International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 04, April-2020

Machine Learning Model for Movie Recommendation System

M. Chenna Keshava
Assistant Professor, Dept of CSE,
JNTUACE, Pulivendula, AP, India

P. Narendra Reddy
Student, Dept of CSE, JNTUACE,
Pulivendula, AP, India

S. Srinivasulu
Student, Dept of CSE, JNTUACE,
Pulivendula, AP, India

B. Dinesh Naik
Student, Dept of CSE, JNTUACE,
Pulivendula, AP, India.

***Abstract*— The primary aim of recommendation systems is to recommend applicable objects to a consumer-based totally on ancient data. If a movie is rated excessive by means of a consumer who also watched the movie you are watching now, it's miles possibly to show up inside the recommendations. The films with the highest overall scores are in all likelihood to be enjoyed by way of nearly everyone. The algorithm which does all these features is called CineMatch. For personal users, it also learns from the conduct of the person to higher expect a movie the consumer is anticipated to be fascinated in. Here we have to increase our CineMatch algorithm 10% by using fashionable collaborative filtering techniques.**

Keywords—Machine learningmodels,Movies,Ratings,Similarity matrix,Sparse matrix.

I. INTRODUCTION

A. Motivation and Scope

We are leaving the age of facts and coming into the age of recommendation. Like many device mastering techniques, a recommender system makes a prediction based on users' ancient behaviors. Specifically, it's to expect user choice for a fixed of items based totally on past experience.

B. Need to study

Recommendation systems are getting increasingly important in today's extraordinarily busy world. People are always short on time with the myriad duties they need to accomplish within the restrained 24 hours. Therefore, the recommendation structures are vital as they help them make the right choices, without having to dissipate their cognitive resources. The reason for a recommendation system essentially is to look for content that would be thrilling to an individual. Moreover, it includes a number of things to create customized lists of beneficial and exciting content unique to every user/individual. Recommendation structures are Artificial Intelligence primarily based algorithms that skim thru all possible alternatives and create a customized listing of objects which might be thrilling and relevant to an individual.

C. Literature Survey/Review of Literature

- The two principal tasks addressed by way of collaborative filtering techniques are rating prediction and rating. In contrast, ranking fashions leverage implicit feedback (e.g. Clicks) so that you can offer the user with a customized ranked listing of encouraged items [1].
- With the increasing need for retaining confidential statistics at the same time as supplying tips, privacy-maintaining Collaborative filtering has been receiving increasing attention. To make statistics proprietors experience more comfortable even as imparting predictions, various schemes were proposed to estimate pointers without deeply jeopardizing privacy. Such methods dispose of or reduce statistics proprietors' privacy, financial, and legal concerns by means of employing exceptional privacy-retaining techniques [2].
- In the spread of information, the way to quickly locate one's favorite film in a massive variety of movies end up a very essential issue. Personalized recommendation machines can play a crucial role in particular whilst the person has no clean target movie. [3].
- In this paper, we design and implement a movie recommendation machine prototype blended with the actual wishes of movie recommendation thru gaining knowledge of KNN algorithm and collaborative filtering algorithm [4].
- In this study, we examine a privacy-retaining collaborative filtering method for binary facts referred to as a randomized reaction technique. We develop a method focused on the second thing of privacy to find out faux binary rankings the usage of auxiliary and public information [5].
- If privacy measures are provided, they may decide to grow to be worried about prediction generation processes. We advocate privacy-maintaining schemes getting rid of e-commerce sites' privateness concerns for imparting predictions on allotted data [6].
- With the improvement of the Internet and e-commerce, the recommendation machine has been widely used. In this paper, the electronic commerce recommendation system has a similar look at and makes a specialty of the collaborative filtering algorithm in the utility of personalized film recommendation system [7].

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 04, April-2020

II. RESEARCH GAP

The data set provided quite a few rating information, and a prediction accuracy bar this is 10% better than what Cinematch algorithm can do on the equal training data set. (Accuracy is a measurement of the way closely predicted scores of films in shape subsequent actual rankings).And we have to Predict the score that a consumer would supply to a movie that she or he has not yet rated. And also Minimize the difference between predict and the actual score.

III. RESEARCH METHODOLOGY

A. *User-Item Sparse Matrix*

In the User-Item matrix, each row represents a person and every column represents an object and every cell represents rating given with the id of a user to an item.

B. *User-User Similarity Matrices*

Here, two customers could be similar to the premise of the comparable ratings given with the id of each of them. If any two users are similar then it means both of them have given very comparable scores to the items due to the fact here the consumer vector is nothing however the row of a matrix which in flip contains rankings given through user to the items. Now considering cosine similarity can variety from '0' to '1' and '1' means the highest similarity, so consequently, all the diagonal elements could be '1' because the similarity of the consumer with him/herself is the highest. But there's one hassle with user-user similarity. User alternatives and tastes change over time. If any consumer favored some item one year in the past then it isn't important that he/she will like the identical object even today.

C. *Item-Item Similarity Matrix*

Here, two items can be comparable to the idea of the comparable rankings given to each of the items via all of the users. If any two gadgets are comparable then it means both of them had been given very comparable ratings by means of all of the users due to the fact here the item vector is nothing however the column of the matrix which in flip contains scores given with the aid of consumer to the objects. Now due to the fact cosine similarity can variety from '0' to '1' and '1' means the highest similarity, so consequently, all of the diagonal elements might be '1' due to the fact the similarity of an item with the identical item is the highest.

D. *Cold Start Problem*

The cold start problem concerns the personalized guidelines for users without a few past histories (new users). Providing suggestions to users with small beyond history turns into tough trouble for CF models due to the fact their studying and predictive ability is limited.

IV. SURPRISE LIBRARY MODELS

A. *XGBoost*

However, with regards to small-to-medium structured/tabular data, choice tree primarily based algorithms are taken into consideration best-in-class proper now.. XGBoost and Gradient Boosting Machines (GBMs) are each ensemble tree techniques that follow the precept of boosting susceptible learners using the gradient descent architecture.

B. *Surprise Baseline*

This Algorithm predicting a random rating based totally on the data.

Predicted rating: (baseline prediction)

$$\hat{r}_{ui} = b_{ui} = \mu + b_u + b_i$$

μ : Average of all trainings in training data.
bu : User bias.
bi : Item bias (movie biases)

C. *Suprise KNNBaseline Predictor*

It is a number one collaborative filtering algorithm considering a baseline rating.

Predicted Rating**: (**based on User-User similarity)

$$\hat{r}_{ui} = b_{ui} + \frac{\sum_{v \in N_i^k(u)} \text{sim}(u,v) \cdot (r_{vi} - b_{vi})}{\sum_{v \in N_i^k(u)} \text{sim}(u,v)}$$

This is exactly same as our hand-crafted features 'SUR'- 'Similar User Rating'. Means here we have taken 'k' such similar users 'v' with user 'u' who also rated movie 'i'. $r_{vi}$is the rating which user 'v' gives on item 'i'. *bvi* is the predicted baseline model rating of user 'v' on item 'i'.Generally, it will be cosine similarity or Pearson correlation coefficient.

Predicted rating (based on Item Item similarity):

$$\hat{r}_{ui} = b_{ui} + \frac{\sum_{j \in N_u^k(i)} \text{sim}(i,j) \cdot (r_{uj} - b_{uj})}{\sum_{j \in N_u^k(j)} \text{sim}(i,j)}$$

D. *Matrix Factorization SVD*

The Singular-Value Decomposition, or SVD for short, is a matrix decomposition technique for decreasing a matrix to its constituent elements in order to ensure the next matrix calculations simpler. The SVD is used broadly both within the calculation of different matrix operations, including matrix inverse, but also as a statistics reduction approach in machine learning.

Predicted Rating:

$$\hat{r}_{ui} = \mu + b_u + b_i + q_i^T p_u$$

qi — Representation of item(movie) in latent factor space.
pu — Representation of user in new latent factor space**.**

E. *Matrix Factorization SVDpp*

Here, an implicit rating describes the fact that a consumer u rated an item j, regardless of the rating value.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 04, April-2020

yi is an object vector. For every object j, there is an object vector *yj* that is an implicit remarks. Implicit feedback in a roundabout way displays opinion by looking at consumer behavior including purchase history, surfing history, seek patterns, or even mouse movements. Implicit comments commonly denotes the presence or absence of an event

$$\hat{r}_{ui} = \mu + b_u + b_i + q_i^T \left(p_u + |I_u|^{-\frac{1}{2}} \sum_{j \in I_u} y_j \right)$$

Iu — the set of all items rated by user u.
yj— implicit ratings.
For example, there's a film 10 in which a person has just checked the info of the film and spend some time there, which will contribute to implicit rating. Now, since here our records set has now not provided us the details that for how long a person has hung out on the movie, so right here we are considering the fact that despite the fact that a user has rated some film then it means that he has spent some time on that film which contributes to implicit rating. If person u is unknown, then the bias *bu* and the elements *pu* are assumed to be zero. The equal applies for item i with *bi*, *qi*, and *yi*

V. IMPLEMENTATION

A. *Reading and Storing Data*

The dataset I am working with is downloaded from Kaggle .https://www.kaggle.com/Netflix-inc/Netflix-prize-data.

It consists of four .txt files and we have to convert the four .txt files to .csv file. And the .csv file consists of the following attributes.

TABLE I. TOP 5 ROWS OF THE DATA SET

	MovieID	CustID	Ratigs	Date
49557332	17064	510180	2	1999-11-11
46370047	16465	510180	3	1999-11-11
22463125	8357	510180	4	1999-11-11
35237815	14660	510180	2	1999-11-11
21262258	8079	510180	2	1999-11-11

MovieID: Unique identifier for the movie.

CustID: Unique identifier for the Customer.

Ratings - 1 to 5: Rating between 1 to 5.

Date: Date on which customer had watched the movie and given rating.

Once the statistics analysis was completed we have to test for empty values for the records set. By the usage of null characteristic. In Python, especially Pandas, NumPy and Scikit-Learn, we mark missing values as NaN. Values with a NaN cost are overlooked from operations like sum, count, etc. We can mark values as NaN without difficulty with the Pandas Data Frame by means of using the replace () feature on a subset of the columns we are involved in.

Then we need to do away with duplicates, Duplicates are the values which befell extra than once inside the given information. Here we should find the duplicates and dispose of it by way of duplicate characteristic

B. *Performing Exploratory Data Analysis on Data*

In statistics, exploratory data analysis isn't the same as initial data analysis (IDA), which focuses extra narrowly on checking assumptions required for version becoming and hypothesis trying out, and coping with lacking values and making transformations of variables as needed.

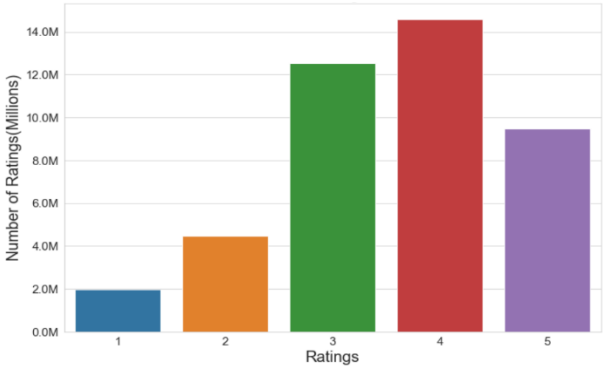

Fig. 1. Distribution of Ratings in data

The above graph shows the distribution of ratings from the data set. For example it implies that there are 2millions of ratings with a rating of 1.And similarly for the reaming ratings also.

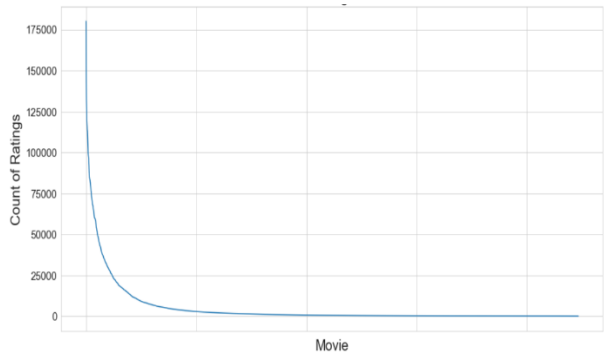

Fig. 2. Analysis of Ratings per movie.

It clearly shows that there are some movies which are very popular and were rated by many users as compared to other movies.

C. *Creating User-Item sparse matrix for the data*

Once the data preprocessing was completed then we have to create a user-Item sparse matrix for the data. Shape of sparse matrix depends on highest value of User ID and highest value of Movie ID.Then we have to find the global average of all movie ratings, average rating per user and average rating per movie. And next we have to compute the similarity matrices, there are mainly two similarity matrices such as user-user and item-item and we have to compute both matrices with our data set.And there is a csv file which consists of movie names for the movie id's which are present in our data set.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 04, April-2020

TABLE I. TOP 5 ROWS OF THE MOVIE TITLE

MovieID	Year_of_Release	Movie_title
1	2003	Dinosaur Planet
2	2004	Isle of Man TT 2004 Review
3	1997	Character
4	1994	Paula Abdul's Get Up & Dance
5	2004	The Rise and Fall of ECW

Let's check does movie-movie similarity works. Pick a random movie and check its top 10 most similar movies.

Suppose pick a movieid with number 17767. The number with particular movieid is picked from the movie titles and will show the name of the movie. Then by using the movie-movie similarity matrix we can find the total number of ratings given to the particular movie and it will also show the similar movies.

For example the movie with movieid 17767 is American experience. The top ten similar movies for American experience are as follows.

TABLE II. TOP 10 SIMILAR MOVIES FOR THE MOVIEID(17767)

Movie ID	Year_of_Release	Movie_Title
9044	2002	Fidel
7707	2000	Cuba Feliz
15352	2002	Fidel: The Castro Project
6906	2004	The History Channel Presents: The War of 1812
16407	2003	Russia: Land of the Tsars
5168	2003	Lawrence of Arabia: The Battle for the Arab World
7100	2005	Auschwitz: Inside the Nazi State
7522	2003	Pornografia
7663	1985	Ken Burns' America: Huey Long
17757	2002	Ulysses S. Grant: Warrior / President: America...

D. Applying Machine Learning Models

Before us applying the models we have to featurize data for the regression problem. Once it was completed we have transform data to surprise models. We can't give raw data (movie, user, and rating) to train the model in Surprise library. Following are the models which we are applying for the data.

1) XGBoost was the first model which we are applying for the featurize data. When we run the model we get the RMSE and MAPE for the train and test data.

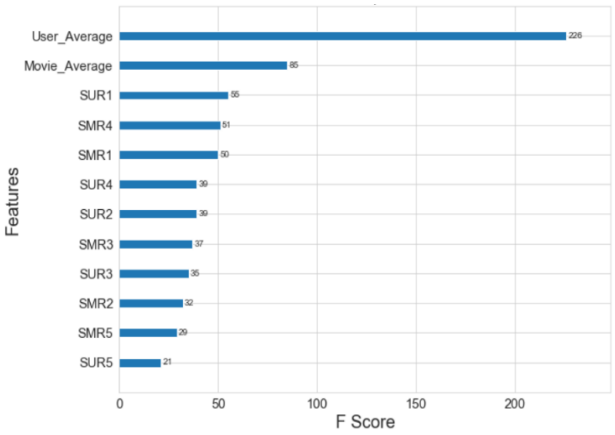

Fig.1. Feature importance of xgboost model.

Here user average and movie average are most important features. Here the RMSE and MAPE are two error metrics which we are used for measuring error rate.

TABLE I. ERROR RATES OF XGBOOST MODEL

	TRAIN DATA	TEST DATA
RMSE	0.8105686945196948	1.0722769984483742
MAPE	24.1616427898407	33.160274170446975

2) Surprise Baselineonly was the next model we are using. Here we are updating the train and test data with the extra feature baseline only. When we run baseline model we get the following as the output.

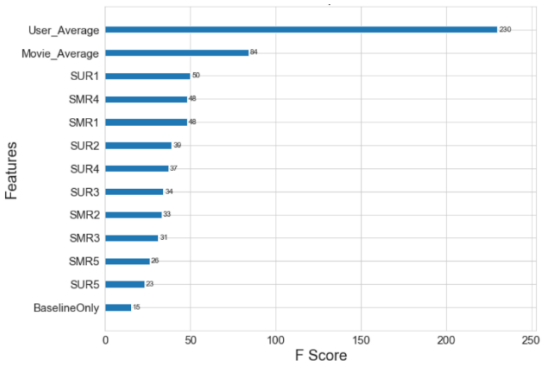

Fig. 2. Feature importance of baseline only model

From the above graph we can say that user average and movie average are most important features while baselineonly is the least important feature. And the error rates for the baseline model is as follows.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 04, April-2020

TABLE II. ERROR RATES OF BASELINEONLY MODEL.

	TRAIN DATA	TEST DATA
RMSE	0.8102119017805783	1.0688807299545566
MAPE	24.166917800903332	33.334272483120664

3) Surprise KNNBaseline was the next model we are applying for the data set. Here we have to update our data set with the features from the previous model. When we run the model we get the following as the output.

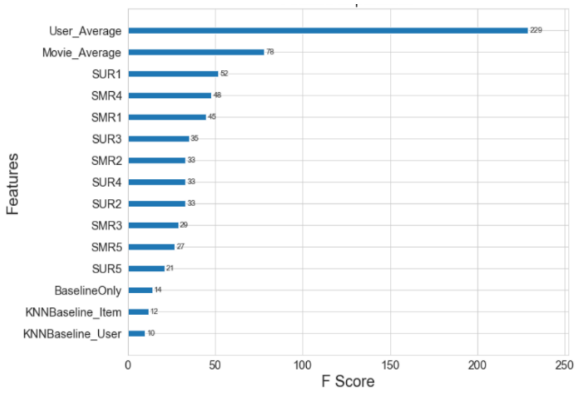

Fig. 3. Feature importance of surprise knnbaseline model.

From the above graph we can say that user average and movie average are most important features while baseline_user is the least important feature. And the error rates for the baseline model is as follows.

TABLE III. ERROR RATES OF SURPRISE KNNBASELINE MODEL

	TRAIN DATA	TEST DATA
RMSE	0.810123471320971	1.0717424411624028
MAPE	24.16132688522339	33.18525885602669

4) Matrix Factorization SVD was the next model we are using. And here we have to update our data set each time. And when we run the matrix factorization svd we get the following as the output. And the error rates for the model is as follows.

TABLE IV. ERROR RATES OF MATRIXFACTORIZATION SVD MODEL

	TRAIN DATA	TEST DATA
RMSE	0.8915292018008784	1.0676633276455576
MAPE	27.929401130209502	33.39843901285594

5) Matrix Factorization SVDpp was the final model we are applying for the data set. And here we have to update the data set with the features with from the previous models.And the error rates for the model is as follows.

TABLE V. ERROR RATES OF MATRIX FACOTRIZATION SVDpp MODEL

	TRAIN DATA	TEST DATA
RMSE	0.7871581815662804	1.0675020897465601
MAPE	24.06204006168546	33.39327837052172

VI. RESULT ANALYSIS

Comparison of all the models.

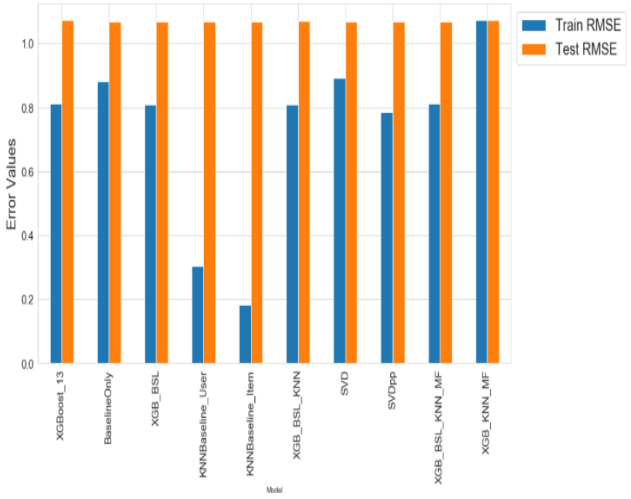

Figure 1. Train and Test RMSE and MAPE of all Models.

The above graph will show the comparision of all model with error values.

TABLE I. SUMMARY OF ALL THE MODELS WITH TRAIN AND TEST RMSE VALUE

S.NO	MODEL	TRAIN RMSE	TEST RMSE
0	XGBOOST	0.810569	1.07228
1	BASELINEONLY	0.881143	1.06784
2	XGB_BSL	0.810212	1.06888
3	KNNBASELINE_USER	0.304498	1.06765
4	KNNBASELINE_ITEM	0.181651	1.06765
5	XGB_BSL_KNN	0.810123	1.07174
6	SVD	0.891529	1.06766
7	SVDPP	0.787158	1.0675
8	XGB_BSL_KNN_MF	0.810568	1.0687
9	XGB_KNN_MF	1.07269	1.07276

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 04, April-2020

VII. CONCLUSION

So, far our best model is SVDpp with Test RMSE of 1.0675.Here we are not much worried about our RMSE because we haven't trained it on the whole data . Our main intention here is to learn more about Recommendation Systems .If we taken whole data we would definitely get better RMSE .

VIII. FUTURE ENHANCEMENT

Tune hyper parameters of all the Xgboost models above to improve the RMSE. Here we used 10K users and 1K movies to train the above models due to my pc ram issues. In the future, I am going to run on the entire information set using cloud resources.

REFERENCES

[1] Davidsson C, Moritz S. Utilizing implicit feedback and context to recommend mobile applications from first use. DOI: 10.1051/,04008 (2017) 712012ITA 2017 ITM Web of Conferences itmconf/201 40084 In: Proc. of the Ca RR 2011. New York: ACM Press, 2011. 19-22.http://dl.acm.org/citation.cfm?id=1961639[doi:10.1145/1961634.1961

[2] Bilge, A., Kaleli, C., Yakut, I., Gunes, I., Polat, H.: A survey of privacy-preserving collaborative filtering schemes. Int. J. Softw. Eng. Knowl. Eng. 23(08), 1085–1108 (2013)CrossRefGoogle Scholar.

[3] Calandrino, J.A., Kilzer, A., Narayanan, A., Felten, E.W., Shmatikov, V.: You might also like: privacy risks of collaborative filtering. In: Proceedings of the IEEE Symposium on Security and Privacy, pp. 231–246, Oakland, CA.

[4] Okkalioglu, M., Koc, M., Polat, H.: On the discovery of fake binary ratings. In: Proceedings of the 30th Annual ACM Symposium on Applied Computing, SAC 2015, pp. 901–907. ACM, USA (2015).

[5] Kaleli, C., Polat, H.: Privacy–preserving naïve bayesian classifier based recommendations on distributed data. Comput. Intell. 31(1), 47–68(2015).

[6] Munoz-Organero, Mario, Gustavo A. Ramíez-González, Pedro J. Munoz-Merino, and Carlos Delgado Kloos. "A Collaborative Recommender System Based on Space-Time Similarities", IEEE Pervasive Computing, 2010.

[7] Peng, Xiao, Shao Liangshan, and Li Xiuran. "Improved Collaborative Filtering Algorithm in the Research and Application of Personalized Movie.