Developing Hybrid-Based Recommender System with Naïve Bayes Optimization to Increase Prediction Efficiency

Ndung'u Rachael Njeri, Kamau Gabriel Ndung'u*, Wambugu Geoffrey Mariga

Department of Information Technology
Murang'a University of Technology
Murang'a, Kenya
Email: cndungu [AT] mut.ac.ke

Abstract—**Commerce and entertainment world today have shifted to the digital platforms where customer preferences are suggested by recommender systems. Recommendations have been made using a variety of methods such as content-based, collaborative filtering-based or their hybrids. Collaborative systems are common recommenders, which use similar users' preferences. They however have issues such as data sparsity, cold start problem and lack of scalability. When a small percentage of users express their preferences, data becomes highly sparse, thus affecting quality of recommendations. New users or items with no preferences, forms cold start issues affecting recommendations. High amount of sparse data affects how the user-item matrices are formed thus affecting the overall recommendation results. How to handle data input in the recommender engine while reducing data sparsity and increase its potential to scale up is proposed. This paper proposed development of hybrid model with data optimization using a Naïve Bayes classifier, with an aim of reducing data sparsity problem and a blend of collaborative filtering model and association rule mining-based ensembles, for recommending items with an aim of improving their predictions. Machine learning using python on Jupyter notebook was used to develop the hybrid. The models were tested using MovieLens 100k and 1M datasets. We demonstrate the final recommendations of the hybrid having new top ten highly rated movies with 68% approved recommendations. We confirm new items suggested to the active user(s) while less sparse data was input and an improved scaling up of collaborative filtering model, thus improving model efficacy and better predictions.**

Keywords-Collaborative filtering, Association rule mining, hybrid ensemble, recommender systems

I. INTRODUCTION

Today's world is marked by precipitous advancement in information technology and the consequent advancement in social network applications and electronic commerce. With the advent of analytics, the need to make recommendations to users of various platforms aimed at maintaining interests and easing navigation arose as the platforms arose. Recommender systems have become a crucial part of many internet spaces in the modern world. The main aim of recommenders is on making suggestions on items that would be of interest to the user(s) and make search of such items easier and better. The suggestion(s) would be made based on active user's preferences, what similar users have preferred and what is frequently requested by the user. Application of recommenders are on many online marketplaces like in entertainment media, recommendation systems are used to make entertainment suggestions based on user's preferences, while in marketing, recommenders are crucial in making personalized suggestions of products and services to the application users.

Recommender systems are categorized depending on the input data for the algorithms. They are divided into two major categories, the Collaborative Filtering Recommendation Systems (CFRS) and the Content-Based Recommendation Systems (CBRS). Their major difference being the data they consume, in that CFRS uses usage data while the CBRS uses the metadata and user profile data to make recommendations. In this paper, our focus was on CFRS reducing data sparsity and cold start problem and at the same time improving data scalability while forming user-item matrix.

Collaborative Filtering Recommender Systems (CFRS) are based on users who had similar preferences in the past and are likely to have similar preferences in future. The users who interact with items on a similar manner share similar preferences and users with shared preferences are likely to respond in the same way to similar items. Thus, recommendations given by CFRS uses choices made by similar users to make suggestions. Lack of users expressing their preferences affects how the CFRS recommends due to sparse data. New users who have not expressed their preferences also affects recommendations through cold start problem. The results of algorithms that learn the past interactions of user-item matrices for predictions, were influenced by the fitted data, thus affected the predictions accuracy [1]. Having substantial reliable data for training would go a long way to improve the predictions. Lack of scalability during user-item matrices formation is also a key weakness of CFRS, which uses a lot of computation resources. This work proposed classification using naïve bayes and subsequent choice of positive class labelled

data input in the CFRS as a method to decrease input of sparse data, which suppressed user-item matrix formation. Efficiency in the recommender systems was thought of as reduction of time used to produce the recommendations over the quality of suggestions output.

Development of a hybrid model was aimed to mitigate the lone weaknesses of the CFRS. Hybrid systems mostly gain from positive synergies of the ensemble algorithms used. This project aimed to use Association Rule Mining (ARM) to discover other latent factors or patterns from other features besides the user preference-based response variable- the rating. The ARM model input was the low rated /unrated data, which using other features like the genres, was used to discover other existing relationships among user-items association. The discovered data patterns would be used to formulate the final recommendations list, while taken together with those recommendations produced by the CFRS.

Development of the hybrid was through python programming on Jupyter notebook™, using machine learning algorithms. MovieLens 100k and 1M datasets were used for optimization using Naïve Bayes (NB) classifier. The NB classifier binarized the data into two class labels. For CFRS model, matrix factorization using Singular Value Decomposition (SVD) was used, while for ARM, Apriori algorithm that uses market basket analysis to discover patterns from frequent itemsets.

Proposed hybrid model was a collection of various Machine Learning (ML) algorithms taking different input data at different stages depending on the ML algorithm been used. The NB classifier input was the cleaned data, with selected predictors, while the CFRS model input was the positive labelled data. The ARM data input were the negative labelled data. The outputs of the CFRS and ARM models were taken through another ensemble formulation to have the final list of top recommended items, as shown in Figure 2, section III.

The rest of the paper is organized as follows: Section II explains the related work, Section III describes methodology while Section IV describes the proposed model. Section V gives insights of results and discussions, while the paper concludes at Section VI.

II. Related Work

This section describes related research work based on the current study. We explored the naïve bayes classifier, the collaborative filtering and the association rule mining algorithms.

A. Naïve Bayes Classifier

Naïve Bayes classifier is a group of classification machine learning algorithms, which were generative in nature that they use a joint probability of the inputs (x) and the label (y) to make predictions. Naïve Bayes classifier uses Bayesian theorem where they take every feature as a random and independent variable to calculate prior [p (y | x)], and then pick the most likely label y [2]. The Bayesian theorem states that

$$\rho(X|Y) = \{\rho(Y)|\rho(X)\}/\rho(Y) \quad (1)$$

Of the form

Posterior = (Prior * Likelihood)/ Evidence

Where y was the response variable and x were the input attribute, while using the algorithm in a probability distribution of the variables in the dataset and predicting the probability distribution of the response variable.

NB classifier is used when there is prior knowledge of the model being developed and applicable when the model is not using massive amounts of data for training since its supervised learning. It's mostly used because it's a fast predicting and easy to use algorithm, which performs well in a multiclass and in real-time predictions. It's also easy to evaluate and interpret its results. In datasets where independence of features holds, Naïve Bayes classifier performs well and requires less training data as compared to other models such as logistic regression. It works with both discrete and continuous data and it scales well. It's commonly used because of its probabilistic and simplistic nature of its implementation. The major challenge of this algorithm is the assumption of independence of features in the dataset. This is independence is brought about by its 'naïve' nature, which assumes that features are measured independently, though it is not true in real world. Though it's a weakness with the classifier, independence helps the training model to learn parameters fast and in a very simplified way [3,4,5].

Developing the classifier, its handling of integer numbers and float numbers was different in that, the classifier acquired 100% model accuracy with integer data but about 94-97% accuracy with float data.

B. Collaborative filtering Recommenders

The history of user profiles and what items they like, user's interactions with the items by either purchasing them or spending substantial time with the items play important role in predictions done by the system. Collaborative filtering suffers from data sparsity due to lack of enough rated data from past user interactions, either because many users do not express their preferences, for example through rating or liking an item, or because the users are new and have not interacted with the items before. This lack of enough data, weakens the power of prediction thus recommendations suggested become not so accurate [6,7, 8]. Another weakness with Collaborative filtering is that its recommendations are based on similarity of items and popular items tends to have common features thus putting popular items to be recommended more, and little unknown items are never suggested, not because they are bad choices but because the CFRS are biased on similarity [9]. Another weakness of Collaborative filtering recommender system is that they are not scalable especially with lots of computations of user-item matrices that would take a lot of computation resources [7]. In order to overcome the shortcomings of the CFRS, many models have been studied and developed so as to

generate personalized recommendation systems. As outlined by [10] such models were those that dealt with data sparsity, modeled using dimensionality reduction methods, neural networks and many other methods [7, 11, 12], but [10] concludes that there were no unique models for the real-world purview.

In this paper, data sparsity was handled by classifying the input data into two labels, low rated and unrated data as negative label and other data as positive label. These two data separations saw only rated data input in the Collaborative filtering recommender systems, thus minimizing sparsity at input level.

C. Association Rule Mining

Large volumes of non-numeric data can be analyzed using other methods to mine and learn new patterns or associations. Association rule mining is among the commonly used methods while mining for associations among data items by discovering frequent itemsets that are interrelated, which are discovered using the if/then relationship rules and quantified by the support and confidence metrics [13]. ARM was aimed to get correlated relationships by analyzing the data for patterns given user-items interactions. Important of this was discovering of rules, which were significant to build recommendations based on the strong discovered rules.

Given itemset I of $\{i_1, i_2, i_3, \ldots i_m\}$ items in a transaction T of $\{T_1, T_2, T_3, \ldots T_k\}$. Each transaction T has a unique ID, which contains a subset of items in I. An implication of the form: X→Y, where X, Y⊆ I. Both X and Y are itemsets where X is the antecedent and Y the consequent of the rule. Retained association rules must satisfy a minimum support (minsupport) *s* and minimum confidence (mincofidence) *c*. Support of frequent itemset is defined as part of transactions in T where X is a subset

$$Support\ (s) = \frac{X}{|T|} \quad (2)$$

Confidence of rule is defined as the probability of observing Y given that we observe X,

$$Confidence\ (X \rightarrow Y = \frac{s(X \cup Y}{s(X)} \quad (3)$$

Discovering the frequent itemsets was computationally intensive and would be more intensive especially if the transactions in the dataset were massive [13, 14]. While using the 1M MovieLens datasets, rules discovered were different as compared to when using the 100K MovieLens dataset. This had a direct effect on the final recommendations. While using massive datasets though consumption of computing resources was high, it produced better association rules.

III. METHODOLOGY

Proposed hybrid model is a collection of various machine learning (ML) algorithms taking different input data at different stages depending on the ML algorithm been used. The NB classifier input was the cleaned data, with selected predictors, while the CFRS model input was the positive labelled data of highly rated movies (with ratings from 2.0 to 5.0). The ARM data input were the negative labelled data, which was unrated and low-rated movies (with 0 ratings and below 2.0 ratings). The outputs of the CFRS and ARM models were taken through another ensemble formulation to have the final list of top recommended items.

The following Figure 1 shows how data was before classification. Unrated data was indicated by 0 which was 93.7%, with only 6.3% rated. This demonstrates how sparse data can be and if used to calculate recommendations, the predictive power of the model could be compromised. Naïve bayes classifier was used to classify these data into a binarized format of two class labels.

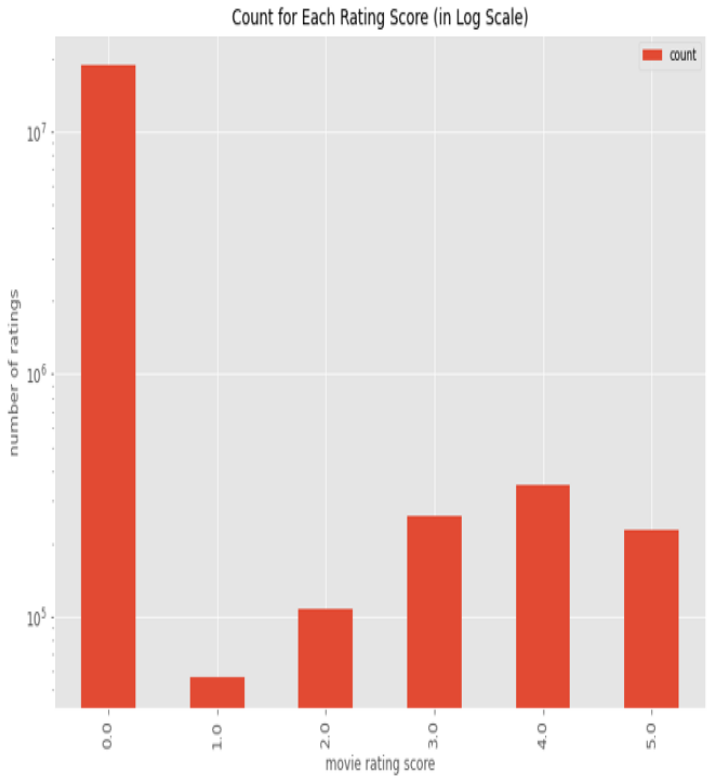

Figure 1 Dataset showing how ratings were spread in 1M MovieLens dataset before classification

Classification was aimed at reduction of sparse data in the CFRS and allow some 'sort of' scaling up of favorable inputs in the CFRS model. As shown on Figure 2 below, the unrated and low-rated ratings data were input in the association rule mining model using the Apriori algorithm, for pattern discovery, which was used for recommendations.

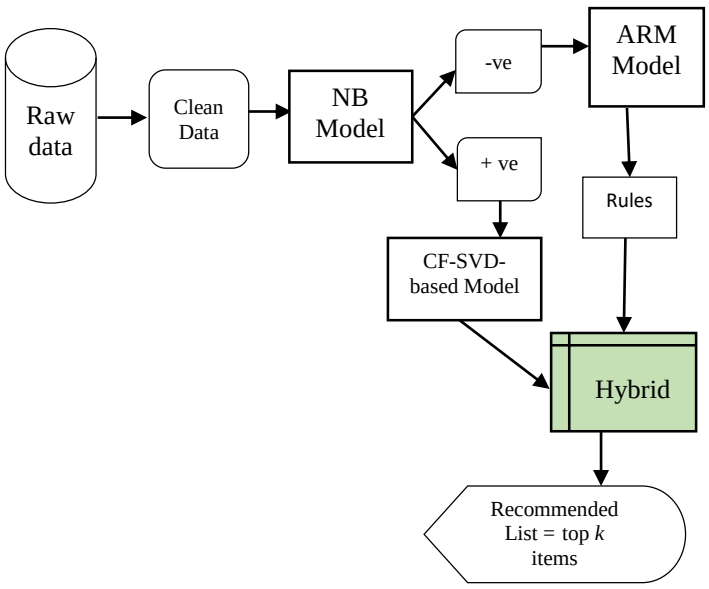

Figure 2 Naive Bayes-based Hybrid Recommender Model

IV. HYBRID RECOMMENDER MODEL

Proposed recommender model uses hybrid techniques to make recommendations. The hybrid model combines the naïve bayes classifier to output data, which would be used by Singular Value Decomposition Collaborative Filtering (SVD-CFRS) model and the ARM model. Results from the CFRS and the ARM models were combined using anensemble formulation that involved the rules mined to approve final recommended list to be presented to the user.

A. Naïve Bayes classifier

Increasingly, sparsely rated data increases the size of the user-item (m*n) matrix and consequently, surge on the complexity in the computation of the matrix factorizations used in CFRS [7]. Inputting data ratings to create a user-item matrix was countered with large memory requirements thus problem of scaling up when we input more data. A methodical way of data reduction was proposed to cut down the volumes of sparse input data without impacting the data set's overall information [15]. Data preparation and selection of predictors that would improve the overall performance of the model was performed through feature selection method and the cleaned data was then used as input to the NB model. The naïve Bayes classifier uses Bayesian theorem to calculate the priors by taking all predictors as independent of each other. The classifier was used to group input dataset into two class labels over a given threshold. After the classification, the data were stored into two separate *csv* files.

Input: cleaned data
Output: two class (+ve and -ve) labeled data
Step 1: input data from dataset
Step 2: split data into 80% training and 20% testing
Step 3: fit training data
Step 4: build model
Step 5: output

The model accuracy attained 100% while using 1M MovieLens data and about 94% while using 100K dataset. This was attributed by the ratings predictor, which was an integer in the 1M dataset and a float in the 100K dataset. This confirmed that ML algorithms work well with integer than other type of data [16]. The data after been separated into two datasets, one with ratings above 2, which were labeled positive and the other with zero-rated and below 2 ratings as negative label.

B. SVD-Collaborative filtering

Collaborative filtering systems uses similar preferences expressed in a user-item matrix. The power of recommendation systems lies on the kind of data input and the characteristics of algorithms used to fit and learn for better prediction accuracy.

Sparse data for recommendation system results in poor recommendations. Grouping data into those highly rated and low rated / no ratings was thought as a way to deal with sparsity. The CFRS model was developed using SVD matrix factorization. The separation of the user and the item through SVD matrix factorizations facilitates the building of similarity functions with the item latent factors. This method offers accurate predictions and its computation is of high efficiency and expansibility if the data input is not sparse [17]. The matrix factorization was of the form

$$R_{n*m} = U_{n*f} * V_{f*m} \quad (4)$$

Where
R_{nm} is the user-item matrix
U_{nf} is the relationship between *n* number of users and *f* characteristics
V_{fm} is the relationship between *f* features and *m* items
f specifies the hidden latent factors.

SVD constructs a matrix with the row of users and columns of items and the elements given by the users' ratings. SVD decomposes a matrix into three other matrices and extracts the factors from the factorization of a high-level (user-item-rating) matrix [17]. For instance, our items were movies in different categories. SVD generated factors when looking into genre dimension space, such as action vs comedy. The input was the positive labelled data from the NB classifier, which resulted to highly rated items (movies) been top recommendations for the particular active user.

Input: positive labelled data (ratings >= threshold)
Output: top *n* highly rated recommendations
Step 1: input data
Step 2: SVD- matrix factorization
Step 3: select active user
Step 4: output recommendations based on the active user

C. Association Rule mining Algorithm

Apriori algorithm was used to discover rules and patterns from the dataset that was labelled negative, from NB classifier. The algorithm operates on two-part principle. The *if* which is defined as the antecedent or tail of the rule, and the *then* statement defined as the consequent or head of the rule. The rules were of the form $A \rightarrow B$, where A was the antecedent and the B was the consequent with association been determined based on support and confidence parameters, where frequency was used to determine the two parameters.

Input: negative labelled data (unrated movies and those rated <threshold)
Output: selected rules
Step 1: input data
Step 2: set minsupport and mincofidence
Step 3: find frequent itemsets
Step 4: collect inferred set of rules into a DataFrame
Step 5: set threshold to select used rules

Support depicts the number of times a given pattern was observed in a given data repository, or between different data repositories. Confidence indicated how many times each observation made of a given pattern, proved to be true. Strong rules were those with support higher than min-support and confidence higher than min-confidence. Association rule mining assesses the confidence of the established rules [13,18].

D. Hybrid Ensemble

Hybrid model developed outputs results by combining recommendation from collaborative filtering recommender systems with those rules discovered from ARM model to formulate the final score. Final top *k* recommended items were formulated from the ensemble hybrid function.
Input: SVD-based collaborative filtering results and rules selected from the ARM model
Output: top *k* recommended items (movies)

Step 1: From selected rules on ARM model and recommended items from CFRS
Step 2: Output the antecedent and the consequent
Step 3: Repeat
For any favourite movie recommended
Output genre
Step 4: Test rules against recommended movies
Approve recommendation if true
Final list of recommendations for the active user.

```
Define function (ensemble)
  get recommended movies = for current user
   get recommended movie genre
   get recommended movies id
output recommended movie list
store recommended movie list on DataFrame
output DataFrame D
/*check if genre in recommended item is in the rules inferred, if yes, approve for final recommendation otherwise don't approve*/

Define function(results)
 get selected rules = s
  while rules
   list final results
    antecedent= s [0]
    consequent = s [1]
  end while
 output antecedent, consequent
          while i, r in D
            recommended movie id =r[rid]
            recommended movie genre =r[r_genre]
          output recommended movie
             if antecedent in recommended movie genre
               if consequent in recommended movie genre
               final results = approve
            elseif consequent in recommended movie genre
             if antecedent in recommended movie genre
              final results = approve
            else
             final results = disapprove
                    endif
                  endif
                endif
              endif
       output results = final results
```

The final results approved were a combination of the rated movies, which were recommended by the SVD-based collaborative filtering model and the unrated movies, which informed the rules from the ARM model. If the model was purely collaborative filtering, the unrated movies could not be considered for recommendation since they form the sparse data but by having the hybrid ensemble, they were part of contribution in the final result, thus giving more competent recommendations.

V. RESULTS AND DISCUSSIONS

In this section, experimental results conducted using python on Jupyter notebook are described and discussed.

a) Experimental Data

Dataset used were that of MovieLens 1M containing 1,000,209 anonymous ratings of 3,952 movies provided by 6,040 users. While MovieLens 100Kcontains 100,000 ratings from 943 users on 1,682 movies, who had at least rated 20 movies. The dataset was collected by GroupLens Research project [19].

b) Naïve bayes classifier results

Applying the dataset to naïve bayes classifier using a criterion threshold of ratings above or equal to 2.0 as favorable ratings. The classifier labelled the favorable data with positive label and unfavorable with a negative label. It was noted that when classifying ratings with float format, the accuracy decreased to 94%, as shown in Table 1 and when classifying ratings with integer data format, the classifier acquired 100% accuracy, as shown in Table 2. For effective results, we used the 1M dataset having ratings with integer format and that acquired 100% accuracy in classification, as shown in Table 2, to validate final results. True and False indicates whether the classification was correctly done or not. Rating 1.5 was classified positive instead of negative, thus a false classification. This meant that some ratings that were less than threshold were classified with positive label instead of negative label.

Table 1 NB Classifier with float format ratings and 94% classification accuracy

INDEX	Predicted label	Ratings	Classification
1	positive	1.5	FALSE
2	positive	5	TRUE
3	positive	3.5	TRUE
4	positive	3	TRUE
5	positive	1.5	FALSE
6	positive	5	TRUE
7	positive	4	TRUE
8	positive	3.5	TRUE
9	positive	4	TRUE
10	positive	4	TRUE

Table 2 NB Classifier with integer ratings and 100% classification accuracy

INDEX	Predicted Label	Rating	Classification
1	positive	4	TRUE
2	positive	4	TRUE
3	positive	2	TRUE
4	positive	3	TRUE
5	positive	4	TRUE
6	positive	5	TRUE
7	positive	4	TRUE
8	positive	4	TRUE
9	negative	1	TRUE

c) Collaborative filtering recommender model results

The classified data was stored into two different files, for usage in modeling the collaborative filtering recommender systems model and association rules mining model as earlier shown in Figure 2 above. The positive labeled data (+ve) was input in the SVD-based Collaborative filtering model. Data without classification for the entire data set of 1,000,209 ratings of 3,952 movies provided by 6,040 users. The data input using the positive class label was 188,728 ratings of 27,278 movies, drastically reducing the data input in the CFRS. Reduction of sparsely rated data was a novel technique used in this work to scale up useful data input to the collaborative filtering model, which previously was marked with low scaling up, due to considerable amount of sparse data. The CFRS model results were based on the active user's previous watched and rated movies. The following Table 3 shows an excerpt of CFRS results, for user id 1741, with 5 top ranked movies the user has never seen previously. This excerpt was among 200 active users sampled. Sampling was done in simple random sampling method.

Table 3 NB optimized SVD-based CF (OSVD-CFR) model recommendation list

Ranking	User Id	Recommended Movie Id	Rating	Movie Title	Genre
1	1741	1925	5	Wings (1927)	Drama, Romance, War
2	1741	342	5	Muriel's Wedding (1994)	Comedy, Romance
3	1741	1301	5	Forbidden Planet (1956)	Sci-Fi
4	1741	909	5	Apartment, The (1960)	Comedy, Drama
5	1741	509	5	Piano, The (1993)	Drama, Romance

Running SVD-based Collaborative filtering model with no data optimization for the first 5 items, the results were as shown in Table 4.

Table 4 Recommendations from unoptimized data using SVD CF

Rank	Movie Id	Title	Genre	User Id	Rating
1	26487	Star 80 (1983)	Drama	1741	3
0	1425	Fierce Creatures (1997)	Comedy	1741	2
Recommended Movies based on active user preferences					
1	2571	Matrix, The (1999)	Action, Sci-Fi, Thriller		
2	2028	Saving Private Ryan (1998)	Action, Drama, War		
3	1291	Indiana Jones and the Last Crusade (1989)	Action, Adventure		
4	223	Clerks (1994)	Comedy		
5	1220	Blues Brothers, The (1980)	Action, Comedy, Musical		

d) Association Rule Mining model results

The data labeled negative (-ve) from the NB classifier was input in an Association Rule Mining (ARM) model. The results from the model were discovered patterns based on other factors, specifically the genre. While the CFRS model used ratings of the movies given by the users, ARM used the genres of movies watched by users, whether rated or not. The aim of the model was to discover unique patterns and relationships of movies as watched by various users. The results showing the patterns of movies watched are as shown on Table 5 below. It was discovered that those users who watched action movies, also watched adventure movies, with an almost 13% support and 41% confidence, and all those who watched adventure movies also watched action movies with support of 13% and confidence rate of 58%.

Table 5 ARM mined movie patterns and relationships

Serial No.	Antecedents	Consequents	Support %	Confidence %
1	(Action)	(Adventure)	12.72	41.2
2	(Adventure)	(Action)	12.72	57.89
3	(Thriller)	(Action)	11.91	51.76
4	(Action)	(Thriller)	11.91	38.58
5	(Romance)	(Comedy)	11.79	63.75
6	(Comedy)	(Romance)	11.79	24.58
7	(Action)	(Sci-Fi)	10.98	35.58
8	(Sci-Fi)	(Action)	10.98	57.93
9	(Romance)	(Drama)	9.13	49.38
10	(Drama)	(Romance)	9.13	31.35

e) Hybrid-model results

The ensemble hybrid model gave the final recommendation list after considering the results of the CFRS model and the ARM model. The ensemble calculated final recommendations based on the genres of the recommended top ranked CFRS list against the rules (antecedent and consequent) list of ARM model. Table 6 below shows an excerpt of results based on a recommended movie list with approved and not approved, where approve were those movies recommended for the final list. They were marked as 'approved' since they met the ensemble formulation based on the users' favorite movie genre against those recommended by the SVD-CFRS model and the ARM model (Rules). Those marked 'not approved' did not meet the ensemble formulation, since there were no rules to approve as shown in Table 6.

Table 6 Hybrid-ensemble results with approvals

Index	Active user Favorite movie genre	Recom_ genre	Recom movie id	Approved	Rules	Rating
1	Drama, Comedy	Drama Romance, War	1925	True	Drama, Romance	5
2	Drama, Comedy	Comedy, Romance	342	True	Comedy, Romance	5
3	Drama, Comedy	Sci-Fi	1301	False	-	5
4	Drama, Comedy	Comedy, Drama	909	False	-	5
5	Drama, Comedy	Drama, Romance	509	True	Drama, Romance	5
6	Drama, Comedy	Adventure, Crime, Sci-Fi, Thriller	3770	False	-	5
7	Drama, Comedy	Comedy, Drama	3543	False	-	5
8	Drama, Comedy	Comedy, Romance	898	True	Comedy, Romance	5
9	Drama, Comedy	Drama, War	1224	False	-	5
10	Drama, Comedy	Action	2949	False	-	5

The recommended final list for the active user [user id =1741], had four movies [approved = True] from top 10 recommended. The four new movies from pure SVD-Collaborative filtering recommender systems without data optimizationwere also not recommended as shown on Table 6. The results from recommendations of ensemble showed suggestions of higher ratings being the top ranked. Over the 200 active users sampled, an average of 68% had new items (movies) been recommended and approved.

It was confirmed that machine learning works better with integer data other than float data, as was shown on the Naïve Bayes classifier that had 100% accuracy with integer ratings and about 94% accuracy with float data. The idea of data segregation and having input rated data within threshold and above ratings using the NB classifier worked well and gave the expected improved highly ranked recommendations as shown on Table 3. The hybrid ensemble final results were suggested after running the results of SVD-CFRS model over the rules from ARM model. The efficiency sought after in reduction of sparse data into collaborative filtering recommender systems model and suggestions of highly rated items(movies) was achieved. The model also showed new items been suggested to

the active user, novelty concept that is key for recommender systems.

VI. CONCLUSIONS AND FUTURE WORK

We presented a hybrid recommender model based on ensembles of two algorithms, the collaborative filtering using the SVD matrix factorization and the association rule mining using Apriori algorithm, which were receiving their inputs from a Naïve Bayes classifier. The classifier was aimed to group the data into two class labels- negative and positive, those with low/no rating and those with high ratings, respectively. NB classification was 100% accurate, with data classified positive fed into SVD-CFR model and those labeled negative into ARM model. The final result was generated with approved items forming the final list of recommendations. We did not measure the prediction accuracy of the hybrid ensemble model against the pure collaborative filtering recommender model with no data optimization but we confirmed that the recommendations suggested on the final list were new to the active user. This was a good way to confirm efficiency of recommendations, given the 68% approvals.

The contribution of this work was that naïve bayes classifier was used to optimize data input to have no sparse data (low rated and no rated movies) input in the Collaborative filtering recommender systems, as a measure to curb sparsity. Another contribution on this work was having the sparse data used for recommendation, but in another model other than the Collaborative filtering recommender systems. We used the ARM using Apriori algorithm to get rules and associations from users-items matrix using different predictor other than the response variable used in the collaborative filtering model.

In future, we shall be keen to evaluate and validate the prediction accuracy of the hybrid ensemble against the pure collaborative filtering model with no data optimization.

REFERENCES

[1] X. Guan, C. Li and Y. Guan, "Matrix Factorization with Rating Completion: An Enhanced SVD Model for Collaborative Filtering Recommender Systems," in IEEE Access, vol. 5, pp. 27668-27678, 2017, doi: 10.1109/ACCESS.2017.2772226.

[2] Li, Y., Bradshaw, J., & Sharma, Y. (2019, May). Are generative classifiers more robust to adversarial attacks? In *International Conference on Machine Learning* (pp. 3804-3814). PMLR.

[3] Lavecchia, A. (2015). Machine-learning approaches in drug discovery: methods and applications. Drugs discovery today, 20(3), 318-331.

[4] Çano, E., & Morisio, M. (2017). Hybrid recommender systems: A systematic literature review. Intelligent Data Analysis, 21(6), 1487-1524.

[5] Portugal, I., Alencar, P., & Cowan, D. (2018). The use of machine learning algorithms in recommender systems: A systematic review. Expert Systems with Applications, 97, 205-227.

[6] Isinkaye, F. O., Folajimi, Y. O., & Ojokoh, B. A. (2015). Recommendation systems: Principles, methods and evaluation. Egyptian Informatics Journal, 16(3), 261-273.

[7] Wei, J., He, J., Chen, K., Zhou, Y., & Tang, Z. (2017). Collaborative filtering and deep learning-based recommendation system for cold start items. *Expert Systems with Applications*, *69*, 29-39.

[8] Li, J., Jing, M., Lu, K., Zhu, L., Yang, Y., & Huang, Z. (2019, July). From zero-shot learning to cold-start recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 33, pp. 4189-4196).

[9] Hu, Y., Shi, W., Li, H., & Hu, X. (2017). Mitigating data sparsity using similarity reinforcement- enhanced collaborative filtering. *ACM Transactions on Internet Technology (TOIT)*, *17*(3), 1-20.

[10] Yao, L., Xu, Z., Zhou, X., & Lev, B. (2019). Synergies Between Association Rules and Collaborative Filtering in Recommender System: An Application to Auto Industry. In *Data Science and Digital Business* (pp. 65-80). Springer, Cham.

[11] Paradarami, T. K., Bastian, N. D., & Wightman, J. L. (2017). A hybrid recommender system using artificial neural networks. *Expert Systems with Applications*, *83*, 300-313.

[12] Seo, Y. D., Kim, Y. G., Lee, E., & Baik, D. K. (2017). Personalized recommender system based on friendship strength in social network services. *Expert Systems with Applications*, *69*, 135-148.

[13] Aggarwal, C. C. (2016). *Recommender systems* (Vol. 1). Cham: Springer International Publishing.

[14] L. Zhang, W. Wang and Y. Zhang, "Privacy Preserving Association Rule Mining: Taxonomy, Techniques, and Metrics," in IEEE Access, vol. 7, pp. 45032-45047, 2019, doi: 10.1109/ACCESS.2019.2908452.

[15] Natarajan, S., Vairavasundaram, S., Natarajan, S., & Gandomi, A. H. (2020). Resolving data sparsity and cold start problem in collaborative filtering recommender system using linked open data. *Expert Systems with Applications*, *149*, 113248.

[16] Brownlee, J. (2019, November 27th). How to Choose a Feature Selection Method for Machine Learning. Retrieved on August 30, 2020 from: https://machinelearningmastery.com/feature-selection-with-real-and-categorical-data

[17] Chen, V. X., & Tang, T. Y. (2019, August). Incorporating singular value decomposition in user-based collaborative filtering technique for a movie recommendation system: A comparative study. In *Proceedings of the 2019 the International Conference on Pattern Recognition and Artificial Intelligence* (pp. 12-15).

[18] Falk, K. (2019). *Practical recommender systems*. Manning Publications.

[19] Da Silva, J. F. G., de Moura Junior, N. N., & Caloba, L. P. (2018, July). Effects of data sparsity on recommender systems based on collaborative filtering. In *2018 International Joint Conference on Neural Networks (IJCNN)* (pp. 1-8). IEEE.