THE JOURNAL OF COMPUTER SCIENCE AND ITS APPLICATIONS
Vol. 24, No 2, December, 2017

A MODEL-BASED COLLABORATIVE FILTERING WITH DIMENSIONALITY REDUCTION

[1]Adewole, A. Philip and [2]Kawedo, C. Victor
Department of Computer Sciences, University of Lagos, Lagos, Nigeria
[1]Padewole@unilag.edu.ng, [2]vcniga@gmail.com

ABSTRACT
In this day and age, the measure of data accessible online multiplies exponentially. With such development rate, it is getting to be distinctly troublesome for clients to approach things of interest subsequently bringing about information overload issue. This overload produces information in very high dimensions and makes it challenging for these systems to suit or accommodate this increment in data. One of the issues with high-dimensional datasets is that, in many cases, not all the measured factors are "vital" for comprehending the underlying phenomena of interest. The use of mathematical procedures to tackle these problems by reducing the dimensions of the data can successfully alleviate such problems and generate more accurate recommendations. This paper proposes a Model-Based Collaborative Filtering (CF) algorithm that integrates dimensionality reduction technique to lessen known limitations of collaborative filtering techniques. The algorithm consists of building a recommender system for movies using data from the MovieLens Recommender System containing 100,000 ratings. The analytic model was constructed using the standard CRISP-DM methodology. According to the experimental results obtained, the proposed algorithm proved to be very effective as far as dealing with both the sparsity and scalability problems and thus produced more accurate predictions and recommendations when contrasted with the standard Item-based CF technique and the random CF technique.

Keywords: *collaborative filtering, dimensionality reduction, model-based, scalability, sparsit*

1.0 INTRODUCTION

In today's world, the amount of information available online increases exponentially. Consistently, we are immersed with decisions and choices. With such growth rate, it is becoming difficult for users to approach items of interest hence causing information overload problem. Indeed, even straightforward choices can be difficult without earlier direct information of the options [1]. Generally, individuals have utilized an assortment of techniques to take care of such basic decision-making issues: suggestions and notice from their associates, acquiring data from a trusted outsider, or just checking the Internet. Would it not be awesome to have a reasonable individual counsel who helps us make good decisions efficiently? These systems are called Recommender Systems[2],[3]. The construction of systems that support users in their decision making is the main goal of the field of recommender systems. Two main paradigms have emerged as a result of studying the problem of recommending items. Content-based and Collaborative filtering Paradigms. Content-based procedures require gathering external information that might not be accessible or simple to gather[4]. An alternative to the content filtering technique depends just on past client conduct or behavior. For instance, past transactions or item ratings, without requiring the formation of express profiles. This approach is known as Collaborative filtering, an expression created by the developers of Tapestry, the first recommender system[4]. Instinctively, this approach accepts that, if clients concur about the quality and importance of some things, then they will probably concur about other things [1]. Two approaches exist for collaborative filtering, model-based and memory-based. With thememory-based approach, the database of entries is loaded into memory and used directly to generate a recommendation. A disadvantage of this approach is that as the data gets sparse, its performance decreases, and this often occurs with web-related data. This makes it difficult for such systems to be expanded to meet future needs, thereby hindering the scalability of this approach and creating problems with very large datasets. In contrast to this strategy, the Model-based approach first builds a model by using data mining and machine learning algorithms that attempt to find patterns based on training the data. These are then used to make predictions for real data. This approach has a more all-encompassing objective to uncover latent factors that explain observed ratings. It handles the sparse data better than the memory based approach, improves scalability and the prediction performance and gives an intuitive rationale for the recommendations[5].

In collaborative filtering, different aspects of an item are rated by users in new dimensions, thereby increasing the size and sparsity of the rating matrix. A solution to these problems is a mathematical procedure that effectively confronts these problems by reducing the dimensionality of the initial data. In highly dimensional spaces, the notions of density and distance between points become less significant. This is known as the Curse of Dimensionality. Dimensionality reduction techniques help overcome this problem by transforming the original high-dimensional space into a lower-dimensional space. Some relevant dimensionality reduction algorithms in the context of Recommender Systems are Principal Component Analysis (PCA), Singular Value Decomposition (SVD) and Latent Semantic Indexing[6]. The questions that arise and which this research aims to answer are: Can we represent each data point with fewer features, without losing much information and still reproduce most of the variability of the data set? Can the observations be explained by linear combinations of few underlying factors? Is there a way to develop a scalablealgorithm that can enable us to extend collaborative filtering to large user bases and facilitate deployment on e-commerce sites? Can we find a method that lends itself well to pre-computing the similarity matrix thereby providing performance gains and increasing prediction performance rather than doing this at the time when prediction or recommendation is needed? The aim of this paper is to propose a method of combining model-based collaborative filtering with dimensionality reduction technique using principal component analysis to improve the prediction accuracy and recommendation quality of movie recommendations to viewers.

2.0 RECOMMENDER SYSTEMS

Recommender systems are engines that offer automated and customized suggestions of items to be of use to customers. This customized suggestions relate to several decision-making processes carried out by customers, such as "What do I wear? Which car would be advisable for me to purchase? Which film would be a good idea for me to lease [1]? The construction of systems that support users in their decision making is the main goal of the field of recommender systems [2], [3]. Two principal things are required for a recommender system to function properly. Evidence about the fondness or preference of the user for particular items and a means to decide if an item will be interesting or fascinating for a user[7]. To determine how fascinating an item will be for a user usually, depends on the computation of the similarities that exist among users and items in the system. This will be discussed in more details in the later part of this section.

2.1 Classification of Recommender Systems

Recommender systems can be characterized as indicated by the prediction technique they use to make predictions.

A. Content-Based

This approach attempts to retrieve useful information from items of the collection. It extracts features from the items themselves rather than depending on user's behavior or ratings, permitting recommendation of thenovelnotpreviouslyseen items. This extracted information indicates how much the items are relevant to the user[8]. The general rule of content-based approaches is to distinguish the common attributes of items that have gotten a great rating from a user and after that prescribe to that user new items that share these qualities. Since these strategies rely on the investigation of the content, the nature and quality of the accessible information about items is a determinant factor in the quality of the outcomes delivered by the system[7].

B. Collaborative Filtering

Different from the content-based approaches, which utilize the content of items earlier rated by a user, collaborative filtering approaches depend on the ratings of the user and also those of different users in the system. The key thought is that the rating of a user for a novel item is probably going to be like that of another user in the system if both users have rated other items in a similar way[9]. The larger part of collaborative filtering systems in operation today, work by first creating predictions of the user's preference for items that have not been rated by the user and after that deliver their suggestions by ranking contender items by anticipated preferences [1]. Collaborative filtering can be grouped into two broad categories of Memory-based and Model-based.

- **Memory-Based**

The memory-based is also reffered to as neighborhood approach. It computes similarities between neighbours on memory. In this technique, the entire database of entries stored in the system is inserted into memory and subsequently used directly to predict ratings and produce recommendations. At first, these calculations make use of different statistical strategies and heuristics to recognize a set of users similar to the active user, known as neighbors. Once that set of neighbors is created, several algorithms are utilized to combine the preferences of these neighbors. The prediction is then processed as an aggregate of their ratings[10]. A probable deficiency of memory-based strategies is that they are to a great extent very delicate to data sparseness. For them to be relevant, the similarity measure on which they are based in reality regularly requires that a minimum amount of users have entered some minimum number of ratings. Likewise, these techniques frequently experience the ill effects of scalability issues.

- **Model-Based**

Model-based algorithms were proposed to tackle a portion of the weaknesses of memory-based strategies. In contrast to neighborhood-based systems, which use the stored ratings directly in the prediction, model-based approaches use these ratings to learn a predictive model. The technique behind model-based algorithm comprises in learning a model on the ratings and after that utilizing it to make predictions. The hidden target is to recognize complex patterns in the data and to utilize it to produce smart an intelligent suggestions. Model-based algorithms utilize procedures from linear algebra (SVD, PCA) or techniques obtained from the machine learning community (Neural networks, bayesian models, clustering models)[10]. The general hypothesis is that factors signifying latent attributes of the users and items in the system, such as the preference class of users and the classification class of items are utilized to model the user-item interactions. This model is then trained utilizing the available information and later used to suggest ratings of users for novel items[9]

C. *Hybrid*

If the information about the community is available and detailed information can be gathered about the individual items, then the recommender system could be enhanced by hybridizing content-based techniques with collaborative filtering. The quality of a recommender system can be improved by integrating content into collaborative filtering. This holds especially when data is too sparse, as additional content information can be a solution to fitting global probabilistic models[7].

2.2 Fundamental Problems of Collaborative Filtering Recommender Systems

A. *Sparsity*

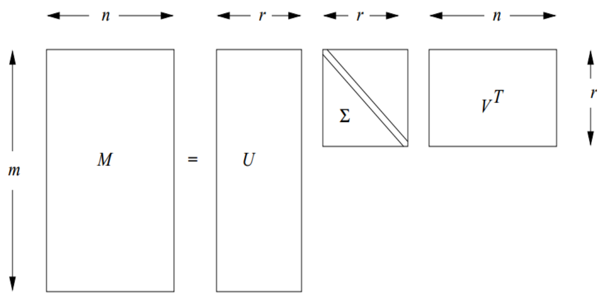
This issue emerges as a result of the scarce or insufficient number of evaluations and feedback data supplied by the users, which limits the usablity of the recommender systems. Due to the few ratings of the total number of items available in the system, the user-item matrix is along these lines to a great degree, sparse. Due to the fact that the collaborative filtering algorithm is majorly centered around similarity computation evaluated over all the rated items, the high degree of sparsity can bring about less accurate predictions or poor recommendations. It is very likely that the similarity between two given users in a very sparse user-item matrix is zero, making collaborative filtering futile[11],[12].

B. *Scalability*

The issue of scalabilty arises as a result of the inability of the recommender system to handle a growing amount of users and items in the database. Collaborative filtering searches the entire database to carry out its functions and therefore experiences poor scalability as more and more items and users are included in the database. As the number of items and users rises, the time and effort used to carry out the collaborative filtering computations increases exponentially[13]. As a result of this increase, the productivity of the system is reduced.

2.3 Mathematical Solutions to the Problems of Recommender Systems

Fig 1: The form of a singular value decomposition
Source: (Leskovec, Rajaraman, & Ullman, 2014)



These problems can be confronted successfully by making use of mathematical procedures that discover efficient ways of reducing the dimensionality of the initial data. One technique to achieve this is Singular Value decomposition. This is a more general method of understanding change of basis[14]. Due to SVD'S capacity to be used in collaboration with standard filtering methods, it was suggested by [15],[16], majorly bacause of its proficiency in producing the best-low ranked aapproximation of the original data set. This paper focuses on an alternative, butthe closely related approach to SVD called principal component analysis (PCA). In a similar manner to SVD, PCA can reduce the dimensionality of the initial data set and produce more efficient predictions and recommendations.It is a mathematical solution that attempts to transform a set of possibly related variables into a new set of unrelated variables whose members are known as principal components. Each of these components corresponds to linear combinations of the initial variables[16]. They are usually sorted in reduced variability, meaning that the primary principal component which is a grouping of the original varibales with the greatest amount of variation, has the biggest conceivable variance; each succeeding component has the most elevated conceivable variance under the constraint that it is orthogonal to the previous component. Since the principal components obtained from the procedure are sorted as far as their variance, keeping the first "m" principal components ought to likewise hold the greater part of the data while decreasing the dataset dimensionality.

A. Singular Value Decomposition

In SVD, the rows and columns of the matrix are connected by a lesser number of "concepts". A smaller, but close estimate of the original matrix is represented by eliminating the least significant concepts[17].The SVD of a matrix **M** is the factorisation of **M** into three component matrices such that

$$\mathbf{M = U\,\Sigma VT} \qquad \text{Eq. (1)}$$

where **M** is an m x n matrix and **r** is the rank of **M**. **U** is an m x r column-orthogonal matrix. **T** is an n x r column-orthogonal matrix. **V** is always used in its transposed form, therefore it is the rows of $\mathbf{V^T}$ thatare orthogonal. **Σ** is a diagonal matrix whose elements are the singular values of the decomposition.

An ideal approach to decrease the dimensionality of the original matrix is to set the smallest of the singular values to zero. By setting the s smallest singular values to 0, it implies that we can likewise eliminate the corresponding s segments of U and columns of V^T, since these rows and columns may as well not be there. Therefore the resulting matrix M' obtained uses only the retained singular values[17].

B. Principal Component Analysis

Principal component analysis is the most popular algorithm for dimensionality reduction. It is the best in the view point of mean-square error[18]. The essential objective of principal components analysis is to define variation in a set of correlated variables, $x^T = (x_1, \ldots, x_q)$, in terms of a new set of uncorrelated variables, $y^T = (y_1, \ldots, y_q)$, each of which is a linear combination of the x variables. The PCA algorithm comprises the following five steps as described below:

Steps:

1. The mean is first subtracted from the each of the data dimension.
   This mean is the average across each dimension. This creates a data set whose mean is zero.
2. The covariance matrix is computed:
   $C^{m\,x\,n} = \left(C_{i,j}, C_{i,j} = cov(Dim_i\,, Dim_j)\right)$ Eq. (2)
   Where $C^{m\times n}$is a matrix whose entries are the product of calculating the covariance between two distinct dimensions.
3. The eigenvectors and eigenvalues of the covariance matrix are computed.
4. The components are chosen and a Feature vector is formed
   The principal component of the data set is essentially the eigenvector with the greatest eigenvalue. The number of eigenvectors chosen forms the number of dimensions of the new data set. A Feature vector is formed (matrix of vectors) by taking the eigenvectors chosen from the original list of eigenvectors and creating a matrix with them in the columns.
   $Feature\ Vector = (eig1, eig2, \ldots, eign)$
5. The new data set is derived.
   The transpose of the featue vector is mutiplied by the original data set, transposed.
   $FinalData$
   $= RowFeatureVector \text{ x } RowDataAdjusted$
   Where
   $RowF\ eatureV\ ector$is the matrix of eigenvectors in the columns transposed
   $RowDataAdjusted$ is the mean-adjusted data transposed
   [19].

2.4 Computing Similarity

The similarity values between items in a data set are gotten by observing all the users who have expressed preferences for both items.The similarity between various items in the data set can be figured out by using one of various similarity measures.

A. Cosine Similarity

The similarity is derived by calculating the cosine distance beween two rating vector. The similarity value is gotten by dividing their dot product by the product of their euclidean norms.

$$sim(i,j) = \cos(\vec{\imath},\vec{\jmath}) = \frac{\vec{\imath}\text{x}\vec{\jmath}}{||\vec{\imath}||^2\text{x}||\vec{\jmath}||^2} = \frac{\sum_{k=1}^{n} R_{k,i}R_{k,j}}{\sqrt{\sum_{k=1}^{n} R_{k,i}{}^2 \sum_{k=1}^{n} R_{k,j}{}^2}}$$

Eq. (3)

Where *i* is the target item, *j* is the is the other item, *n* is the total number of ratings given to item i and item j, $R_{k,i}$is the rating given to the target item by user k and $R_{k,j}$is the rating given to the other item by user j.

B. Pearson Correlation

This technique measures the linear dependence or correlation between two rating vector in the dataset. The similarity value is gotten by dividing the covariance of the two vectors by the product of their standard deviation.

$$sim(i,j) = \frac{(R_{k,i} - A_i, R_{k,j} - A_j)}{||R_{k,i} - A_i||\ ||R_{k,j} - A_j||} = \frac{\sum_{k=1}^{n}(R_{k,i} - A_i)(R_{k,j} - A_j)}{\sqrt{\sum_{k=1}^{n}(R_{k,i} - A_i)^2 \mathrm{x} \sum_{k=1}^{n}(R_{k,j} - A_j)^2}}$$

Eq. (4)

Where A_i is the average rating given to the target item i for all users who rated the item and A_j is the average rating given to the other item j for all users who rated the item.

C. Euclidean Distance

This technique is based on the straight-line distance between two rating vector. In its attempt to place preference values between items, it sets up coordinate points and estimates the euclidean distance between each point[20].

$$sim(i,j) = \sqrt{\sum_{k=1}^{n}(R_{k,i} - R_{k,i})^2} \qquad \text{Eq. (5)}$$

Other similarity measures that have been proposed include; mean-squared difference, spearman rank correlation, constrained Pearson correlation and Tanimoto coefficient [1], [20].

3.0 PROPOSED METHODOLOGY AND ALGORITHM

The methodology used for this research was the standard CRISP-DM algorithm. **Cross Industry Standard Process for Data Mining**, normally known by its acronym **CRISP-DM** is a data mining process model that depicts commonly utilized approaches that data mining specialists use to handle issues. CRISP-DM is a comprehensive data mining methodology and process model that provides anybody, from beginners to data mining specialists, with a complete blueprint for conducting a data mining job. CRISP-DM breaks down the life cycle of a data mining project into six phases: business understanding, data understanding, data preparation, modelling, evaluation, and deployment. Fig 2 shows the phases of a data mining process[21].

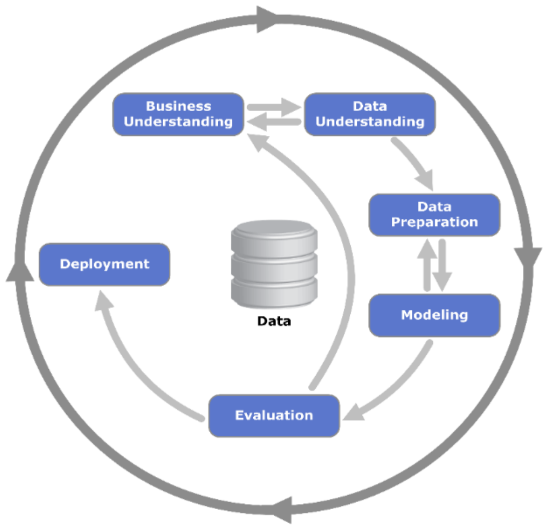


Fig 2: Phases of the CRISP-DM reference model
Source: (Chapman, et al., 2000)

The first stage concentrates on the comprehension of the goals and prerequisites of the project from a business point of view and then transforms this information into a data mining problem. Transforming the dataset into a user-item matrix is an an activity in the data undestanding phase whose goal is to get familiar with the data.Dimensionality reduction is an activity in the data preparation phase which covers all activities to construct the final dataset from the initial raw data.Computing similarity and predictions are activities in the Modelling phase and then the evaluation and deployment of the model.

The Model-Based collaborative filtering algorithm

The proposed Model-Based collaborative filtering algorithm worked as follows:

1. The data set was first loaded and transformed into a real ratings user-item Matrix which was 93.7% sparse
2. Then, Normalization was carried out on the real ratings data to remove rating bias
3. Principal component analysis was then applied to the data set to obtain vectors in a low dimensional space
4. The data was then split into train set and test set
5. The similarity between items was computed using cosine similarity measure
6. The Model was built using the train data set obtained from step 4
7. Predicted ratings was computed using the known sample of the test data
8. The Top-N recommendations were generated.

9. Error calculation was carried out between the prediction and the unknown sample of the test data and also on the accuracy of the recommendetions.

4.0 EXPERIMENTAL SETUP AND RESULT

4.1 Data Set

The data set used for the research was acquired from the MovieLens Recommender System. The data was transformed into a matrix-like object known as realRatingMatrix which contained the dataset about movie ratings. Each of the rows represented a user and each column represented a movie. Each value corresponded to a rating. Exploring the data set, we discovered that they are 943 users and 1682 movies. Fig 3 visualized the first ten users and their ratings on fifteen movies by building a heat map whose colours corresponds to the ratings given by users to movies. The white cells signify a missing value.

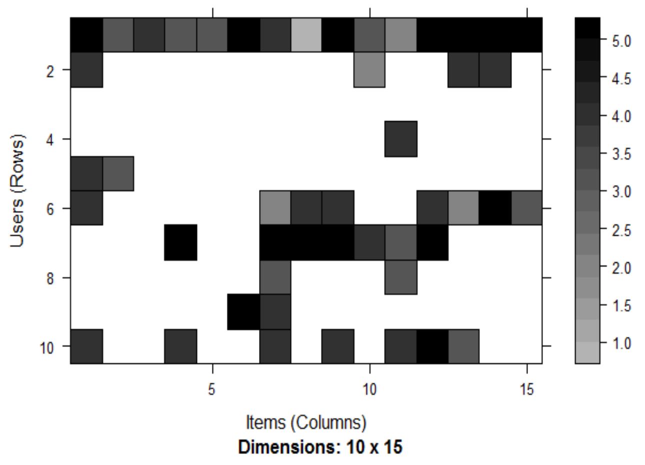


Fig 3. Heat map of the ratings of the first 10 users on 15 movies

4.2 Normalization

The information utilized by recommender system are sometimes biased in ways that can produce undesired results, such as critical users given low ratings to all items or the popularity of some items influencing the ratings. Normalization of ratings implies altering or modifying rating values evaluated on several scales to a notionally common scale. The intention is to align the entire distribution of ratings. Also known as standardization of variables, it attempts to rescale the ratings into a specific range of values[22].

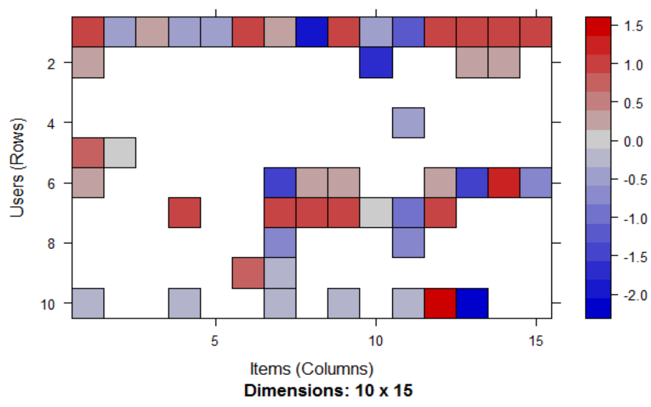


Fig. 4: Heatmap of normalized rating data for the first 10 users on 15 movies

The z-score technique was used for the normalization. Each existing rating was subtracted from the mean of all ratings provided by that user, i.e., subtracting the row mean from all ratings in the row, and then dividing by the satndard deviation of the row.

$$h(r_{ui}) = \frac{r_{ui} - \bar{r}_u}{\sigma_u} \qquad \text{Eq. (6)}$$

Where $\bar{r}_u$ corresponds to the mean of all available ratings in row u and σ_u corresponds to the standard deviation of ratings given by user u. Fig 4 shows the heatmap of normalized rating data for the first 10 users on 15 movies. The major and most noticeable difference we can see from this figure and the figure of the unnormalized data of Fig 3 is the colours, and this is as a result of the data being continuous.

4.3 Dimensionality Reduction

The curse of dimensionality is the difficulty in analysing and organising data with hundreds of dimensions, in this case, 943 dimensions are associated with the dataset. This is the problem of too many variables. Consequently, a dimension reduction technique was sought to reduce the many dimensions into a few principal components that explains most of the variation in the dataset. The first 29 principal components explained 0.45325 i.e 45% of the variation within the data set[23], proposed that the selection of components explaining between 70 and 90 percent variation is acceptable. In our case that would be the first 124 components. The reason we have so many pca is because the variation in our data set is democratised across all the dimensions i.e the dimensions have little or no correlation between each other. The summary of the 10 most important principal component is shown in Table 2 and its corresponding bar plot in Fig 5.

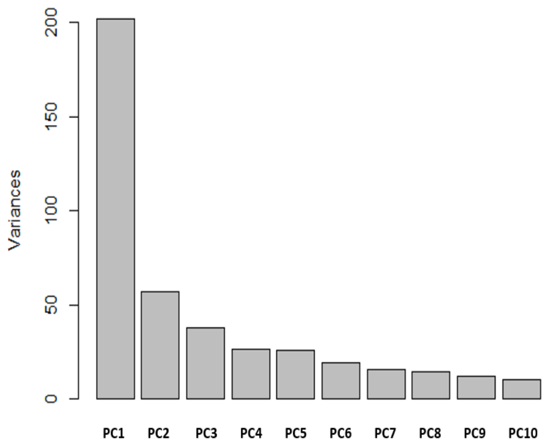


Fig. 5: Bar Plot of the first 5 items

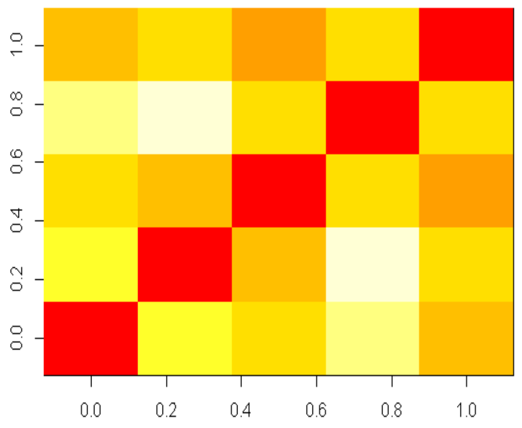


Fig 6: Heatmap of the similarity between the first 5 items

4.4 Computing the Similarity matrix

The cosine similarity measure was used for this paper. Table 1 shows a similarity matrix that displays how similar the first five items are with each other using the cosine distance, while Fig 6 visualizes the similarity between this five items using a heatmap. The rows and columns of both charts correspond to an item, and each of the cells inthe matrix represents the similarity between two items.

Table 1: Similarity matrix of the first 5 items

	1	**2**	**3**	**4**	**5**
1	0	0.4024	0.3302	0.4549	0.2867
2	0.4024	0	0.2731	0.5026	0.3188
3	0.3302	0.2731	0	0.3249	0.2130
4	0.4549	0.5026	0.3249	0	0.3342
5	0.2867	0.3188	0.2130	0.3342	0

Note that the diagonal of the matrix is 0, seeing as it compares each item with itself. Likewise, the diagonal of the heatmap is red, because the redder a cell is, the more similar the two items being compared are. The red diagonal signifies an item compared with itself.

4.5 Splitting the data

The data is split using the “split” method. The evaluation scheme was run once. The users in the data set which corrresponds to the rows of the matrix were randomly partitioned into two sets.The training and test set. 80% ofthe users which corresponded to 754 users where reserved for the test set from which the model learned. 20% of the total users which corresponded to 189 users where reserved for the training set. For each user in the test set, 15 items were withheld or given for the evaluation which was used to generate predictions and recommendations, while the remaining items were used to test the model's accuracy and compute the error.A rating value of 3 was chosen which represented a threshold at which ratings are deemed good forevaluation. This means any item with an actual user rating, equal to or greater than 3 is recognized as being a positive rating in the evaluation procedure. The splitting process produced three sets in all: **Train -** the data used to train the model, **Known -** the portion of the test set with the items used to genererate predictions and recommendations and **Unknown** - the portion of the test set with the items used to test the model’s accuracy and compute the error.

Table 2: Summary of the 10 most important principal component

	PC1	**PC2**	**PC3**	**PC4**	**PC5**	**PC6**	**PC7**	**PC8**	**PC9**	**PC10**
Standard deviation	14.21	7.54	6.13	5.16	5.07	4.42	4.00	3.81	3.46	3.23
Proportion of Variance	0.17	0.05	0.03	0.02	0.02	0.02	0.01	0.01	0.01	0.01
Cumulative Proportion	0.17	0.22	0.25	0.27	0.29	0.31	0.32	0.33	0.34	0.35

Table 3: Distribution of ratings in the Data sets

Data Set	Dimension	Ratings
Training	754 x 1682	79356
Known	189 x 1682	2835
Unknown	189 x 1682	17809

The Model was learned using 754 users which made up 80% of the original data set. The training data set was a 754 x 1682 rating matrix with a total of 79356 ratings. The model was tested on 189 users which were made up of 20% of the original data set. The known test data set was a 189 x 1682 rating matrix with a total of 2835 ratings, which consisted of the 15 items for each user given to the model for the generation of predictions. The predicted ratngs generated was a 189 x 1682 rating matrix with a total of 315063 ratings. The predicted ratings generated accounted for all the cells in the input 189 x 1682 rating matrix that previosly had no rating value. The number of items to be recommended was specified as 5 per user. The algorithm identified the top 5 predicted ratings for each of the 189 users, which are the top-rated items identified by the model.

4.6 Evaluation Metrics and Discussion of Results

Evaluating the results is very important in this kind of application to prevent models from overfitting to the training set, thus decreasing the performance on the test set. To evaluate or assess the model, the recommendations are contrasted with the unknown ratings or the remaining items kept for testing the accuracy of the model. The threshold 3, was used to characterize what constituted a good or a bad item or a utilized and non utilized rating. Ratings above or equal to 3 were considered as good ratings, while ratings under 3 were seen as bad ratings. The error is calculated between the predictions generated and the unknown part of the test data set. This portion of the test data contains all but the 15 items given to test the model. Therefore, the actual (user-provided) and the corresponding predicted ratings were compared to determine the prediction accuracy. It measures how far the estimations created by the technique go astray from their known ratings. The smaller the error value, the better the Recommender System functions. There are two popular approaches to evaluating a recommender system. The predicted ratings can be evaluated and also, the recommendations generated can be evaluated.

A. Evaluating the Predicted Ratings Accuracy

The prediction accuracy measures the accuracy given a suggestion. This technique attempts to measure how close the generated predictions are to the true users rating. It evaluates the predictive ability of a model. The generated predictions are passed to the evaluation scheme to generate the evaluation metrics. There are different techniques used to evaluate the predictions, some of which are the Mean Absolute Error (MAE), the Root Mean Square Error (RMSE) and the Mean Square Error (MSE).

- **Mean Absolute Error**

This metric finds the absolute difference between the actual ratings values and the predicted ratings values and then measures the mean.

$$MAE = \frac{1}{|k|}\sum_{(i,j)\in k}|r_{ij} - \hat{r}_{i,j}| \quad \text{Eq. (7)}$$

- **Root Mean Square Error**

This evaluation metric finds the difference between the actual ratings values and the predicted ratings values and then measures the standard deviation.

$$RMSE = \sqrt{\frac{\sum_{(i,j)\in k}\left(r_{ij}-\hat{r}_{i,j}\right)^2}{|k|}} \quad \text{Eq. (8)}$$

- **Mean Square Error**

This metric calculates the squared difference between the actual ratings value and the prdicted ratings values and then measures the mean. It squares the RMSE.

$$MSE = \frac{\sum_{(i,j)\in k}\left(r_{ij}-\hat{r}_{i,j}\right)^2}{|k|} \quad \text{Eq. (9)}$$

To evaluate the quality and accuracy of the proposed algorithm, the predictions generated were evaluated using the three evaluation metrics discussed above – the RMSE, the MSE and the MAE. It is worth mentioning that the lower the values of these metrics, the higher the prediction accuracy of the algorithm. Utilizing these evaluation metrics, we compared and contrasted the effectiveness of the proposed algorithm with existing approaches and chose which algorithm offered a better performance. For benchmark purposes, the results of our proposed algorithm is compared with results of two widely used colllaborative filtering algorithms. The Item-based algorithm and the Random collaborative algorithm. Table 4.4 shows a summary of the results derived from all three algorithms.

Table 4: Comparing the prediction accuracy of each algorithm

	RMSE	MSE	MAE
RANDOM	1.4124	1.9950	1.1142
ITEM-BASED	1.1467	1.3149	0.8360
MODEL	1.0662	1.1368	0.8642

From the summary of Table 4, the proposed algorithm has demonstrated its superiority over the benchmark algorithms by obtaining the highest prediction accuracy on the RMSE and the MSE metric. In the aspect of RMSE, the model decreased by 7.02% compared with theItem-based algorithm and 24.51% compared with the random CF algorithm. In the aspect of MSE, the model decreased by 13.54% compard with the Item-based algorithm and 43.02% compared with the random CF algorithm. However, its MAE is arguably higher than the Item-based algorithm by 0.0282 which was a 3.26% increase.But when compared with the random CF algorithm, the MSE was reduced by 27.86%.

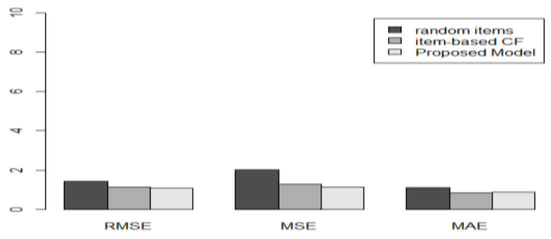


Figure 7: Comparing the prediction accuracy of each algorithm

Figure 7 shows a plot of that visualizes the predicted ratings accuracy of the algorithms showing the various evaluation metrics that was used to determine the accuracy of the predicted ratings.

B. Evaluating the Accuracy of the Recommendations

This metric is not interested in whether or not the system accurately predicted the ratings of movies not previously rated by the user, but rather its interest is in whether or not the recommended movies are indeed viewed, purchased or utilized by the user. It evaluates the frequency with which a recommender system settles on accurate or inaccurate choices about whether an item is good or will be utilized by the user. It compares the recommendations made with the purchases having a positve rating i.e., being utilized. The outcome of a recommedation to a user can have four possible results which are summarised in table5.

Table 5 : Summary of the Possible Results of a Recommender

	RECOMMENDED	NOT RECOMMENDED
UTILIZED	True-Positive (TP)	False-Negative (FN)
NOT UTILIZED	False-Positive (FP)	True- Negative (TN)

True Positives (TP) - This outcome represents system recommended items that have been utilized by the user.**False Positives (FP) -**This outcome represents system recommended items that have not been utilized by the user.**False Negatives (FN) -** This outcome represents system non recommended items that have been utilized by the user.**True Negatives (TN) -** These items represent system non recommended items that have not been utilized by the user.

- **Precision**

This is the also known as Positive Predictive Value. This measures the portion of the recommended items that are utilized. It represents the probability that a recommended item is used by the user.

$$\boldsymbol{Precision} = \frac{\mathrm{TP}}{\mathrm{TP+FP}} \qquad \text{Eq. (10)}$$

- **Recall**

This is also known as **sensitivity** or **True Positive Rate (TPR)**. This measures the portion of the utilized items that are recommended. It represents the probability that a utilized item will be recommended.

$$\boldsymbol{Recall} = \frac{\mathrm{TP}}{\mathrm{TP+FN}} \qquad \text{Eq. (11)}$$

- **False Posititve Rate (FPR)**

This measures the portion of the non utilized items that have been recommended.

$$\boldsymbol{False\ Positive\ Rate} = \frac{\mathrm{FP}}{\mathrm{FP+TN}} \qquad \text{Eq. (12)}$$

Table 6: Confusion Matrix showing performance Indices

	TP	FP	FN	TN	Precision	Recall	TPR	FPR
1	0.4444	0.5556	70.69	1595	0.4444	0.01005	0.01005	0.000343
3	1.228	1.772	69.91	1594	0.4092	0.02335	0.02335	0.001095
5	1.862	3.138	69.28	1593	0.3725	0.03486	0.03486	0.001942
10	3.206	6.794	67.93	1589	0.3206	0.05689	0.05689	0.00421
15	4.492	10.51	66.65	1585	0.2995	0.07576	0.07576	0.006509
20	5.677	14.32	65.46	1582	0.2839	0.09583	0.09583	0.008877

A rating threshold of 3 was defined to represent positive ratings. The table below shows a summary of the results of the evaluation. It shows a confusion matrix that contains the performance indices used for the evaluation. We evaluated the Top-1, Top-3, Top-5, Top-10 and the Top-20 recommendations.

The table 6 summary shows an approximate precision score of about 0.4 at 1, meaning that 40% of the movies recommended was utilized. Therefore, 4 out of every 10 movies recommended was purchased. The recall score of approximately 0.06 at 10 indicates that about 6% of the utilized or purchased movies were recommeded by the model. We can also deduce from the pattern seen in the table summary that as the number of movies recommended increases, so does the recall metric, while the precision decreases. The FPR score of 0.004 at 10 means that 4 out of every 1000 movies that were not utilized were actually recommended. The TPR gives the same result as the Recall which evaluates that 6% of the utilzed movies were recommended by the model.

In a case where the number of recommendations is not predetermiined, it becomes preferable to evaluate the recommendations offered to a user over a spread of recommendation list sizes, rather than utilizing a fixed size. Therefore, curves that evaluate precision to recall, known as Precision-Recall curves, or evaluate true positve rate to false positive rate, known as Receiver Operating Characteristics are used. Both curves measure the percentage of the utilized items that are indeed recommended. The threshold is varied on how many items the recommender system is allowed to recommend. These thresholds trace out the curves.

- **Precision-Recall Curve**

This curve highlights the proportion of the recommended items that have been utilized. It displays and illustrates the trade-off between the precision and the recall.

- **Receiver Operating Characteristics Curve (ROC)**

This curve highlights the proportion of items which are not utilized items but end up being recommended. The ROC curve displays these two factors, the TPR and the FPR. It plots the sensitivity against the complement of the specificity.

To evaluate the exactness and accuracy of the proposed algorithm, the recommendations generated were evaluated using the precision-recall and the ROC curves. The **area under the curve(AUC)**, which is the area under the precision-recall and ROC curves is an indication that suggests the performance of the model. A good performance index has a large area under the curve. The algorithm with the larger area under the curve signifies the algorithm with the higher recommendation accuracy. Making use of these metrics, we compared the exactness of the proposed algorithm in making recommendations to new users with the existing approaches and decided which of the algorithms performed better. The result of our algorithm is compared with the Item-based algorithm and the Random collaborative algorithm. Fig 8 and Fig 9 visualizes the comparison between the algorithms with respect to their

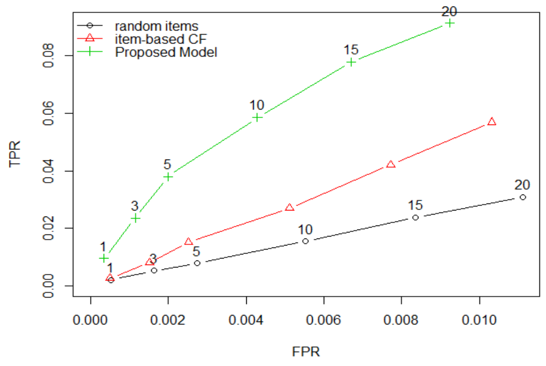


Fig 8: Precision-Recall

precision-recall curves and ROC curves respectively.

The charts help us summarize the performance indices across all the folds. The Precision-Recall curve of Fig 8 reflects the relationship and trade-off between precision and recall. From the curve, it is evident that as the number of movies recommended increases from 1 to 20, the precision decreases, while the recall increases with the number of

movies.Therefore as the number of movies recommended increased, the percentage of recommended items utilized reduced, while the percentage of utilized items recommended increased. On the other hand, the ROC curve of Fig 9 shows that as the number of movies recommended increases, both the True Positive Rate and the False Positive Rate increase as well. Therefore, as the number of movies recommended increased, the percentage of utilized items recommended and the percentage of non utilized items recommended also increased at a steady rate. For the experimental data set, the proposed model clearly outperformed the Item-based and the random collaborative algorithms. In Fig 8 and 9, it is proven that the proposed method dominates the other methodssince, for every length of the top-N list, it offered a much preffered combination of precision and recall as well as TPR and FPR.This is evidenced by the larger area under the curve for the proposed model.

4.7 Contribution to Knowledge

The following few contributions to the body of knowledge were accomplished from this study:

1. This work presented a model which exemplified a unified framework that offered the necessary infractructure to both develop and test collaborative filtering with dimensionality reduction.
2. The model lent itself well to precomputation of the similarity matrix, thereby incresing prediction performance
3. The model has the ability to extend collaborative filtering to theextremely large dataset, thereby facilitating deployment on e-commerce sites.
4. The execution of this work provided the capability of merging the task of collaborative filtering with the step-by-step phases present in the Cross Industry Standard Process for Data Mining.

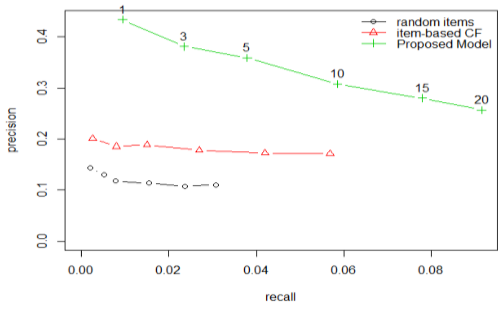


Fig 9: ROC Curve

5.0 CONCLUSION AND FUTURE WORK

In this work, a model-based collaborative filtering with dimensionality reduction is proposed, by integrating the standard CRISP-DM methodology which breaks down the life cycle of the project into six phases, namely; business understanding, data understanding, data preparation, modelling phase, evaluation and the deployment phase. In this work, the first five phases were considered. The proposed model eliminated two fundamental problems in recommender systems which are data sparsity and scalability. The dimensionality reduction technique helps to overcome the data sparsity problem in the rating matrix, while the model-based approach helps to overcome the scalability problem. Therefore, the computation time is greatly decreased and the prediction performance is increased. The application of the evaluation metrics and techniques showed that the prediction method on the average achieves a mean absolute error of 0.8642 and a root mean square error of 1.0662 on data that is unknown to the model. The experimental results obtained have demonstrated that the proposed algorithm can fundamentally enhance the accuracy of predictions and usability of recommendations and furthermore tackle the problems of scalability and sparsity.

Throughout the research, many potentially intriguing and significant research subjects displayed themselves. Yet to keep concentrated on the goals of this research, these subjects had to be disposed of. Some of these subjects are listed.

- The implementation of the recommender system proposed gives rise to a question which is the fact that it is based on the ratings only. In many other situations, there are additional data sources such as the descriptions of the items and the profiles of users. A preferred solution is a synchronization of all relevant information.
- This work was carried out in an offline setup with offline data. However, in a live collaborative filtering system, ratings are continuously added and changed by users. A possible reserch subject is an effect of updating ratings or preferences on the offline data computed.
- The evaluation metrics discussed in this work involves measuring variables that we believe will affect the utility of a recommender system to the user and consequently affect the reaction of the user to the system. The question of how to directly evaluate user "reaction" to a recommender system is a subject for futher research. This can employ surveys and interviews, users behaviour can be logged in and subjcted to several sorts of investigations.

REFERENCES

[1] M. D. Ekstrand, J. T. Riedl and J. A. Konstan, "Collaborative Filtering Recommender Systems," *Foundations and Trends Human–Computer Interaction,* vol. 4, no. 2, pp. 81-173, 2010.

[2] D. Jannach, M. Zanker, A. Felfernig and G. Friedrich, Recommender Systems: An Introduction, New York: Cambridge University Press, 2011.

[3] F. Ricci, L. Rokach, B. Shapira and P. B. Kantor, Recommender Systems Handbook, New York: Springer Science+Business Media, 2010.

[4] Y. Koren, R. Bell and C. Volinsky, "Matrix Factorization techniques for recommender systems," *Institute of Electrical and Electronics Engineers Computer Society,* pp. 42-49, 2009.

[5] C. A. Levinas, "An Analysis of Memory Based Collaborative Filtering Recommender Systems with Improvement Proposals.," (Master's thesis, Universitat de Barcelona, Universitat Politècnica de Catalunya, Universitat Rovira i Virgili), 2014.

[6] X. Amatriain, A. Jaimes, N. Oliver and J. M. Pujol, "Data Mining Methods for Recommender Systems," in *Recommender System Handbook*, New York, Springer, 2011, pp. 39-67.

[7] H. S. D. Castillo, "Hybrid Content-Based Collaborative-Filtering Audio Recommendations," (Master's thesis, University of Twente, The Netherlands), 2007.

[8] A. Uitdenbogerd and R. G. v. Schyndel, "A review of factors affecting music recommender success," *The International Society of Music Information Retrieval,* pp. 13-17, 2002.

[9] C. Desrosiers and G. Karypis, "A comprehensive survey of neighborhood-based recommendation methods," in *Recommender Systems Handbook*, Newyork, Springer Science+Business Media, 2011, pp. 107-144.

[10] G. Louppe, "Collaborative filtering - Scalable approaches using restricted Boltzmann machines," (Master's thesis, University of Liege), 2010.

[11] A. Kumar and A. Sharma, "Alleviating Sparsity and Scalability Issues in Collaborative Filtering Based Recommender Systems," in *Proceedings of the International Conference on Frontiers of Intelligent Computing: Theory and Applications (FICTA)*, Springer Berlin Heidelberg, 2013, pp. 103-112.

[12] Y. Chen, C. Wu, M. Xie and X. Guo, "Solving the Sparsity Problem in Recommender Systems Using Association Retrieval," *Journal of Computers*, vol. 6, no. 9, pp. 1896-1902, 2011.

[13] N. S. Bhosale and S. S. Pande, "A Survey on Recommendation System for Big Data Applications," *Data Mining and Knowledge Engineering*, vol. 7, no. 1, pp. 42-44, 2015.

[14] J. Shlems, "A tutorial on Principal component Analysis - Derivation, Discussion and Singular Value Decomposition," 2003.

[15] B. Sarwar, "Sparsity, Scalability, and Distribution in Recommender Systems," (Doctoral Thesis, University of Minnesota), 2001.

[16] M. G. Vozalis and K. G. Margaritis, "A Recommender System using Principal Component Analysis," *11th Panhellenic Conference in Informatics*, pp. 271-283, 2016

[17] J. Leskovec, A. Rajaraman and J. D. Ullman, Mining of Massive Datasets, California: Cambridge University Press, 2014, pp. 384-414.

[18] I. K. Fodor, "A survey of dimension reduction techniques," Center for Applied Scientiflc Computing, Lawrence Livermore National Laboratory, 2002.

[19] L. I. Smith, "A tutorial on Principal Components Analysis," University of Otago, New Zealand, 2002.

[20] Y. Lee, "Recommendation System using Collaborative Filtering," (Master's thesis, San Jose State University), California, 2015.

[21] C. Shearer, "The CRISP-DM Model: The New Blueprint for Data Mining," *Journal of Data Warehousing*, vol. 5, no. 4, pp. 13-22, 2000.

[22] M. F. M. Mohsin, A. R. Hamdan and A. A. Bakar, "The Effect of Normalization for Real Value Negative Selection Algorithm," Springer-Verlag Berlin Heidelberg , Berlin, 2011.

[23] B. Everitt and T. Hothorn, "An Introduction to Applied Multivariate Analysis with R (Use R!)," Springer, New York, NY, 2011.