ISSN No:-2456-2165

Evaluation of Models Based on Movie Recommendation System

Aditya Kumar Yadav
UG Scholar
PSG College of Technology

Nallaperumal.K
UG Scholar
PSG College of Technology

Abstract:- The movie recommendation system is an information filtering tool, which is mainly based on big data to predict the ratings of users and articles in order to recommend their preferences. The movie recommendation system provides a mechanism to help users rank other users with similar interests. It is a major part of e-commerce websites and applications. The project focuses on the evaluation of different models and algorithms, and its main purpose is to compare different algorithms (such as collaborative filtering) and models such as slope 1 etc. It also compares with existing methods and analysis and interprets the results. The average absolute error (MAE), standard deviation (SD), root mean square error (RMSE) and t value of the movie recommendation system give better results because our method provides lower error values. The film lens experiment data set can help you find the best method to achieve high performance in terms of reliability and efficiency, and provide accurate and personalized film recommendations for current methods.

***Keywords**:- Recommendation System, Evaluation of Recommendation Systems, Collaborative Filtering, Content Based Filtering, Slope One, k Fold Validation, Precision and Recall, f1 Score*

I. INTRODUCTION

With a huge volume of data available at present in the world, organizations are looking for ever more proper ways to use these data. Based on the similarities between the films, the systems can make predictions for the evaluation of a new film and rating data to predict how other users will like a particular film. Recommendation systems are ubiquitous today and try to benefit from customers and meet their needs properly. To achieve this, the systems must analyze a large volume of data and collect it, sometimes from various sources, and anticipate how a user will like the film. This takes a considerable amount of computing power. Additionally, organizations are try to avoid sending customers' inboxes with a number of products every morning, leaving them waiting for an email or text message that gets the customer to search and act. Recommendation systems are ubiquitous in today's consumer life of data to predict the preferences of a user or their similarity to the other user group is the most important part of a recommendation system. Collaborative, content-based and hybrid filtering are approaches to use a recommendation system. Many algorithms can be applied to data in predicting the preference of a user. User based, element based, and model-based methods are some of the ways of predicting the preferences a user using the number of items, users or groups in each. Let's find out how the feature performs and discuss the algorithms that are used to build a product recommendation system (movie) to help users find products (movies, articles) they haven't found (seen) yet. The movie recommendation system) aims to provide customers (viewers) with the most relevant products (films) they are looking for. Such a specific approach should generate a high conversion rate and make online advertising very effective and fluid. We are going to study how to create an effective recommendation system that can predict the products (movies) that customers will like. With the results of some existing models and algorithms, we can improve these application-specifically and design five recommendation systems, article (film) similarity, user similarity, content-based and two-based models. They can be used to predict the rating of a product (movie) that a customer has never rated (seen) based on the data (ratings) of all other users (viewers) and their ratings in the system.

We used a total of five algorithms for the recommendation system. There are 2 styles of collaboarative filtering algorithms: memory-based techniques and model-based techniques. The memory-based technique simply remembers all the points and makes suggestions supported the connection between the client item and therefore the rest. Within the case of model-based strategies, the prediction of the parameterized model should 1st match the analysis matrix, and so build recommendations supported the tailored model. The 2 hottest memory-based strategies ar user-based and item-based collaboarative filtering. These strategies ar samples of neighbor-based strategies that are associated with user ratings or similar things and build recommendations supported the weighted add of the ratings of the nighest users/items. For a few positions, they need similar ratings for alternative positions. The same applies to element-based CF, however in terms of components. Construct a constant model and advocate the highest-level things came by the model. For instance, the Slope One technique examines a straight forward set of predictors, with only 1 constant variable. So this variable represents the common distinction between the values of the 2 things. Adopting this technique, we will simply deliver the goods quick calculations and affordable accuracy.

II. DATA UNDERSTANDING

2.1 THE MOVIE LENS DATA SET

This data set (ml-latest-small) describes 5-star ratings and promotions with free Movie Lens text tags, a movie recommendation service which includes 10,0004 ratings of 9,125 movies and 1,296 tag applications. These data were created by exactly 671 users between the time period 09/01/1995 and 16/10/2016. Users are randomly selected and all selected users have rated at least 20 movies, demographic information is not included and each user is represented by an identifier and no further information is provided. The data are contained in the files: links.csv, movies.csv, ratings.csv and tags.csv.

2.2 User Ids

Movie Lens users are randomly selected and the IDs will be anonymized. The user IDs in the files rating.csv and tags.csv are the same (that is, the same ID refers to the same user in both files).

2.3 Movie Ids

The data set only contains movies with at least one rating or label. These movie IDs are the same as those used on the Movie Lens website (for example, ID 1 is the URL https://moviens.org/movies/1). The movie IDs among rating.csv, tags.csv, movies.csv, and links.csv are the same (that is, in these four data files, the same ID refers to the same movie).

2.4 Ratings Data File Structure (ratings.csv)

All ratings are included in the rating.csv file. Each line after the title line in this file represents the user's movie rating, and has the following format: user ID, movie ID, rating and timestamp. The lines in this file are first sorted by user ID, and then by movie sorting ID within the user. A 5-star rating with half-star steps (0.5-5.0 stars). The timestamp represents the number of seconds since midnight UTC on January 1, 1970.

2.5 Movies Data File Structure (movies.csv)

The movie information is contained in the movies.csv file. Each line in this file after the header represents a movie and has the following format: Movie ID, Title and Genres. Movie titles are entered manually or imported from https: //www.themoviedb. org / and includes the year of publication in parentheses. These titles can contain errors and inconsistencies. The genres are a bar-separated list and are selected from the following:

- Action
- Adventure
- Animation
- Children's
- Comedy
- Crime
- Documentary
- Drama
- Fantasy
- Film-Noir
- Horror
- Musical
- Mystery
- Romance
- Sci-Fi
- Thriller
- War
- Western
- (no genres listed)

III. DATA PREPARATION

3.1 Pivot Transformation

Normalized data set being transformed into less normalized one is Pivot transformation. For instance, a normalized order data set that lists products, customer names and purchase quantities which usually contains numerous rows for each customer who purchases more than one product, and each row of the customer displays order related information for various products. In the product column, Pivot Transform can create a data set with only one row for each customer. The name of the product purchased by the customer is displayed as the column heading, and the number of products purchased is displayed as the value in the product column. Since all customers do not buy all products, many columns might contain null values. When rotating a data set, the input column plays a different role in the rotation process. You can refer to columns in the following ways:

- Pass the columns to the output as they are. Since many inputs can only produce one output row, the transformation takes only the first input value of a column.
- A key or part of a key that defines a series of records is the column.
- The pivot point is defined by the column. These column values are linked to the columns in the pivot data set.
- The values in the column are suitable for the column generated by the Pivot Chart. This conversion has an input, a normal output, and an error output

3.2 Sort and Duplicate Rows

In order to pivot data efficiently, i.e. to create less records as possible in the output dataset, the input data in the pivot column has to be sorted. If in case the data is unsorted, multiple values/rows for each data can be generated by the pivot transformation in the set key, which is membership of the set defined by the column. For instance, if a data set is panned in a Name column however the names don't seem to be sorted, the output dataset could have over one row for every customer since a pivot occurs all the time when Name changes. The input data can contain duplicate rows, causing the pivot transform to fail. "Duplicate rows" means rows that have the same values in the set key columns and the pivot columns. To avoid errors, you can configure the transform to redirect the error lines to an error output, or you can add values beforehand to make sure there are no duplicate lines.

IV. DATA MODELING

4.1 TRADITIONAL COLLABORATIVE FILTERING

The traditional collaborative filtering methods includes three types. They are Item-based, Model-based and User-based collaborative filtering methods. We are going to be using the following notations to show how these methods work.

Let U be a set of N users and I be a set of M items.

4.1.1 USER-BASED COLLABORATIVE FILTERING:

In this methodology, we are predicting the behavior of a user against an item using the weighted total of deviations from the mean ratings of the users that antecedently rated this item and the user mean rate. First, we use the following formula for computing user mean rate:

$$\overline{v_u} = \frac{\sum_{i \in S_u} v_{ui}}{|S_u|}$$

FIG 1. MEAN RATE FORMULA

The aforementioned weight can be computed using Pearson correlation in keeping with the subsequent formula

$$w(a,u) = \frac{\sum_{i \in S_a \cap S_u} (v_{ai} - \overline{v_a})(v_{ui} - \overline{v_u})}{\sqrt{\sum_{i \in S_a \cap S_u} (v_{ai} - \overline{v_a})^2 \sum_{i \in S_a \cap S_u} (v_{ui} - \overline{v_u})^2}}$$

FIG 2. PEARSON CORRELATION FORMULA

The prediction formula is shown below

$$p_{ai} = \overline{v_a} + \frac{\sum_{\{u \in U | i \in S_u\}} w(a,u) \times (v_{ui} - \overline{v_u})}{\sum_{\{u \in U | i \in S_u\}} |w(a,u)|}$$

FIG 3. PREDICTION FORMULA BAESD ON USER BASED COLLABORATIVE FILTERING

4.1.2 ITEM-BASED COLLABORATIVE FILTERING:

When the recommendation system used by Amazon was reviewed, we came up with the finding that it did not use the traditional way of collaborative filtering algorithm mentioned above. To explain, user-based and cluster models are not used in Amazon's recommendation system for many reasons. Due to expensive calculation of O (MN), where M is the number of similar users and N is the number of elements in common with these users, Amazon has chosen not to use these methods. The use of clusters has been suggested to reduce the number of elements and users in order to solve the major computing problems. However, this reduces the quality of the recommendations. In other words, if the method compares the user to a small sample, the similarity is not accurate. In addition, by dividing items into the item space, recommendations are limited to certain product types. If the cluster doesn't contain the popular or disliked items, they will never be recommended to users. If the user has already bought these items, they will never be recommended to them. Let us apply Item-based Collaborative filtering to show similar items to the user once they have selected a specific item using the custom cosine formula:

$$sim(i,j) = \frac{\sum_{\{u \in U | i \in S_u \& j \in S_u\}} (v_{ui} - \overline{v_u})(v_{uj} - \overline{v_u})}{\sqrt{\sum_{\{u \in U | i \in S_u \& j \in S_u\}} (v_{ui} - \overline{v_u})^2 \sum_{\{u \in U | i \in S_u \& j \in S_u\}} (v_{uj} - \overline{v_u})^2}}$$

FIG 4. CUSTOM COSINE FORMULA

In Addition, we have predicted how the user will rate the item using the previous similarity:

$$p_{ai} = \frac{\sum_{\{j \in S_a | j \neq i\}} sim(i,j) \times v_{aj}}{\sum_{\{j \in S_a | j \neq i\}} |sim(i,j)|}$$

FIG 5. PREDICTION FORMULA BAESD ON ITEM BASED COLLABORATIVE FILTERING

4.1.3 CONTENT-BASED FILTERING

Content-based filtering methods use many discrete element characteristics to recommend additional elements with similar attributes. These methods are usually combined (see Hybrid Recommendation System). By comparing two popular music recommendation systems: Last.fm and Pandora Radio, the difference between collaborative filtering and content filtering can be proved. Last.fm has created a featured song "Radio", which tracks the group and individual tracks that users frequently hear and compares them with the listening habits of other users. .Last.fm plays tracks that are not in the user's library but are often played by other users with similar interests. Because this method takes advantage of user behavior, it is an example of collaborative filtering technology. Pandora uses attributes of Songs or artists (a subset of the 400 attributes provided by the Music Genome Project) to create "stations" that play music with similar characteristics. User comments are used to optimize the results of a radio station by downplaying certain attributes when the user does not like a particular song and highlighting other attributes when the user likes the song. This is one example of content-based method. It has its own advantages and disadvantages. In the above example, Last.fm needs a lot of user information to give accurate recommendations. This is an example of common cold start problem in collaborative filtering systems. There is little information at the beginning and much more limited scope (for example, you can only make suggestions similar to the original seed). Recommendation engines are a useful alternative to search algorithms because they can help users find articles they might not have found. It should be noted that recommendation systems are usually implemented using search engines that index non-traditional data. Content-based filtering methods use many discrete element attributes to recommend other elements with similar attributes. These methods are usually combined (see Hybrid Recommendation System). By comparing two popular music recommendation systems: Last.fm and Pandora Radio, the difference between collaborative filtering and content filtering can be proved. Last.fm has created a

ISSN No:-2456-2165

featured song "Radio" to track the group and individual tracks that users listen to regularly and compare them with the listening habits of other users. Last.fm plays tracks that are not in the user's library but are often played by other users with similar interests. Because this method takes advantage of user behavior, it is an example of collaborative filtering technology. Pandora uses song or artist attributes (a subset of the 400 attributes provided by the Music Genome Project) to create a "station" that plays music with similar attributes. User comments are used to optimize the results of a radio station by downplaying certain attributes when the user does not like a particular song and highlighting other attributes when the user likes the song. This is an example of a content-based approach. Each type of system has their own advantages as well as disadvantages. In the above example, Last.fm needs a lot of user information to make accurate recommendations. This is an example of a cold problem. And it often appears in collaborative filtering systems. Although Pandora requires very little information to get started, its scope is much more limited (for example, it can only provide recommendations similar to the original seed). Alternatives to search algorithms because they can help users find items they can't find. It should be noted that recommendation systems are usually implemented using search engines that index non-traditional data.

4.1.4 SLOPE ONE

Slope One was first presented by Daniel Lemire and Anna Maclachlan in the document "Slope One Predictors for Online Rating-Based Collaborative Filtering". This makes it very easy to implement and use, and the precision of this algorithm equals the precision of more complicated and resource-intensive algorithms. Slope One's method works with average grade differences between the individual elements and makes predictions based on their weighted value. Slope One uses a simple linear regression model to solve the data scarcity problem. Combined with the similarities of the users, the k-neighborhood method can optimize the quality of the ratings of the users involved in the prediction. Based on the Slope One algorithm, a new collaborative filter algorithm is presented that combines unsafe neighbors with Slope One. Firstly, different numbers of neighbors are selected dynamically for each user according to the similarities with other users. Second, the average deviations between pairs of relevant items are generated based on the ratings of neighboring users. Finally, the evaluations of the objects are predicted using a linear regression model. Experiments with the Movie Lens data set show that the proposed algorithm offers better recommendation quality and is more robust against data scarcity than Slope One. It also outperforms some other collaborative filtering algorithms in predictive accuracy.

4.1.5 HYBRID MODEL

This assumes that users have preferences for an explicit variety of product, therefore we have a tendency to try and advocate a product that's like what the user thought they liked. Again, the goal is to produce alternatives or replacements for the item being displayed. A user profile is important to see what a user prefers and might be created supported user preferences or viewing behavior. First, we'll convert the ratings to binary format to stay things easy. Ratings of four and five square measure mapped to one, that represents "Likes," and ratings of three or less square measure appointed to -1, that represents dislikes. To form the easy user profile matrix, we have a tendency to cypher the real of the genre matrix of the moving-picture show and therefore the binary rating matrix. Before continuing with the real, you'll realize that the films knowledge set containing 9125 Movies, however the Ratings knowledge set solely contains 9065 Movies. To repair this, we're removing movies that haven't been rated from the genre matrix. User profiles show the mass inclination of every user towards moving-picture show genres. Every column represents a singular user ID, and positive values indicate a preference for a selected gender. The values were yet again simplified to a binary matrix: positive values were appointed one to represent likes, negative values were appointed zero to represent dislikes. A representing a user's preference for the characteristic of an item. We have a tendency to selected Jaccard Distance to live the similarity between user profiles and therefore the movie's genre matrix. Jaccard Distance was our metric of alternative as a result of it had been appropriate for binary knowledge.

V. EVALUATION

Cross-validation, sometimes called rotation estimation, is a model validation technique used to assess how the results of a statistical analysis are generalized to an independent data set. It is mainly used in environments where the goal is prediction and you want to gauge how accurately the predictive model will work in practice. Typically, when a prediction problem occurs, a model is given a set of known data to be trained on (training data set) and a set of unknown data (or data that is being displayed for the first time) to test the model against (called the validation data set or test set). The goal of cross-validation is to define a data set to "test" the model in the training phase (i.e. the validation set) to limit problems like overfitting, to give an idea of how to generalize the model to an independent data set (i.e. an unknown data set, e.g. from a real problem) etc. A cross-validation round comprises dividing a data sample into complementary subsets, performing the analysis analysis on one subset (so-called training set) and validating the analysis on the other subset (so-called Validation kit or test kit). Methods are performed multiple rounds of cross-validation using different partitions, and the results of the validation are combined (e.g., averaged) during the rounds to estimate a final predictive model. One of the main reasons to use cross validation instead of traditional validation (Splitting the data set into two sets of 70% for training and 30% for testing) is that there isn't enough data available to break it down into separate training and test sets without losing significant modeling or testing skills. To properly estimate the performance of model prediction, cross-validation must be used as a powerful general technique.

ISSN No:-2456-2165

5.1 K FOLD VALIDATION:

With the k-fold cross-validation, the original sample is randomly divided into k sub-samples of equal size. Of the k sub-samples, a single sub-sample pattern is kept as validation data for testing the model, and the remaining k-1 sub-samples are used as training and used exactly once as validation data. The k-scores of the folds can be averaged to get a single estimate. This repeated random sub sampling method (see below) is that all observations are used for both training and validation, and each observation is used exactly once for validation. For example, the setting of k = 2 leads to a double cross-validation. In a double cross-validation, we randomly shuffle the data set into two sets d0 and d1 so that both sets are the same size (this is usually implemented by shuffling the matrix data and then dividing it into two). We then train on d0 and validate on d1, followed by training on d1 and validation on d0.

5.2 PRECISION AND RECALL

Precision (also called positive predictive value) is the fraction of relevant instances among the retrieved instances, while **recall** (also known as sensitivity) is the fraction of relevant instances that have been retrieved over the total amount of relevant instances. Both precision and recall are therefore based on an understanding and measure of relevance.

		True condition	
	Total population	Condition positive	Condition negative
Predicted condition	Predicted condition positive	True positive, Power	False positive, Type I error
	Predicted condition negative	False negative, Type II error	True negative

FIG 6. CONFUSION MATRIX

5.3 F1 SCORE:

A test's accuracy is measured using the F1 Score. It considers both the exactitude p and also the recall r of the check to figure the score: p is that the range of correct positive results divided by the quantity of all positive results came back by the classifier, and *r* is the number of correct positive results divided by the number of all relevant samples (all samples that should have been identified as positive). The F1 score is that the harmonic average of the preciseness and recall, wherever associate degree F1 score reaches its best price at one (perfect preciseness and recall) and worst at zero.

VI. RESULT

1. Item based KNN model

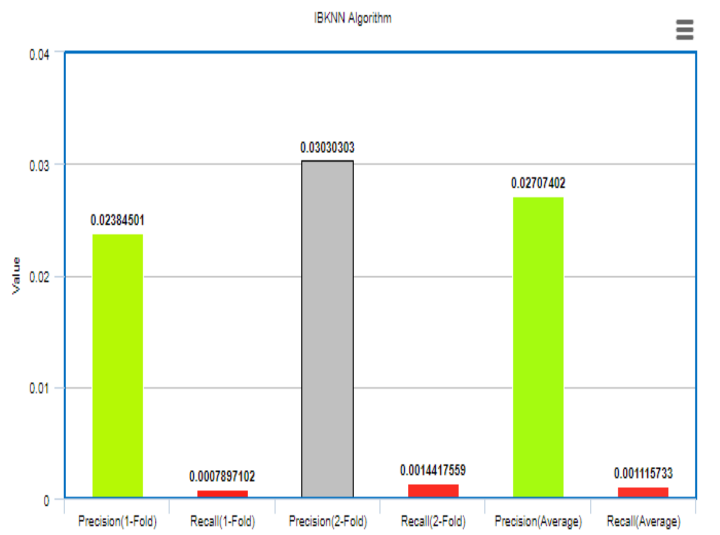

FIG 7. EVALUATION OF IBKNN MODEL

2. Slope One model

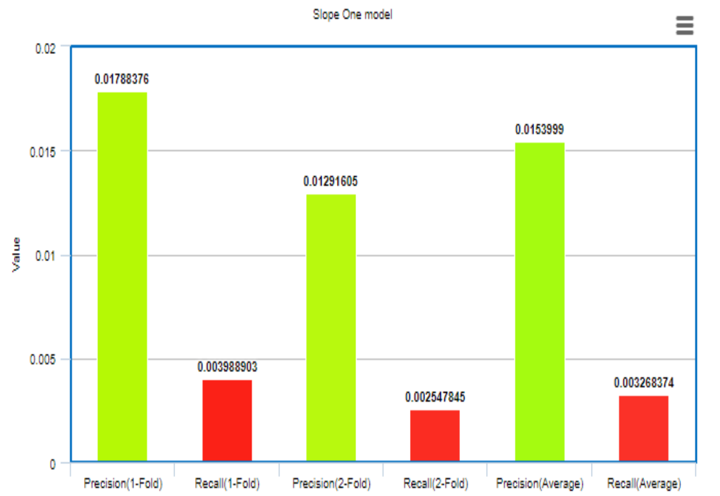

FIG 8. EVALUATION OF SLOPE ONE MODEL

VII. CONCLUSION

It is still a question mark that which recommendation strategies and approaches are the most promising. Occasionally content-based filtering methodology performed higher than the collaborative filtering methodology, and typically it performed worse. We tend to came up with 3 potential reasons for the anomaly of the results.

(A) Many evaluations had limitations. They were supported powerfully cropped data sets, few participants in user studies, or failed to use the correct baselines.

(B) Adopting totally different implementations of identical recommendations approaches, cause variations within the results.

(C) We tend to speculate that minor variations within the data sets, algorithms, or user populations inevitably cause durable variations within the performance of the approaches.

Hence, finding the foremost proper approaches may be a challenge. The second main limitation, we tend to note that several algorithms neglected to require into consideration factors aside from accuracy, for instance overall user satisfaction. But for a sparse data set like Movie Lens we find that the slopeOne model and hybrid model have high accuracy.

REFERENCES

[1]. Gomez-Uribe, Carlos A.; Hunt, Neil (28 December 2015). "The Netflix Recommender System". *ACM Transactions on Management Information Systems*.
[2]. Alan; Bellogín, Alejandro (2014-10-01). *Comparative recommender system evaluation: benchmarking recommendation frameworks*. *Proceedings of the 8th ACM Conference on Recommender Systems*.
[3]. Powers, David M. W. (2011). "Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness & Correlation". *Journal of Machine Learning Technologies*. **2**(1): 37–63
[4]. Olson, David L.; and Delen, Dursun (2008); *Advanced Data Mining Techniques*, Springer, 1st edition (February 1, 2008.