Multi-Level Ensemble Learning based Recommender System

Bhavya Ghai[1], Joydip Dhar[2] and Anupam Shukla[3]
ABV- Indian Institute of Information Technology and Management, Gwalior, India
[1]bhavyaghai@gmail.com, [2]jdhar@iiitm.ac.in,[3]anupamshukla@iiitm.ac.in

Abstract . In this paper, we have tried to go beyond conventional ensemble learning & explore multi-level ensemble learning with reference to recommender systems. In particular, we have focused on stacked generalization for building Movie Recommender System. We have tried to analyze the transition from single level to multi-level ensemble learning and its effects on the overall accuracy. We have used movielens dataset from Grouplens Project and used a host of techniques like Collaborative Filtering, PAM, Content based recommender, Random Forest, SVM, ANN, etc. to optimize accuracy. We have experimented with various combinations of base learners based on their accuracy & diversity to finally arrive at the most accurate ensemble of ensembles. Results show that 2-level stacking gives more accurate results than single level stacking or any individual recommender system.

Keywords: Ensemble Learning, Movielens, Stacking, Bagging, Recommender System

1 Introduction

Ensemble learning is a very powerful machine learning paradigm which can optimize roughly any other learning algorithm. It is based on a very intuitive concept that a group of people can make a better decision than an individual. Instead of solely relying on the smartest learning algorithm, it exploits the wisdom of all learning algorithms and focusses on collective intelligence. Earlier, we used to train a number of different models on the dataset and choose the best one for deployment. Using Ensemble learning, we can use all the models we have trained (provided they have accuracy > 50%) to further optimize our accuracy. It can be used to deal with complex problems such as Classification, Regression, Time Series, Recommender Systems, Reinforcement Learning, Unsupervised Learning, etc. The down side of using ensemble learning is that the complexity of the model increases so it might not be applicable for real-time analysis.

Recommender Systems are supervised learning algorithms which try to model a user based on a host of parameters and recommend accordingly. They impact our everyday life. Most common examples include Amazon, Netflix, Facebook, etc. In this age of Internet, It becomes a herculean task for a user to find the most relevant information on the web. Here, Recommender systems step in & filter through a sea of data to fetch the most relevant data according to our profile. They save our time & enable us to consume the most relevant data which we might have otherwise missed. Recommender systems is currently a hot research topic with research focusing on Personalization, Context-Aware Recommendations, Real-Time Recommendations, etc. Earlier, Recommender Systems were restricted to E-Commerece, Videos, etc. but

now they are being used for Recommending News, Friends, Jobs, restaurants, financial services, life insurance, etc. Recommendations can be based on current location, likings, ratings, content, age, sex and a whole lot of other parameters based on online interaction.

The realm of Recommender Systems using ensemble approach has not been exploited much. Most of the work deals with single level ensemble learning. In this paper, we have tried to extend conventional ensemble learning and used multi-level stacking. We have proposed a novel and more accurate recommender system based on traditional learners using 2-level stacking generalization.

This paper is organized as follows. Section 2 discusses about the related work in this field. Section 3 throws light on ensemble learning in some detail. Section 4 discusses about the recommender systems we have used in this work. Section 5 describes our proposed approach towards multi-level ensemble learning model. Section 7, 8 and 9 contains Results, Conclusion and Future Work respectively.

2 Related Work

Earlier, we used to train multiple models and used the best one for deployment. This way, our time and efforts training other models go waste[15]. Then we came across ensemble learning which is a very powerful technique to optimize other learning algorithms [7] and widely used in data competitions.[9] Ensemble learning has many flavors. It has various architectures[1] and techniques like bagging, boosting, stacking, etc. We explored that Ensemble learning has been used with Recommender systems [2,4,5] but Multi-level ensemble learning has not been exploited much, especially with reference to Recommender Systems[3]. We used a number of models like collaborative filtering, Content filtering, PAM [11,12,13,14], etc. for building multi-level ensemble learning system.

3. Ensemble Learning

Ensemble learning is an optimization paradigm where numerous learners are trained to solve the same problem [23,25,27]. Ordinary machine learning approaches work on a particular hypothesis whereas ensemble learning work on a set of hypotheses and fuse their results together to achieve better accuracy. Ensemble learning refers to a collection of methods that learn a target function by training a number of individual learners and combining their predictions.[11] Ensemble learning is used when we can build component learners that are more accurate than chance and, more importantly, that are independent from each other. They work on the principle that uncorrelated errors of individual learners can be eliminated through averaging. An ensemble is itself a supervised learning algorithm, because it can be trained and then used to make predictions.[22]
Ensemble learning has following Applications :-

- Series Prediction

- Regression
- Recommendation Systems
- Classification
- Reinforcement Learning
- Unsupervised Learning
- Risk Prediction

4. Techniques Used

4.1 UBCF : UBCF stands for User Based Collaborative Filtering. It is based on the principle that user who liked similar things in the past will have like similar things in future. There are mainly two types of recommender systems, as a function of the algorithm used: Content-Based Filtering (CBF) and Collaborative Filtering (CF). CF is one of the most commonly used methods in personalized recommendation systems. Collaborative filtering algorithm recommend items based upon opinions of people with similar tastes. Collaborative filtering can also recommend items that are not similar and like-minded users have rated the items. Collaborative filtering faced some problems by traditional information filtering duly eliminating the need for computers to understand the content of the items. Recommender systems need to store certain information about the user preferences, known as the user profile to achieve this personalization. The system will inform the user of what items are well recommended by other users with similar likes or interests. An analysis of the content by the system is not necessary and the quality or subjective evaluation of the items will be considered. However, these algorithms present problems in their computational performance and efficiency.

Algorithm :-
Step I : Split dataset via 80/20 rule
Step II : Evaluate item-user matrix
Step III : Evaluate user-user similarity matrix
Step IV : Define Neighborhood
Step V : Predict Values
Step VI : Testing

4.2 IBCF : Item-based collaborative filtering is a model-based algorithm for making recommendations. In the algorithm, the similarities between different items in the dataset are calculated by using one of a number of similarity measures, and then these similarity values are used to predict ratings for user-item pairs not present in the dataset. Looks into the set of items the target user has rated and computes how similar they are to the target item and then selects k most similar items. Prediction is computed by taking a weighted average on the target user's ratings on the most similar items.

4.3 PAM – PAM or Partitioning around Medoids is a clustering algorithm similar to k-NN algorithm. The number of clusters(k) is given as input. PAM breaks the dataset into k groups or clusters with each cluster having a representative or medoid in this case. Each point is assigned to a cluster which is closest to it based on distance function. Distance function may be eucledian, manhattan, etc. After assigning each point, median of points in each cluster is chosen as new cluster representative. This process of assigning points to clusters is repeated until convergence is reached For more details please refer [11]. PAM can be directly used in R using 'cluster ' package [12,13].

In this paper, we used PAM as a demographics based recommender system. It clusters users based on their demographic data. The underlying assumption here is that people belonging to similar age group, location, profession, etc. will have similar tastes and user demographics are constant over time. The prediction by a given user for a particular movie will be the mean of ratings given by other users who belong to the same cluster as the user.

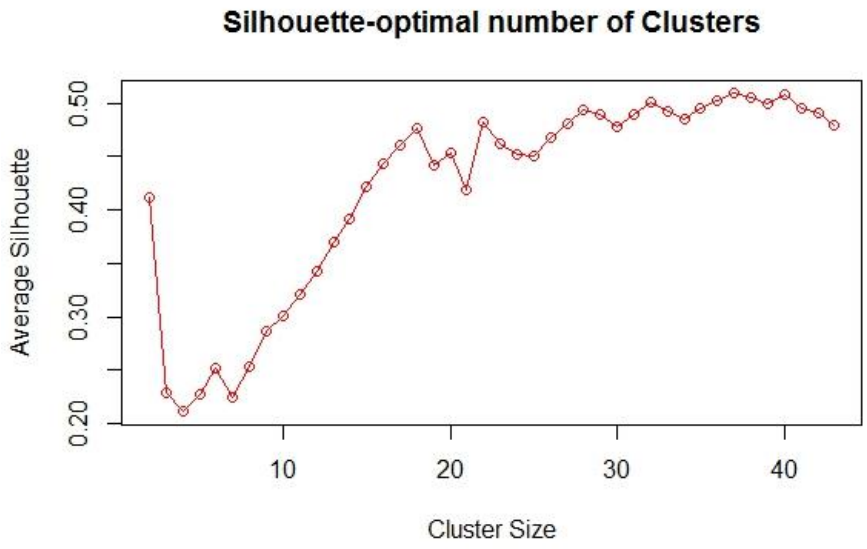


Fig. : A plot between Average Silhouette and Cluster size

We provide user demographic data containing age, sex, zipcode, profession of 943 unique users as input. Since the input data is mixed i.e. it contains numeric (age, zipcode) and categorical (sex, profession) data, we have used gower distance as the metric. Now, we need to choose optimal value of k i.e. number of clusters. We will use a measure known as silhouette to determine k. Silhouette is a measure which determines how well an observation fits into a cluster. It ranges between 0 and 1 with 1 representing best fit. We will train PAM multiple times and calculate average silhouette for different cluster size. Plotting average silhouette vs cluster size, we find that the graph peaks at k = 37 which is the optimal cluster size in our case. For more details about silhouette, please refer [14].

5. Proposed Approach

As discussed above, Ensemble learning includes many models like Bagging, Boosting, Stacking, etc. In this paper, we will primarily focus on stacking. For creating an effective ensemble model, we need to train multiple diverse learners. In our case we chose User based Collaborative filtering, Item based collaborative filtering, PAM & POP. Each of these learners manipulate the dataset in a unique way & make errors according to their individual biases. We trained these models on a portion of dataset & evaluated their accuracy & diversity on the rest. Initially, we split the entire dataset into training & test dataset in the ratio 80:20. Then we trained all the individual learners namely UBCF, IBCF, POP, Content & PAM on train dataset and made predictions on the test dataset. We measured accuracy via RMSE value & diversity via Pearson correlation coefficient. Higher the correlation coefficient, lower is the diversity. For creating 1-level stacking model, we fused all the predictions using many popular machine learning techniques as shown in Table 3.

The predictions made by the individual learners on the test set served as the input for 1st level stacking. After training different 1st level stacking models, we tested it on a portion of the initial test dataset. Here, we expect that 1-level stacking will give better accuracy than the best individual learner considering each learner has high accuracy & diverse in its approach.

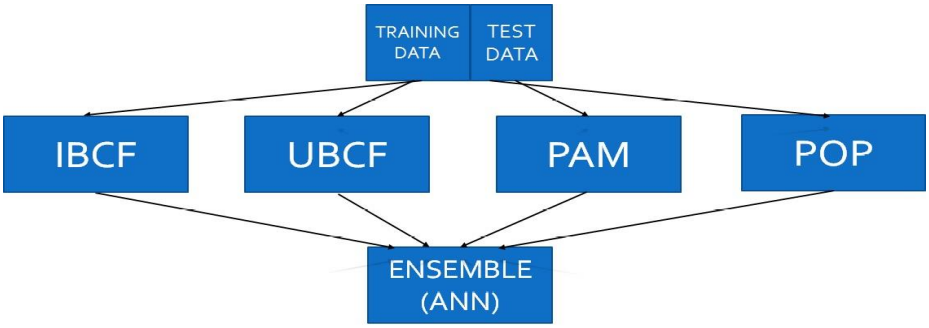


Fig. 1-level Stacking Architecture

On the same grounds, we tried to extend 1-level stacking to 2-level stacking. Our objective here is not to create the most accurate recommender system but to investigate whether subsequent levels of stacking will enhance accuracy. For creating 2-level stacking model, we'll repeat the same process but will take input from 1st level instead of base learners. The term 'best' here means high accuracy & diversity. For first level stacking, we choose the best learners from the ground level based on their accuracy & diversity. After choosing best learners, we can employ various stacking techniques to fuse the predictions. In our case, we used Random Forests, Linear Regression, ANN & State Vector Machines for stacking at 1st level. For second level stacking, we will repeat the process and this time we will choose best learners from 1st level learners instead of ground level learners. We may choose a single model or a combination of models implemented at 1st level stacking for building second level

model based on accuracy & diversity. We have again used many popular ML algorithms for 2nd level stacking. Genetic Algorithms, Fuzzy logic, etc. can also be used for stacking. We are trying to test how different levels of stacking compare to each other.

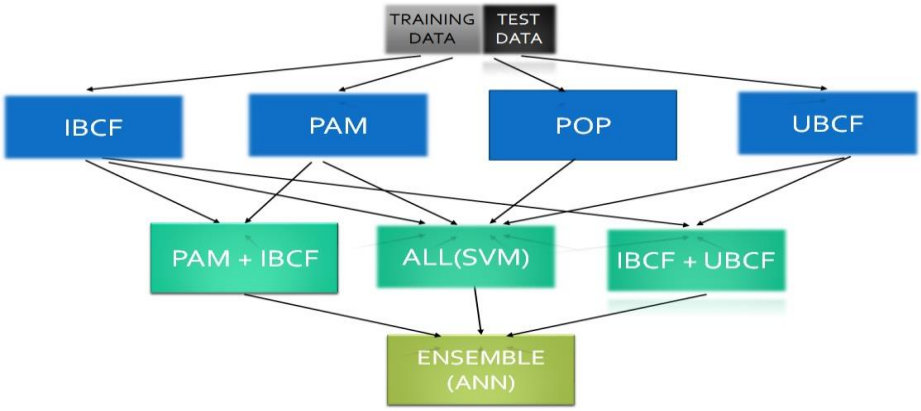


Fig. 2-level Stacking

Algorithm

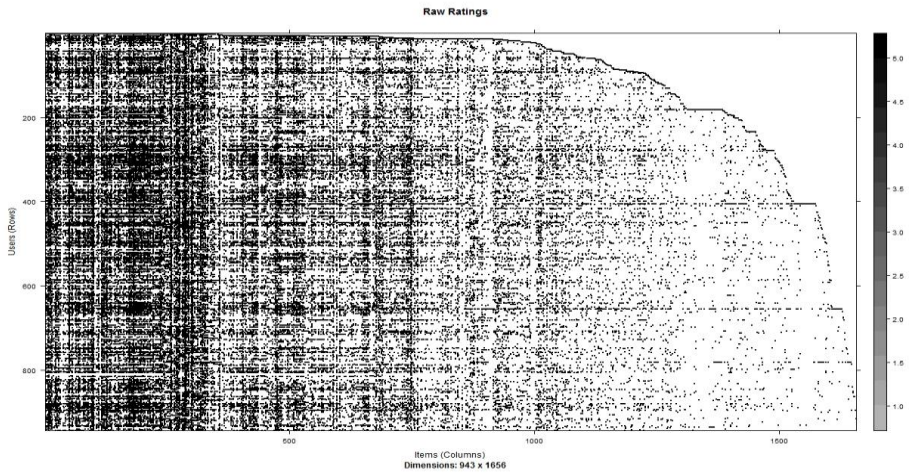
Step 1: Split dataset into train & test dataset
Step 2: Train many diverse learners with accuracy >50%
Step 3: Pick best learners or combination of best learners
Step 4: Implement stacking models on learners chosen in Step 3
Step 5: Pick best learners/ combination of learners from stacking models (Step 4)
Step 6: Implement stacking model on learners from Step 5

For creating n-level stacking model, the value of n can be decided on the basis of diversity between learners. After each level we must calculate the correlation coefficient between learners at that level. If the correlation between all learners is greater than a threshold value (say 0.80), then we must stop. In this paper, we have used n=2.

6. Dataset

We have used *Movielens 100k* dataset from Grouplens project for training & testing our ensemble learning model. This dataset was collected via Movielens web site at University of Minnesota from 19th Sept'97 till 22nd April'98. The dataset contains no missing values and has been used extensively for research purposes. The dataset broadly consists data about Movies, User demographics and Ratings. The user file consists of attributes such as User-Id, Age, Sex, Occupation & Zipcode of 943 unique users. The movie file consists attributes like Movie-Id, Movie title, Release date, Imdb URL & various genres for 1682 movies. The ratings table consists of 100k movie ratings by 943 users for 1682 movies. It has four columns namely User-Id,

Movie-Id, rating, Timestamp. User-Id may vary from 1 to 943. Similarly, Movie Id may vary from 1 to 1682. Movie rating range from 1 to 5. The dataset can be freely accessed & downloaded at [8].



7. Results

All the models have been trained & tested using R and its robust set of packages on Movielens-100k dataset. The output of a recommender system depends on the purpose it was built for. Recommender systems may be used for prediction, classification or ranking. Accordingly, the evaluation metric may vary from RMSE, MAE, Precision, Recall to Tau, rho, etc. We are primarily interested in prediction as classification & ranking can be performed on the basis of predicted values. Hence, we will be using RMSE, MAE & MSE to evaluate recommender models.

Firstly, we trained base learners i.e UBCF, IBCF, PAM, POP, Content, etc. We have discussed these recommenders in section 4. We have trained them on 80% of data. Some of these recommenders can be directly used via Recommenderlab package in R [10]. Table 1 summarizes the accuracy for base learners.

Table 1: Base Learners Accuracy

Recommenders	**MAE**	**MSE**	**RMSE**
UBCF	0.7999	1.0191	1.0095
PAM	0.8954	1.3237	1.1505
IBCF	0.7475	0.9359	0.9674
POP	0.7535	0.9217	**0.9600**
Content	0.9237	1.4151	1.1896

Random	1.7010	4.3604	2.0881

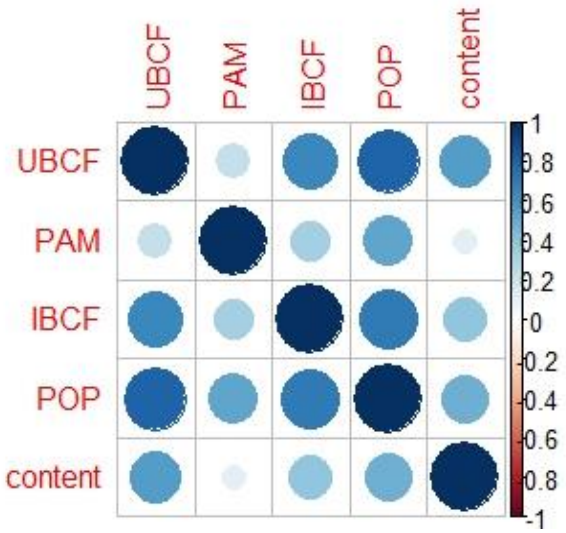


Fig. : Correlation between base learners

Based on the predictions made by base learners on 20% of data, we calculated the correlation between base learners as specified in table 2 and graphically represented in the above figure. In the graphical representation, the bigger & darker a circle is, more is the correlation and hence less diversity.

In table 1, we evaluated all base learners. To put things into perspective, we have also used random. From table 1, we can say that POP and IBCF are most accurate. Surprisingly, POP has outperformed traditional collaborative filtering algorithms. Table 2 contains the correlation coefficients between different base learners. As far as diversity is concerned, PAM seems to be walking away from the crowd. Having done with diversity and accuracy of base learners, we will move to next level.

Table 2 : Correlation between Recommenders at level 1

Correlation	UBCF	PAM	IBCF	POP	Content
UBCF	1	0.2309	0.6443	0.8018	0.5554
PAM	0.2309	1	0.3378	0.5215	0.1175
IBCF	0.6443	0.3378	1	0.7047	0.3967
POP	0.8018	0.5215	0.7047	1	0.4861
Content	0.5554	0.1175	0.3967	0.4861	1

For building 1-level Ensemble learning structure as described in Fig. 1, we will stack all the base learners together using machine learning techniques like Random Forest, SVM, Linear regression & Neural network. We have trained these models on 80% of the predicted data obtained from base learners. Table 3 shows the evaluation of these techniques on the remaining 20% data.

Table 3: Evaluation of 1st level stacking techniques

Model	MAE	MSE	RMSE
Random Forest	0.7442	0.8838	0.9401
SVM	0.7272	0.8732	0.9344
Mean	0.8022	1.031	1.015
Linear Regression	0.7337	0.8652	**0.9301**
Neural Network	0.7898	0.9818	0.9908

Using Table 3, we can say that Linear Regression outperformed other machine learning techniques. Stacked Linear Regression model has performed better than the best individual base learner. The RMSE has dropped from 0.9600(POP) to 0.9301(Linear Regression) which is 3.12% increment in accuracy.

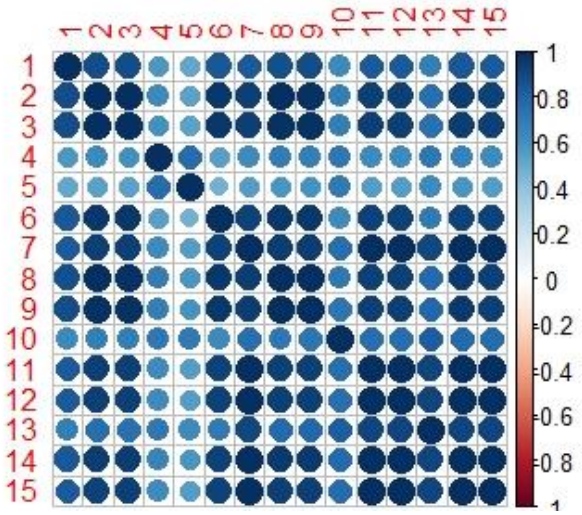
For building 2-level Ensemble learning model, we will first train different combination of base learners to form 1st layer of stacking models. Then we'll build 2nd layer by repeating the same using 1st layer as input. We will choose different combinations based on the accuracy & diversity of base learners. Table 4 summarizes the evaluation of different combinations of base learners along with the model used. Here RF means Random Forest, SVM means State Vector Machines, LR means Linear Regression and NN represent Artificial Neural network.

Table 4: Evaluation of Stacking models for 2-level Ensemble Architecture

Index	Recommenders	Model	MAE	MSE	RMSE
1	IBCF + PAM	RF	0.7847	0.9747	0.9872
2	IBCF + PAM	LR	0.7523	0.9076	0.9527
3	IBCF + PAM	SVM	0.7477	0.9172	0.9577
4	Content + PAM	LR	0.8301	1.0750	1.0368
5	Content + PAM	RF	0.8519	1.1366	1.066
6	IBCF + Content	LR	0.7619	0.9311	0.9649
7	IBCF + POP	SVM	0.7335	0.8806	0.9384
8	IBCF+PAM+ Content	LR	0.7362	0.8701	0.9328

9	IBCF+PAM+ Content	NN	0.7471	0.8949	0.9460
10	UBCF+PAM+ Content	RF	0.7975	1.0121	1.0060
11	IBCF+POP+UBCF	LR	0.7364	0.8700	0.9327
12	POP+PAM+ Content	LR	0.7364	0.8700	0.9327
13	UBCF+POP+PAM+ Content	LR	0.7574	0.9235	0.9610
14	IBCF+POP+PAM+ Content	LR	0.7341	0.8654	**0.9302**
15	IBCF+POP	LR	0.7362	0.8701	0.9328

Table 4 summarizes 15 different models trained for 1st level stacking. Clearly, Model indexed 14 performed the based. Each of these models is trained only on a subset of entire data. For example, IBCF+POP, represents the prediction data obtained from IBCF & POP when applied on test dataset. Next we'll calculate the correlation between these 15 models. Fig *** represents correlation coefficient between all 1st level stacking models. Bigger & darker circles on the figure represents stronger correlation and hence less variance.



Finally, we are ready for the 2nd level stacking for the 2-level architecture. We will choose different combination of models from 1st level stacking models based on accuracy & diversity. Table 5 represents the different combination of stacking models implemented for 2nd level stacking. Numbers mentioned in Input column of Table 5 represents the index of model as mentioned in Table 4. For example, 7+8 represents the combination IBCF + POP(SVM) and IBCF+PAM+ Content(LR).

Table 5 : Evaluation of 2nd level Stacking Models

Input	Model	MAE	MSE	RMSE
3+5+6+13+14	LR	0.7164	0.8404	0.9167
7+8	LR	0.7185	0.8442	0.9188
4+5+7+10	SVM	0.7143	0.8404	0.921
14+15	SVM	0.707	0.8379	**0.9154**

The combination of model indexed 14 & 15 is the most accurate model among all 2-level stacking models. We observe that 2nd level stacking has significantly improved accuracy from 1st level stacking. The RMSE has dropped from 0.93 to 0.9154 which is equivalent to 1.57% increment in accuracy.

8. Conclusion

We implemented many recommender systems & experimented with different ensemble combinations. Our objective here was to investigate the outcomes of multi level ensemble learning with respect to recommender systems. Although, similar results can be expected for other supervised learning algorithms as well. Here, we have used movie recommendation system as a case for experimenting with multi-level ensemble learning. We tried to solve the complex problem of recommendation by splitting into sub-problems and merging via ensemble learning. Each of the base leaner is solving a sub-problem. For example, IBCF, UBCF are recommending on the basis of ratings data, PAM is recommending on the basis of demographics, Content based recommender is based on genre data, etc.

We found that Diversity and Accuracy of base learners together determine the effectiveness of Ensemble learning models. More the accuracy & diversity of base learners, better is the overall accuracy optimization of the model.
Using 2-level ensemble learning, we were able to reduce RMSE from 0.9601 to 0.9154 which is equivalent to 4.65% increment in performance. We also found that moving from 1-level architecture to 2-level architecture is rewarding. There was a 1.56% enhancement in accuracy over 1-level architecture. As we move from the individual base learners to higher levels of stacking, Accuracy increases and diversity between the models at the same level decreases rapidly. Optimum number of levels will be that smallest level at which the correlation between learners becomes greater than some pre-decided threshold value. If we go higher than optimum value, then it will increase the computational complexity of the model without much appreciation in accuracy.

We have used the same base learners for 1-level and 2-level ensembling. The question arises that why 2-level stacking is giving better results than 1-level stacking ? We think that it maybe because 1-level ensemble model is not able to capture the diversity of the dataset substantially. In case of multi-level stacking, we use numerous models at intermediate levels which are based on subsets of intermediate results. The variance caused by using different models and using different subsets of data

iteratively has accounted for better accuracy for multi-level ensemble learning model. There is a trade-off between accuracy & diversity. With subsequent levels, we are able to exploit the diversity of dataset to a greater extent leading to increment in accuracy.

9. Future Work

One of the biggest disadvantage of Ensemble learning models is that they are computationally expensive and so are unfit for Real time Analysis. Future work may include faster Ensemble models which might be used for Real time Analysis. We can also investigate the flexibility of Ensemble learning models by operating the same model on different datasets and checking its effectiveness. Ensemble learning has many architectures. Future work may including experimenting with different Architectures or forming a new hybrid architecture. Ensemble model can also be optimized further by including more number of diverse learners at each level and include other techniques like bagging & boosting.

References

[1] Asmita, Shruti, and K. K. Shukla. "Review on the Architecture, Algorithm and Fusion Strategies in Ensemble Learning." *International Journal of Computer Applications* 108.8 (2014).

[2] Jahrer, Michael, Andreas Töscher, and Robert Legenstein. "Combining predictions for accurate recommender systems." *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2010.

[3] Amatriain, Xavier, et al. "Data mining methods for recommender systems."*Recommender Systems Handbook*. Springer US, 2011. 39-71.

[4] Bar, Ariel, et al. "Improving simple collaborative filtering models using ensemble methods." *Multiple Classifier Systems*. Springer Berlin Heidelberg, 2013. 1-12.

[5] Lili, Cheng. "RECOMMENDER ALGORITHMS BASED ON BOOSTING ENSEMBLE LEARNING." *International Journal on Smart Sensing & Intelligent Systems* 8.1 (2015).

[6] Webb, Geoffrey I., and Zijian Zheng. "Multistrategy ensemble learning: Reducing error by combining ensemble learning techniques." *Knowledge and Data Engineering, IEEE Transactions on* 16.8 (2004): 980-991.

[7] P´adraig Cunningham , Technical Report UCD-CSI-2007-5. Ensemble Techniques.

[8] http://grouplens.org/datasets/movielens/100k/

[9] http://mlwave.com/kaggle-ensembling-guide/

[10] Hahsler, Michael. "recommenderlab: A Framework for Developing and Testing Recommendation Algorithms." *Nov* (2011).

[11]https://en.wikibooks.org/wiki/Data_Mining_Algorithms_In_R/Clustering/Partitioning_Around_Medoids_(PAM)

[12] https://stat.ethz.ch/R-manual/R-devel/library/cluster/html/pam.html

[13] https://cran.r-project.org/web/packages/cluster/cluster.pdf

[14] http://www.stat.berkeley.edu/~s133/Cluster2a.html

[15] David, Jeff, Samir Bajaj, and Cherif Jazra. "A Facebook Profile-Based TV Recommender System." vectors 1: u2.