and when it gets new data, it classifies the new data as a type that is highly similar to the data set it previously stored [26]. We utilize KNN because, if there are two categories, i.e., Category A and Category B, and we have a new data point x1, we can use KNN to determine which of these categories this data point belongs to. To solve this type of problem, we need a K-NN algorithm. With the help of K-NN, we can easily identify the category or class of a particular dataset. Consider the below diagram:

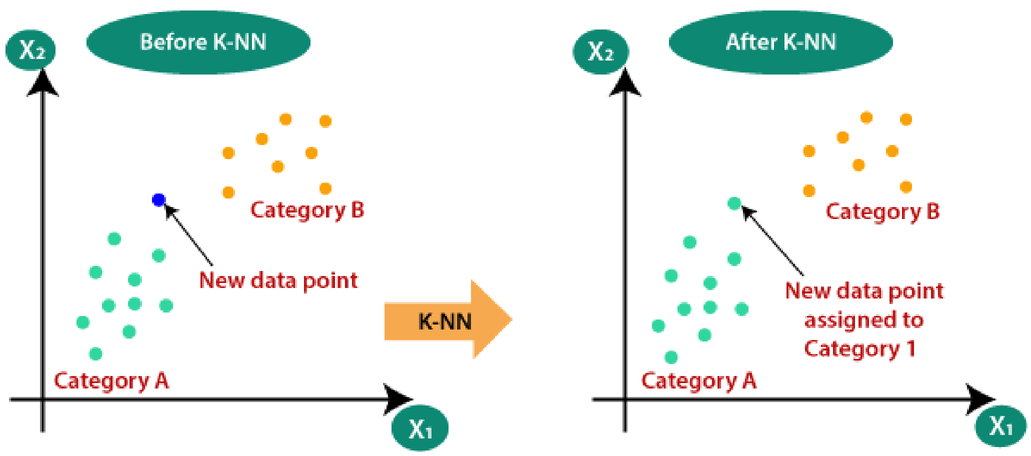

Fig 8. Example of K-NN

3.3.7 Decision Tree

Decision Tree is a supervised learning approach that may be used to solve both classification and regression problems, however it is most often employed to solve classification issues. Internal nodes contain dataset attributes, branches represent decision rules, and each leaf node provides the conclusion in this tree-structured classifier.

Two nodes in a decision tree are the Decision Node and the Leaf Node, which are interchangeable. Choice nodes are used to make any choice and have numerous branches, while Leaf nodes are the consequence of those decisions and do not have any more branches.

The judgments or tests are made based on the characteristics of the provided dataset. It's a graphical depiction for obtaining all feasible answers to a problem/decision depending on certain parameters.

It's termed a decision tree because, like a tree, it begins with the root node and grows into a tree-like structure with additional branches [27].

In order to construct a tree, we use the CART method, which stands for Classification and Regression Tree algorithm. A decision tree simply asks a question and divides the tree into subtrees depending on the response (Yes/No). A decision tree's general structure is seen in the image below:

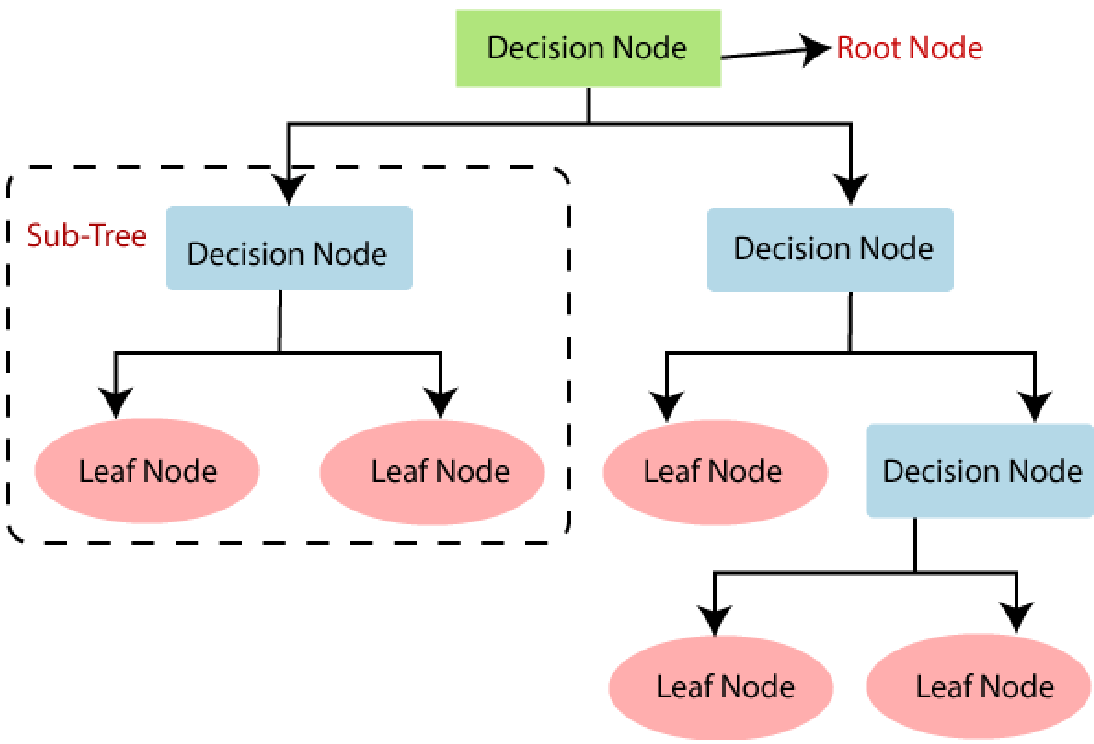

Fig 9. Classification of Decision Tree

3.3.8 Bagged Regression Tree

The decision tree was introduced by Breiman in 1984, and it is widely used in classification and regression analysis. When used for classification, each leaf node in the decision tree represents a category, and when used for regression, each leaf node represents a continuous projected value.

Tree of Regression

It is most often used for continuous variables in Regression Tree, but merely to indicate the estimated amount. If the algorithm decides to make a choice, the value or parent is divided into levels depending on the child's level.

Bagging

Breiman was the first to develop the bagging approach, which operates on the premise of creating and merging several individual learners to perform the final prediction assignment. For reducing the variance of regression trees, this regression approach is often utilized. It also tends to tackle the issue of overfitting in a single tree.

3.3.9 Gradient Boosting

One of the most effective machine learning algorithms is the gradient boosting method, which is also known as GBA. As we all know, mistakes in machine learning algorithms may be roughly categorized into two categories: bias errors and variance errors. Bias errors are the most common types of errors. Since gradient boosting is one of the boosting methods, it is utilized to reduce bias error in the model.

In contrast to the Ad boosting technique, the base estimator in the gradient boosting approach is not explicitly stated by the authors. For the Gradient Boost approach, the base estimator, also known as the Decision Stump, is always the same. We can tweak the n estimator of the gradient boosting method in the same way that we can tune the AdaBoost algorithm. This method will use 100 as the default value of an estimator if we do not provide any other value for an estimator in the input parameter list.

It is possible to utilize the gradient boosting technique to forecast not only continuous target variables (in the form of a Regressor), but also categorical target variables (as a Classifier). For regression, the cost function is the Mean Square Error (MSE), and for classification, the cost function is the Log Loss (Loss Logistic Regression).

CHAPTER 4

4.1 Introduction

Methodology is the study of ideas, procedures, and principles related to a certain topic or issue. To establish a new field of study, a wide variety of research papers are required. All of the relevant data is gathered and used to support the recommendation system that has been proposed. However, each study recommends a different career path, but In the end, it is a collection of recommendations for future scientific analysis in a certain area [14].

4.1.1 Feasibility Study

Four researchers and one supervisor work on this eight-month-long thesis. The analysis will necessitate the assistance of a technical professional [28]. For example, software, dataset, also some necessitates the use of an assessment process. The dataset's legal viability is factored into the equation when gathering data for this thesis. In order to complete the thesis, the university or supervisor did not provide any financial assistance, but they help us in every positive way.

4.1.2 Requirement Analysis

The following are all conditions that must be met by this design:

a) Scientific computations necessitated the use of open-source software libraries.
b) It was an extremely powerful machine for performing python.
c) To bring the machine learning model into work, you'll need these free open-source libraries.

4.1.3 Methodology

In this section, we'll go through how we went about how to do our analysis [29].

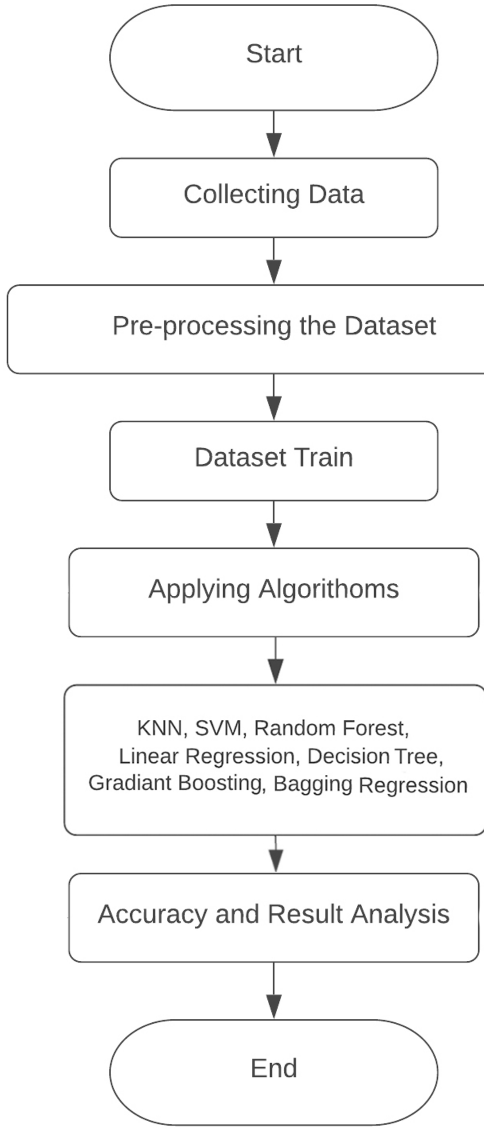

Fig 10. Workflow Diagram

4.2 Collecting Data

We worked with movieLens [30] datasets for the recommendation system. All the rows have been collected and we had Movie title, Movie id, user id, user age, movie ratings, genres, timestamp etc. We've narrowed it down to the rows that are relevant to our current project. Afterward, we compiled the data into a CSV file, which we used to train our machine.

4.3 Data Pre-processing

The process of transforming unstructured data into something usable is known as data processing. Preprocessing of the chosen dataset is required. Because there may be omissions or inaccuracies in the raw data. A few fields are being dropped from the dataset at this time. For our experiment, we only use the fields that are relevant to our research. For the sake of this experiment, we only examined fields that were shown to be useful; these fields are essential to our system's operation. Starting with fields that had no impact on our system, we chose to lower the amount of these fields because it would increase our system's performance while also reducing our time and space complexity. To verify our results, we reduced the number of fields and double-checked them. Our results have not altered. This isn't to say that the vast dataset is without flaw. Finally, we minimized the number of fields that have no effect on our system's output. As a result, we saw an increase in the output of our system's performance.

4.3.1 Dataset Training

For the project, after importing the csv file, we've trained a machine to use our dataset. There is some use of third-party applications and libraries. The Python programming language was utilized in its implementation. We began by putting our machine together. After that, we employ anaconda for implementation. There are some certain libraries here that we'll need in the implementation. As a first step, we need to import our dataset CSV file.

```
In [48]: ratings.head()
```

Out[48]:

	movieId	title	userId	rating
0	1	Toy Story (1995)	1	4.0
1	1	Toy Story (1995)	5	4.0
2	1	Toy Story (1995)	7	4.5
3	1	Toy Story (1995)	15	2.5
4	1	Toy Story (1995)	17	4.5

Fig 11. Simple Output of our dataset

For both user and movie identifiers, we employ plot. So that the user-rated movie storyline graphics are easy to follow [13].

Using the User ID as the X axis, the number of votes cast by each user is shown on the Y-axis.

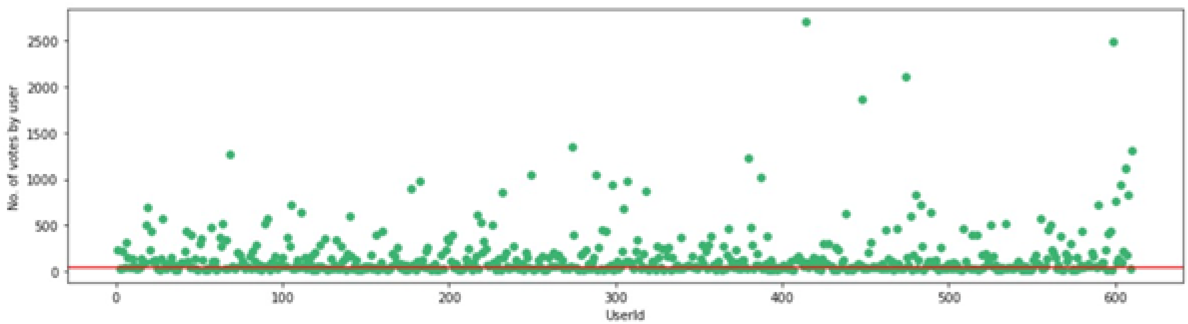

Fig 12. UserId Plot

In contrast, the X and Y axes are based on the movie ID and the No. of votes cast, respectively.

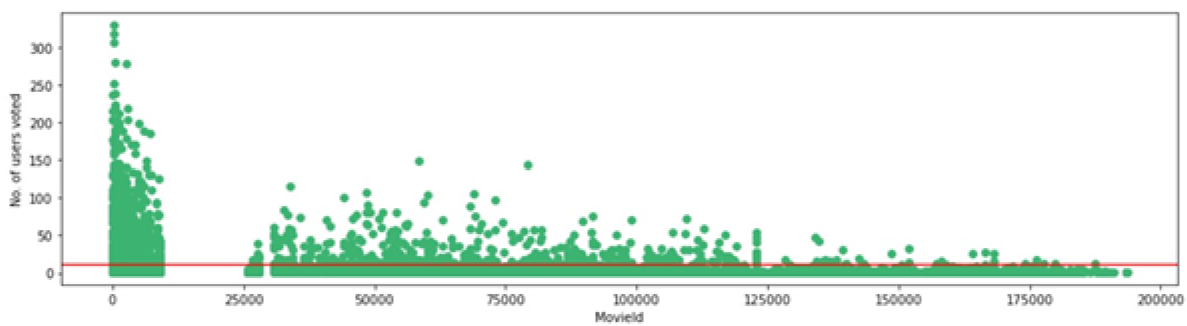

Fig 13. MovieId Plot

Data is being translated from Jason form to reading list form using data frames that we've created. For data frames that had a large number of missing values, we used the droop command.

We utilized the binary encoding function to retrieve binary values from a list because "Machine Learning" doesn't work at the string level and only requires binary data.

4.4 Conclusion

A lot of information regarding our datasets and models has been covered in this chapter. For this study, we focused on only one dataset for the purpose of the study.

Chapter 5

5.1 Introduction

This chapter focuses on the experimental study of the models on the train and test datasets. The models' performance is also briefly discussed in this section.

5.2 Result Analysis

Pearson correlation and cosine similarity were utilized in the Python with jupyter notebook for this project. We compute similarity and correlation between two movies in the same way as we do with cooperating filters and item-item similarity [31]. As part of our dataset, we have the following information: movieId, genre, age of the user etc. To begin, we used pearson correlation to determine the level of association between each film [32] . Cosine similarity is another method we used to calculate the distance between all the movies included in our dataset. Dissimilarity decreases as distance decreases.

title title	'burbs, The (1989)	(500) Days of Summer (2009)	10 Cloverfield Lane (2016)	10 Things I Hate About You (1999)	10,000 BC (2008)	101 Dalmatians (1996)	101 Dalmatians (One Hundred and One Dalmatians) (1961)	12 Angry Men (1957)	12 Years a Slave (2013)	127 Hours (2010)	...	Zack and Miri Make a Porno (2008)	Zero Dark Thirty (2012)	Zero Effect (1998)	Z
'burbs, The (1989)	1.000000	0.063117	-0.023768	0.143482	0.011998	0.087931	0.224052	0.034223	0.009277	0.008331	...	0.017477	0.032470	0.134701	0.15
(500) Days of Summer (2009)	0.063117	1.000000	0.142471	0.273989	0.193960	0.148903	0.142141	0.159756	0.135486	0.200135	...	0.374515	0.178655	0.068407	0.41
10 Cloverfield Lane (2016)	-0.023768	0.142471	1.000000	-0.005799	0.112396	0.006139	-0.016835	0.031704	-0.024275	0.272943	...	0.242663	0.099059	-0.023477	0.27
10 Things I Hate About You (1999)	0.143482	0.273989	-0.005799	1.000000	0.244670	0.223481	0.211473	0.011784	0.091964	0.043383	...	0.243118	0.104858	0.132460	0.09
10,000 BC (2008)	0.011998	0.193960	0.112396	0.244670	1.000000	0.234459	0.119132	0.059187	-0.025882	0.089328	...	0.260261	0.087592	0.094913	0.18

Fig 14. Correlation between movies with Pearson correlation

According to Pearson and cosine similarity, the projected rating and the actual rating are very close. We can recommend the ratings and the 10 most similar movies based on the movies I'd like or the movie names. Example: For “Iron Man (2008)”. In this case, the projected rating is 4.0 out of 5, but the actual rating is 3.9. We might also recommend some of the best movies that are similar to ours “Iron Man (2008)”.

```
Iron Man (2008)                              1.000000
Avengers, The (2012)                         0.611249
WALL·E (2008)                                0.607031
Iron Man 2 (2010)                            0.605294
Dark Knight, The (2008)                      0.599719
                                                ...
Specialist, The (1994)                      -0.089981
Madness of King George, The (1994)          -0.092041
Postman, The (Postino, Il) (1994)           -0.102792
Disclosure (1994)                           -0.114447
Piano, The (1993)                           -0.116370
Length: 2269, dtype: float64
```

Fig 15. All Correlation with “Iron Man (2008)” movie

As can be seen from the table below, the 10 most highly suggested films for fans of “Iron Man (2008)” based on Pearson correlation.

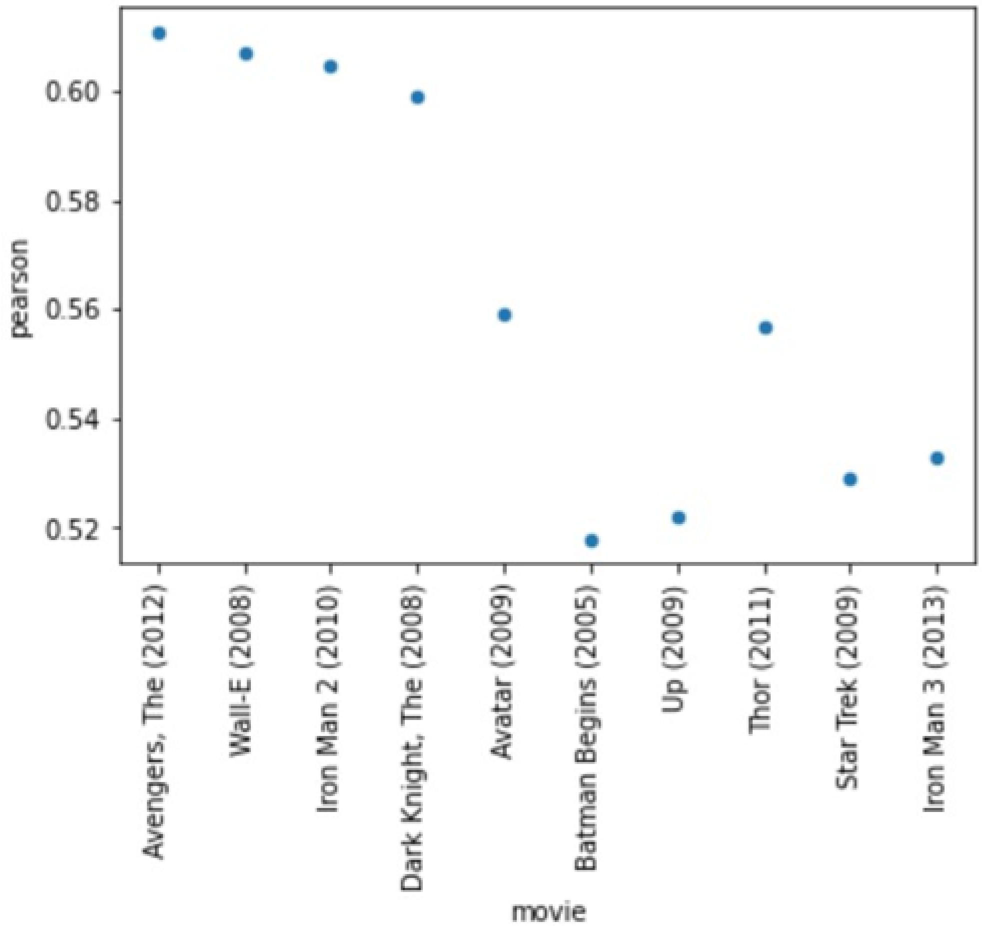

Fig 16. Top 10 recommendation for "Iron Man (2008) movie with Pearson Correlation

After that, we calculated the ten most highly suggested films by comparing their cosine similarity, as seen in the following list.

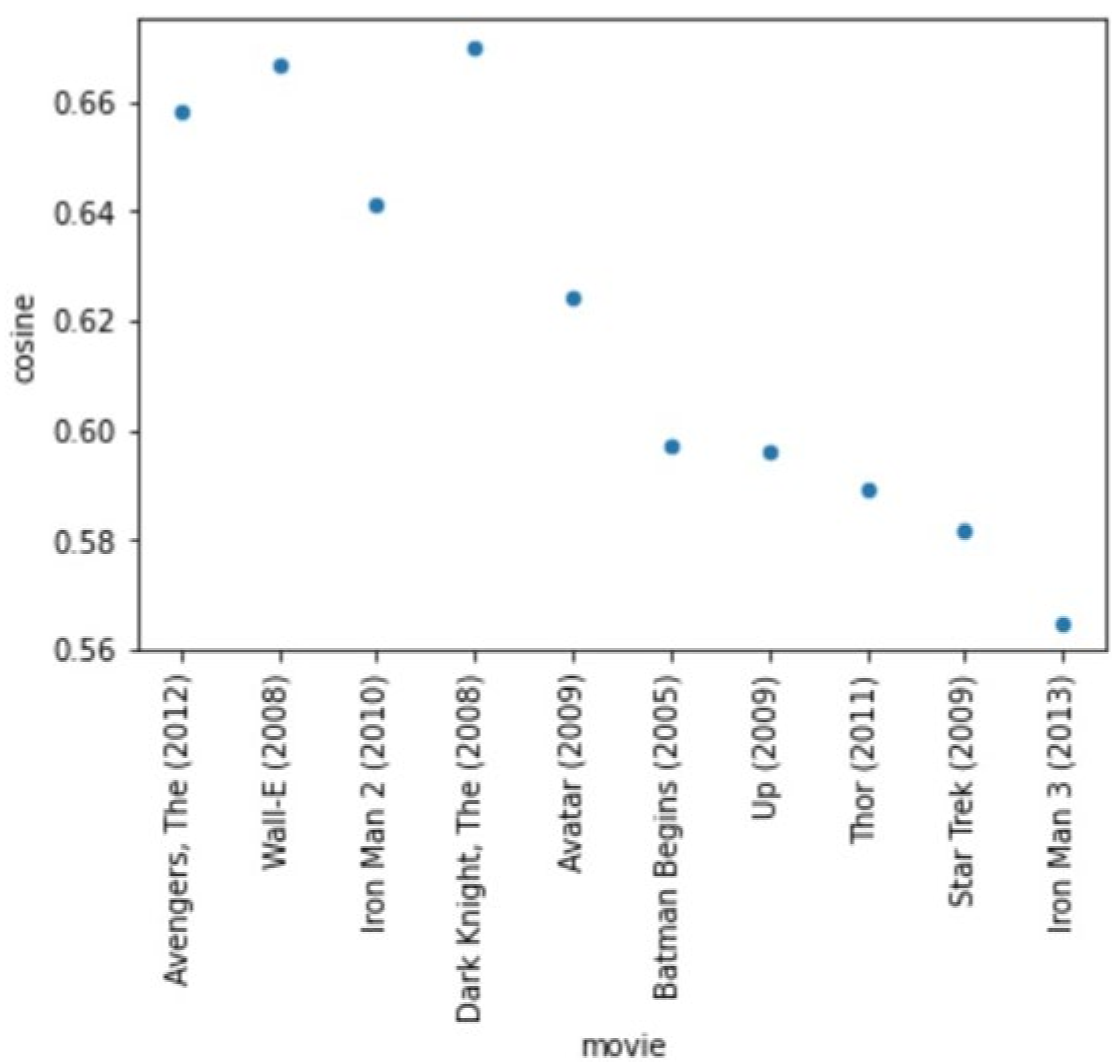

Fig 17. Top 10 recommendation for "Iron Man (2008) movie with Cosine Similarity

We can observe from the cross-validation that SVM [33] delivers a high RMSE score accuracy. KNN, Gradient Boosting, and Random Forest all have lower scores than this one, which is close to 0.7.

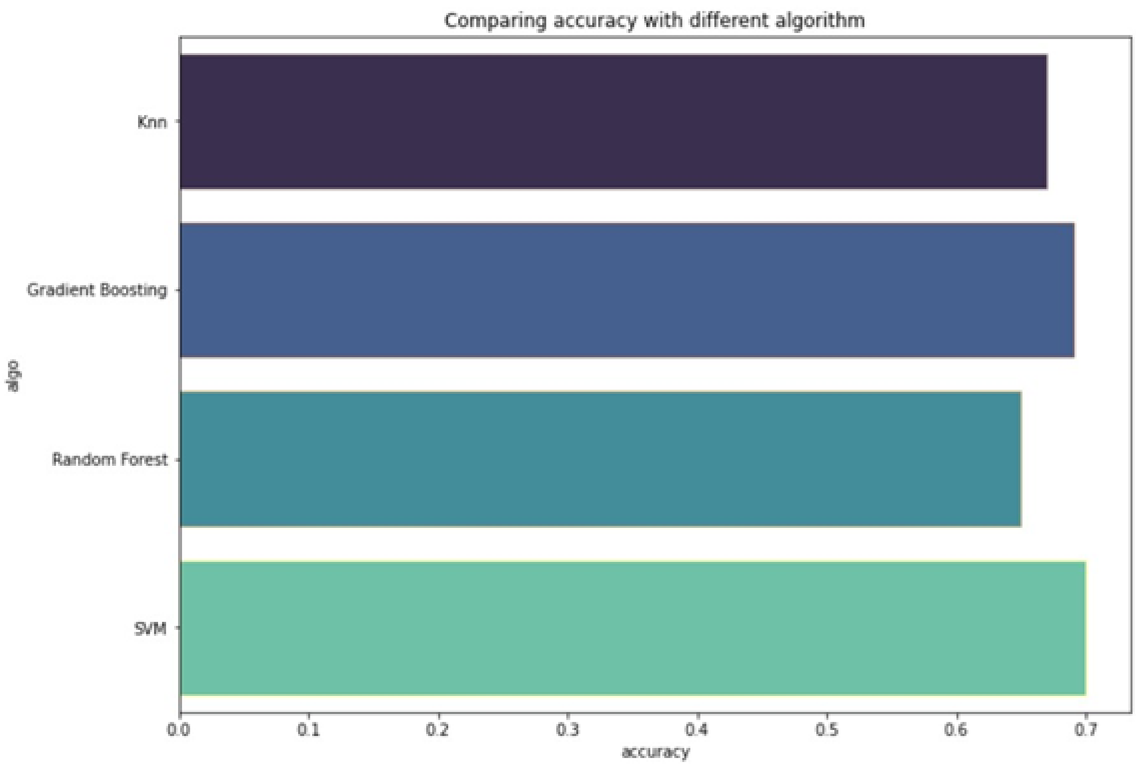

Fig 18. Mean Accuracy

We use the movie Lens dataset to see if our bagging regression is accurate or not [34]. The MovieLens official site has provided us with the information. The Entire Set: 9,000 movies received 100,000 evaluations and 3,600 tag applications from 600 individuals. Movies, user ratings, and other information are all part of the picture. users. Using a 5-star rating system, each user has at least 20 ratings. The recommendation system was implemented in Python. An anaconda was used in the construction of the project. A total of 70,000 datasets, including 30,000 that have been trained and evaluated, were used for movie lens newest tiny data.

5.2.1 Evolution Measures

Evaluation metrics such as Root Mean Squared Error (RMSE), Mean Absolute Error (MAE)were used to compare the performance of our new similarity measure and other traditional similarity measures (RMSE) [35].

Here, Predi is a prediction of a movie's rating, while ri is the actual rating, and |S| is its cardinality. As defined by the RMSE,

$$RMSE = \sqrt{\frac{1}{|S|}\sum_{i=1}^{S} \quad (Pred_i - r_i)^2}$$

It is defined as the Mean Absolute Error (MAE).

$$MAE = \frac{1}{|S|}\sum_{i=1}^{S} \quad |Pred_i - \ r_i|$$

The baseline RMSE score was 1.12. Below is a comparison of the RMSE scores of several models. Our following classifiers are named as follows: Support Vector Machine (SVM), Random Forest, Bagging Regressor. For comparison and prediction, these classifiers were utilized There is a classifier error report and output prediction result table for these classifiers. After that, we'll be able to look at the results.

Model	Root Mean Square Error	Mean Absolute Error
Linear Regression	1.09	0.9
Decision Tree	1.23	1.03
Random Forrest	1.14	0.91
Bagging Regression	1.33	0.96

Fig19. Comparison between result of different classifiers

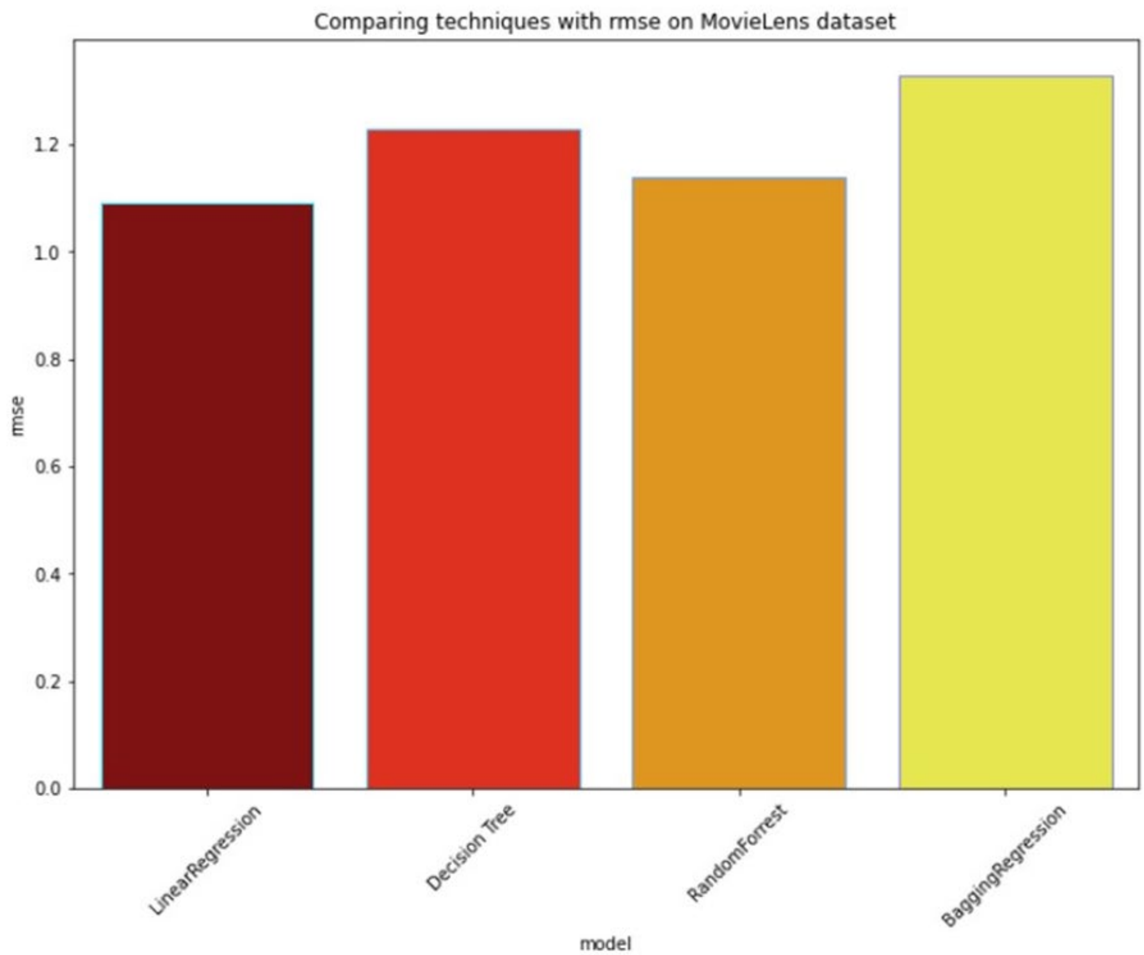

Fig 20. Comparison between Rmse result of different classifiers with bar plot

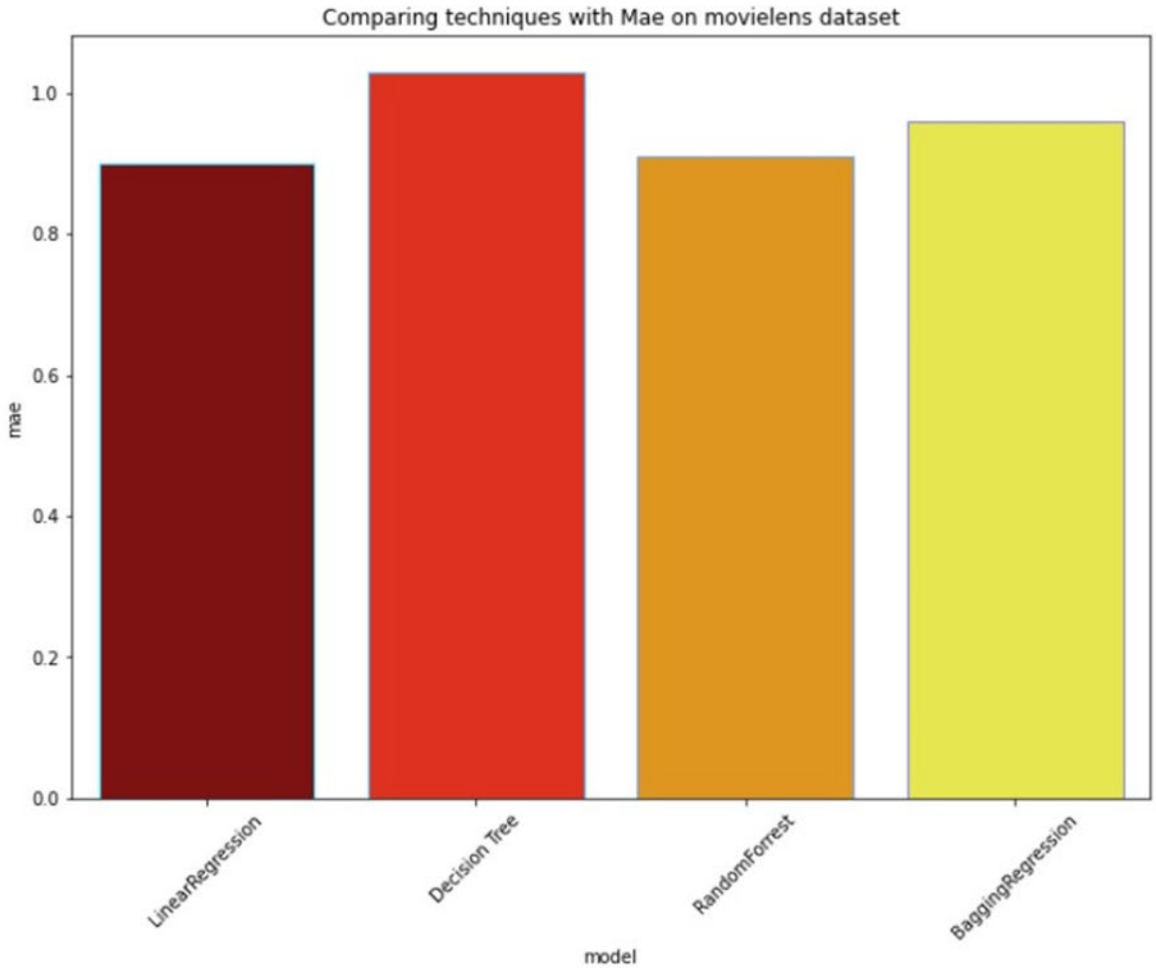

Fig21. Comparison between Mae result of different classifiers with bar plot

In the end, we can say that after all the hard work

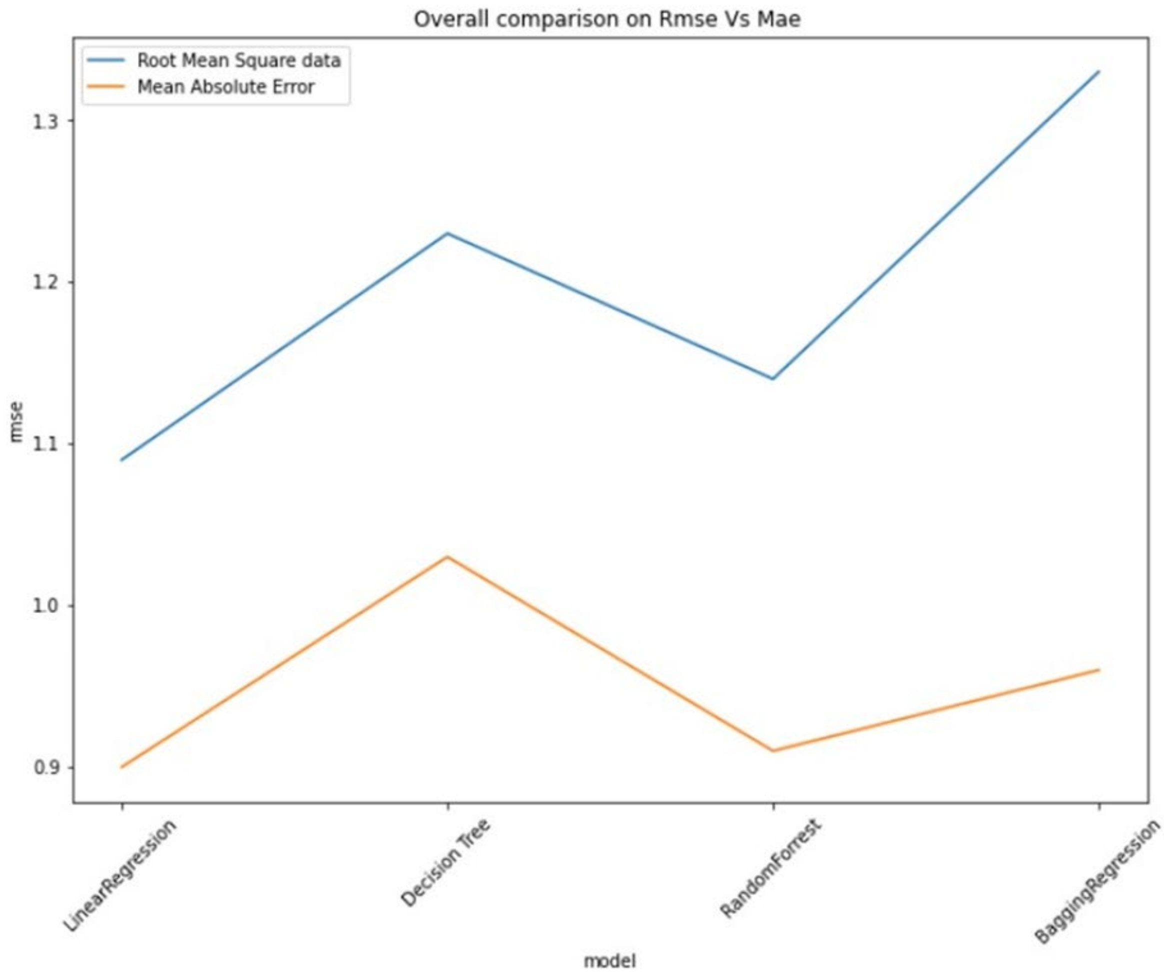

Fig 22. Comparison between Rsme vs Mae

5.2.2 Conclusion

In this chapter we have tried to show our overall results/progress of our project.

Chapter 6

6.1 Introduction

In this section, we'll take a quick look at the research's limitations and where it might go from here.

6.1.1 Future Work and Limitations

We'd like to eventually have a larger database. Our approach can be applied in a variety of fields, including music, film, news, books, and travel. In the future, we intend to employ a hybrid strategy in order to improve outcomes. Deep machine learning will also be used to reduce the sparsity of the data. For limited resources we did not get some accuracy data and also errors. We get accuracy and errors for just four classifiers. It is also the limitation for our project.

6.2 Conclusion

To now, many research have been conducted on this aspect of product recommendation, such as movies [2]. There is a growing need for more movie data because individuals can't easily find the movies, they want in the large movie archives [36]. In the fight against information overload, the recommendation system can help consumers find their favorite movies and give them a positive first impression.

Python code for diagram:

Bar Plot

```
import seaborn as sn

plt.subplots(figsize=(11,8))

sn.barplot(x="model",

y="rmse" ,data=compare,palette='hot',edgecolor=sn.color_palette('cool',7))

plt.xticks(rotation=45)

plt.title('Comparing techniques with rmse on MovieLens dataset')

plt.show()

plt.subplots(figsize=(11,8))

sns.lineplot(x = "model", y = "rmse", data = compare,label="Root Mean Square data")

sns.lineplot(x = "model", y = "mae", data = compare, label="Mean Absolute Error")

plt.xticks(rotation=45)

plt.ylabel("rmse")

plt.title('Overall comparison on Rmse Vs Mae ')

plt.show()
```

Scatter Plot:

```
pearson.plot.scatter(x='movie',y='pearson' );

plt.xticks(rotation=90)
```

References:

[1] H. H. Pyone, “Implementation of New Pearson Correlation Coefficient for Recommendation,” vol. 1, no. 1, pp. 136–140, 2020.

[2] A. K. Singh and P. B. Soundarabai, “Collaborative filtering in movie Recommendation System based on Rating and Genre,” *Ijarcce*, vol. 6, no. 3, pp. 465–467, 2017, doi: 10.17148/ijarcce.2017.63107.

[3] L. J. Chen, Z. K. Zhang, J. H. Liu, J. Gao, and T. Zhou, “A vertex similarity index for better personalized recommendation,” *Phys. A Stat. Mech. its Appl.*, vol. 466, pp. 607–615, 2017, doi: 10.1016/j.physa.2016.09.057.

[4] M. Deshpande and G. Karypis, “Item-based top-N recommendation algorithms,” *ACM Trans. Inf. Syst.*, vol. 22, no. 1, pp. 143–177, 2004, doi: 10.1145/963770.963776.

[5] Q. Jin, Y. Zhang, W. Cai, and Y. Zhang, “A new similarity computing model of collaborative filtering,” *IEEE Access*, vol. 8, pp. 17594–17604, 2020, doi: 10.1109/ACCESS.2020.2965595.

[6] X. Pan, G. Deng, and J. G. Liu, “Weighted bipartite network and personalized recommendation,” *Phys. Procedia*, vol. 3, no. 5, pp. 1867–1876, 2010, doi: 10.1016/j.phpro.2010.07.031.

[7] K. Inagaki *et al.*, “A case of hepatitis E virus infection in a patient with primary biliary cholangitis,” *Acta Hepatol. Jpn.*, vol. 58, no. 3, pp. 183–190, 2017, doi: 10.2957/kanzo.58.183.

[8] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, “Item-based collaborative filtering recommendation algorithms,” *Proc. 10th Int. Conf. World Wide Web, WWW 2001*, pp. 285–295, 2001, doi: 10.1145/371920.372071.

[9] E. A. Leicht, P. Holme, and M. E. J. Newman, “Vertex similarity in networks,” *Phys. Rev. E - Stat. Nonlinear, Soft Matter Phys.*, vol. 73, no. 2, pp. 1–10, 2006, doi: 10.1103/PhysRevE.73.026120.

[10] G. Salton and M. J. McGill, “Introduction to Modem Information,” pp. 375–384, 1983, [Online]. Available: http://portal.acm.org/citation.cfm?id=1893971.1894017.

[11] J. S. Breese, D. Heckerman, and C. Kadie, “Empirical Analysis of Predictive Algorithms for Collaborative Filtering,” pp. 43–52, 2013, [Online]. Available: http://arxiv.org/abs/1301.7363.

[12] U. Shardanand and P. Maes, “Social information filtering,” pp. 210–217, 1995, doi: 10.1145/223904.223931.

[13] M. Hasan and F. Roy, “An item–item collaborative filtering recommender system using trust and genre to address the cold-start problem,” *Big Data Cogn. Comput.*, vol. 3, no. 3, pp. 1–15, 2019, doi: 10.3390/bdcc3030039.

[14] R. H. Singh, S. Maurya, T. Tripathi, T. Narula, and G. Srivastav, “Movie Recommendation System using Cosine Similarity and KNN,” *Int. J. Eng. Adv. Technol.*, vol. 9, no. 5, pp. 556–559, 2020, doi: 10.35940/ijeat.e9666.069520.

[15] S. Rajarajeswari, S. Naik, S. Srikant, M. K. Sai Prakash, and P. Uday, “Movie Recommendation System,” *Adv. Intell. Syst. Comput.*, vol. 882, pp. 329–340, 2019, doi: 10.1007/978-981-13-5953-8_28.

[16] F. M. Harper and J. A. Konstan, “The MovieLens Datasets,” *ACM Trans. Interact. Intell. Syst.*, vol. 5, no. 4, pp. 1–19, 2016, doi: 10.1145/2827872.

[17] N. S. Sahni, S. Narayanan, and K. Kalyanam, “An Experimental Investigation of the Effects of Retargeted Advertising: The Role of Frequency and Timing,” *J. Mark. Res.*, vol. 56, no. 3, pp. 401–418, 2019, doi: 10.1177/0022243718813987.

[18] B. Mobasher, R. Cooley, and J. Srivastava, “Automatic Personalization Based on Web Usage Mining Architecture for Usage-based Web Personalization Mining Usage Data for Web Personalization,” *Commun. ACM*, vol. 43, no. 8, pp. 1–20, 2000.

[19] W. Lin, S. Alvarez, and C. Ruiz, “Collaborative recommendation via adaptive association rule mining,” ... *Int. Work. Web Min.* ..., 2000, [Online]. Available: http://ai.stanford.edu/users/ronnyk/WEBKDD2000/papers/alvarez.pdf.

[20] E. H. Han and G. Karypis, "Feature-based recommendation system," *Int. Conf. Inf. Knowl. Manag. Proc.*, no. January, pp. 446–452, 2005, doi: 10.1145/1099554.1099683.

[21] D. Heckerman, D. M. Chickering, C. Meek, R. Rounthwaite, and C. Kadie, "Dependency networks for inference, collaborative filtering, and data visualization," *J. Mach. Learn. Res.*, vol. 1, no. 1, pp. 49–75, 2001, doi: 10.1162/153244301753344614.

[22] J. Zhang, Y. Wang, Z. Yuan, and Q. Jin, "Personalized real-time movie recommendation system: Practical prototype and evaluation," *Tsinghua Sci. Technol.*, vol. 25, no. 2, pp. 180–191, 2020, doi: 10.26599/TST.2018.9010118.

[23] R. Nagamanjula and A. Pethalakshmi, "A novel scheme for movie recommendation system using user similarity and opinion mining," *Int. J. Innov. Technol. Explor. Eng.*, vol. 8, no. 4S2, pp. 316–322, 2019.

[24] C. Kaleli and H. Polat, "Privacy-preserving naïve Bayesian classifier-based recommendations on distributed data," *Comput. Intell.*, vol. 31, no. 1, pp. 47–68, 2015, doi: 10.1111/coin.12012.

[25] "memory based." https://towardsdatascience.com/how-to-build-a-model-based-recommendation-system-using-python-surprise-2df3b77ab3e5.

[26] "K-NN." https://www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning.

[27] "decision tree." https://www.javatpoint.com/decision-tree-induction.

[28] A. A. Kothari and W. D. Patel, "A Novel Approach Towards Context Based Recommendations Using Support Vector Machine Methodology," *Procedia Comput. Sci.*, vol. 57, pp. 1171–1178, 2015, doi: 10.1016/j.procs.2015.07.408.

[29] K. Oku, S. Nakajima, J. Miyazaki, and S. Uemura, "Context-aware SVM for context-dependent information recommendation," *Proc. - IEEE Int. Conf. Mob. Data Manag.*, vol. 2006, p. 109, 2006, doi: 10.1109/MDM.2006.56.

[30] "movieLens Dataset." https://grouplens.org/datasets/movielens/.

[31] J. Fraser and O. Lemon, "Spoken Conversational AI in Video Games – Emotional Dialogue

Management Increases User Engagement," pp. 179–184.

[32] D. Roobaert, G. Karakoulas, and N. V. Chawla, "Information gain, correlation and support vector machines," *Stud. Fuzziness Soft Comput.*, vol. 207, no. 2006, pp. 463–470, 2006, doi: 10.1007/978-3-540-35488-8_23.

[33] A. Dypiemr and A. Dypiemr, "RECOMMENDATION SYSTEM USING KNN AND COSINE SIMILARITY," vol. 8, no. 9, pp. 1256–1260, 2020.

[34] H. Linusson and A. Olausson, "Random Forest Using Sparse Data Structures," 2012.

[35] Suryakant and T. Mahara, "A New Similarity Measure Based on Mean Measure of Divergence for Collaborative Filtering in Sparse Environment," *Procedia Comput. Sci.*, vol. 89, pp. 450–456, 2016, doi: 10.1016/j.procs.2016.06.099.

[36] J. Chen, C. Zhao, Uliji, and L. Chen, "Collaborative filtering recommendation algorithm based on user correlation and evolutionary clustering," *Complex Intell. Syst.*, vol. 6, no. 1, pp. 147–156, 2020, doi: 10.1007/s40747-019-00123-5.