Enhancing Collaborative Filtering Recommendation Systems Using SHAP-Driven Demographic Data Clustering

Aditya Singampalli

July 29, 2024

Abstract

Recommendation systems have been critical in enhancing user experiences online in various domains through relevant product suggestions extracted from user behavior. Collaborative filtering, one of the most common recommendation algorithms, utilizes interaction data between users and items. However, this data is rather one-dimensional and fails to consider additional factors such as demographics. This research paper explores the integration of SHapley Additive exPlanations (SHAP) values with demographic data to enhance the clustering of similar users before employing collaborative filtering. Using the MovieLens 100k dataset, an XGboost model predicts user genre preferences from demographic information. SHAP values are then calculated from this model to determine the significance of each demographic feature in influencing the model output. Then, the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm is augmented by the calculated SHAP values, creating communities of similar users. Within each of these communities, traditional collaborative filtering techniques are employed and predicted ratings are calculated for each user within each cluster. Based on the Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE), the SHAP-value-enhanced model performed better. This method provides more accurate and tailored user recommendations and explores the application of Shapley values from game theory to the machine learning tasks present in recommendation systems.

Keywords : SHAP, collaborative filtering, memory-based filtering, demographic data, clustering, DBSCAN, prediction, recommender systems

1 Introduction

As e-commerce, entertainment, and music platforms have grown over the decades, the need to cater to user interests and automate product suggestions has never been greater. This is where recommendation systems come into play. These technologies have become a significant part of our online experience, as users are presented with tailored suggestions based on their online behaviors and preferences. By employing sophisticated machine learning techniques and algorithms to interpret vast datasets, these systems can deliver personalized recommendations to users online (Adomavicius & Tuzhilin, 2005).

However, traditional recommendation algorithms, namely collaborative filtering, typically rely purely on the interaction data between users and items, such as the ratings users give items. While these are effective and sufficiently capable to a certain extent, they are not without their limitations. One of the main issues with these systems is that they ignore additional contextual details, mainly demographic factors such as age, gender, and occupation. Not accounting for these factors can result in lower-quality recommendations and decreased user satisfaction (Rashid et al., 2002). This gap indicates the necessity for more sophisticated approaches that not only take into account user behavior data but also historical data such as demographic characteristics.

Recent studies exploring recommendation systems have indeed discussed the integration of demographic data to overcome these limitations. By incorporating demographic data into recommendation algorithms, they can obtain a deeper understanding of the preferences of the user, contributing to better-quality suggestions. This method improves recommendation accuracy but also increases user satisfaction as users are not presented with items that resonate with their interests (Sridevi & Rao, 2017).

In this context, SHAP (SHapley Additive exPlanations), a concept derived from cooperative game theory but adapted for machine learning, have been proven to be a potent tool for understanding and interpreting machine learning models. They are adapted from Shapley values from collaborative game theory, which measures the contribution of each 'player' to the reward generated by the group as a whole. In this context, SHAP values measure the contribution of each feature, or 'player' to the output, or 'reward' of the machine-learning model, allowing an easier interpretation of the importance of said features (Lundberg & Lee, 2017). In this scenario, the application of SHAP values in recommendation algorithms can provide insight into the extent to which demographic characteristics influence online user preferences, which can be extremely helpful in clustering users that are similar and can lead to more accurate recommendations.

The primary goal of this study is to explore how the integration of SHAP values with demographic data can improve online recommendation systems of products, specifically movie recommendations. This approach utilizes SHAP values to quantify demographic feature influence on the preferences of users. Based on the significance of these features, accurate user clusters can be made and prediction capabilities will be enhanced. The specific objectives of this research are:

- To utilize SHAP values to quantify the influence of demographic features on user movie preferences.
- To form more accurate user clusters based on demographic similarities informed by SHAP values.
- To evaluate the effectiveness of the proposed SHAP-enhanced collaborative filtering approach through appropriate evaluation metrics and comparison with traditional methods.

This research contributes to the field of recommendation systems through the introduction of a novel approach that incorporates demographic data, aided by SHAP-value interpretation, with collaborative filtering. This research provides a means for online platforms to improve user satisfaction through highly personalized and relevant recommendations, which ultimately can lead to a boosted user base and business prosperity.

The structure of this paper is as follows:

- The next section (2) provides a literature review, focused on exploring prior studies related to collaborative filtering, the application of demographic data, and SHAP values.
- In the following methodology section (3), the employed dataset, algorithm, and experimental design are described.
- This is then followed by the results section (4) of the study whereby the findings concerning the study, including the use of SHAP-enhanced collaborative filtering, are highlighted.
- The discussion section(5) analyzes the results, highlights the significance of the study, and identifies its contributions.
- Lastly, the conclusion (section 6) provides a brief overview of the entire research and its outcomes in addition to highlighting potential future research avenues.

2 Literature Review

2.1 Types of Recommendation Systems

Of the three main recommendation algorithms, collaborative filtering, content-based filtering, and hybrid filtering, the former is by far the most popular and widely used. Two main techniques have been explored for creating collaborative filtering recommendation systems: memory-based and model-based collaborative filtering. Memory-based collaborative filtering works by directly utilizing user-item interaction and relies on historical user behavior data to make accurate predictions. It uses similarity measures such as the Pearson correlation coefficient and vector cosine similarity to find similar users and make predictions accordingly. Meanwhile, model-based collaborative filtering techniques build machine-learning models that predict user ratings based on current user-item interactions. It uses matrix factorization to split user-item matrices into latent factors. The dot product of these factors is calculated to make the predictions. P. H. Aditya et al. compared the performance of both techniques in Indonesian e-commerce platforms and found that model-based recommendation systems outperformed memory-based recommendation systems in three regards: accuracy, computation time, and relevance of the recommendations. However, model-based recommendation systems are not only more complex and therefore more difficult to deploy, but they also are not as easily interpretable as memory-based algorithms. As a black-box algorithm that makes predictions on factors unknown to the user, model-based systems lack transparency, making it difficult to understand how specific recommendations are generated (Aditya et al., 2018). Therefore, the collaborative filtering techniques used in this study will mainly involve memory-based ones.

Within memory-based collaborative filtering, there are two main kinds: user-based and item-based filtering. User-based collaborative filtering involves finding similar users based on their past interaction history. Meanwhile, item-based collaborative filtering prioritizes finding items that are similar to what the user previously interacted with. Several studies have explored the effectiveness of both. Boström et al. compared both types of recommendation systems using the MovieLens 100k dataset. Using the vector cosine similarity technique, recommendations were produced through both, but user-based collaborative filtering was found to be superior in terms of accuracy in predicting user ratings and producing recommendations (Boström et al., 2013).

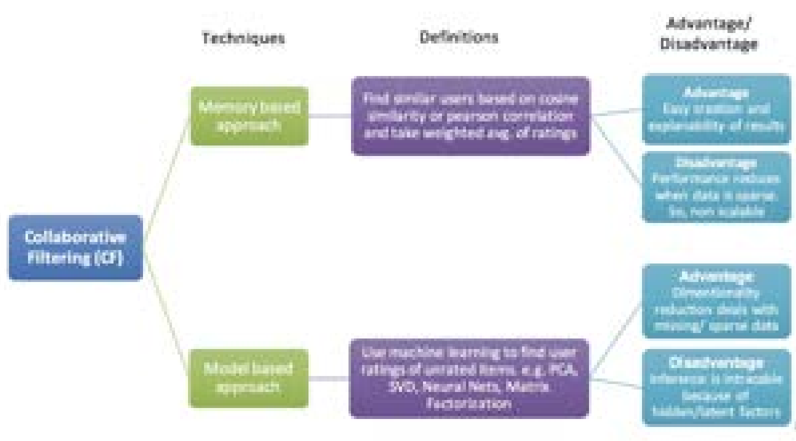

Figure 1: from "Various Types of Movies Recommendation System" - Scientific Figure on ResearchGate.

2.2 Clustering Algorithms

The general approach of clustering the user data into 'similar' communities, before applying the conventional recommendation algorithms has been studied widely. Clustering algorithms are one of the categories of unsupervised machine learning, which aims to sort similar data points in a group. The purpose of clustering is to discover relationships in data that would enable us to make forecasts and draw conclusions.

The clustering algorithms can be categorized into several groups; however, DBSCAN and K-Means are the most popular ones. K-Means is a type of clustering algorithm where the data is partitioned into k clusters depending on the distance between the elements and the mean (centroid) of the cluster. In other words, it employs the selection of k random points that are termed as initial centroids and then proceeds with the process of iteration for the improvement of the centroids of the clusters. DBSCAN, on the other hand, is a density based clustering algorithm which clusters the data points based on how close they are to each other. This is done by finding core points, which are points that have a minimum number of neighboring points within a certain radius, and then enlarging clusters around these points.

While each has their advantages and disadvantages, DBSCAN is more advantageous to use in this scenario because the clusters are not as well separated and the data is much more packed together

(Karl et. al).

Recently, supervised learning using SHAP values are being explored extensively and showing greater results, in terms of cluster accuracy.

Traditional Clustering Versus Supervised Clustering

Traditional approaches to clustering are quite simple, and are typically implemented as follows:

1. Pre-processing is conducted to rescale and tidy the data
2. The data can undergo dimensionality reduction, usually to two dimensions, for visualization purposes
3. Clustering algorithms are applied, and various metrics are used to assess the quality of clusters that are identified

This frequently yields unsatisfactory results, with weakly defined, highly overlapping clusters that are difficult to distinguish. As seen in Fig.2, supervised clustering uses a similar procedure plus one additional step.

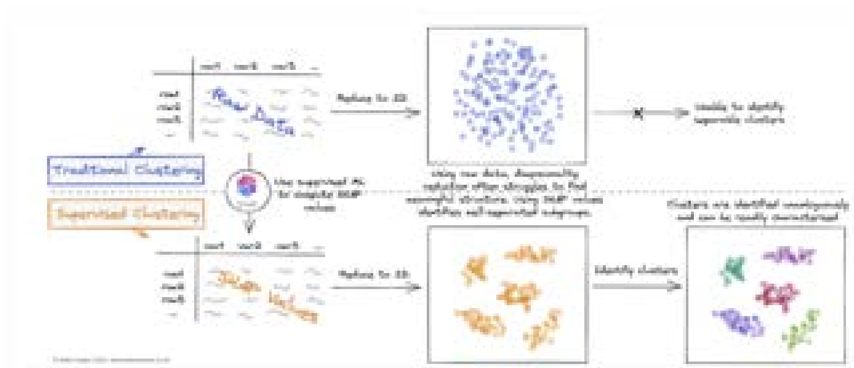

Figure 2: from "Supervised Clustering for Subgroup Discovery: An Application to COVID-19 Symptomatology" - Scientific Figure on ResearchGate.

Supervised clustering first transforms the raw data into SHAP values before clustering on the raw data itself (or an embedding thereof). This entails training a supervised machine learning model on the raw data, then using the trained model to calculate SHAP values. The outcome is an array with the same dimensions as the raw data, but values based on how relevant the data is about the target variable (Cooper et al.).

For this to be possible, the following is a requirement for supervised clustering: there must be a suitable target variable that will be used in the training of the prediction model. This is quite a deviation from the traditional cluster analysis that is normally considered as a purely unsupervised method. Fortunately, many clustering applications are already naturally suited to this process. Once a target variable is identified and these communities of users who behave similarly are created, these recommendation algorithms can make more accurate and relevant recommendations within each user cluster. The clustering preprocessing step also reduces the data dimensionality, making the recommendation algorithm far more efficient.

For example, in their study, Gong proved that clustering users and items before applying collaborative filtering not only improves the scalability of the collaborative filtering algorithm, but it also improves the recommendation qualities. Because the algorithm only has to compute within smaller and more densely populated clusters, the recommendation system can provide more relevant recommendations and improve model scalability (Gong 2010).

Studies centered around recommendation systems have also explored the utilization of demographic data to address the limitations of traditional collaborative filtering. Sridevi and Rao, in their study, first partitioned the users based on their demographic qualities and then applied traditional collaborative filtering techniques. The accuracy of the model, calculated by finding the mean absolute error (MAE), was found to be higher compared to standard collaborative filtering models. The incorporation of demographic data not only improves the relevance of the recommendations but also boosts user satisfaction by suggesting products that the user is interested in (Sridevi and Rao, 2017).

SHAP (SHapley Additive exPlanations) is a game theoretic approach to explain the output of any machine learning model. It connects optimal credit allocation with local explanations using the classic Shapley values from game theory and their related extensions. Sample explanation is demonstrated below in Fig.3.

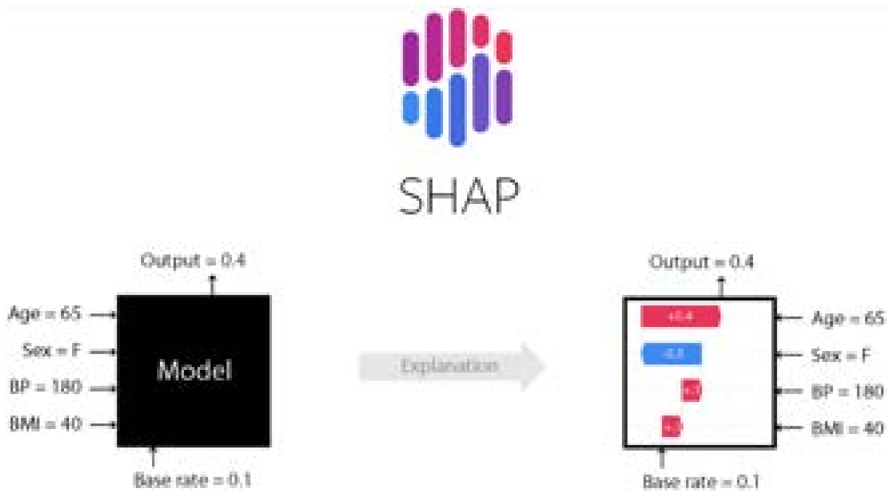

Figure 3: from “Welcome to the SHAP documentation” - Figure from SHAP Documentation

SHAP values provide a measure of the contribution of each input feature to the model’s output, enabling a clear understanding of feature importance. Lundberg and Lee (2017) introduced SHAP values to the machine learning community, highlighting their ability to offer consistent and interpretable insights into model predictions. Applying SHAP values in recommendation systems can offer insights into how demographic characteristics influence user preferences, facilitating the creation of more accurate and interpretable user clusters.

Now regarding SHAP values, studies have been conducted in the past demonstrating them as a useful tool for understanding and explaining machine learning model predictions, but this was not done so specifically on demographic data and to improve recommendation systems. For example, Lundberg and Lee explored in their study how to use SHAP values to assign the amount of influence a certain machine-learning feature has on the model prediction. They found that the use of SHAP made it significantly easier to interpret the ‘weight’ of each feature, and could thus adjust the model accordingly. Gramegna and Giudici compared clustering on the data directly versus the SHAP values to characterize the buying behaviors of insurance customers and found that clusters were better differentiated using the SHAP-based approach. Similarly, applying these SHAP values in recommendation algorithms could provide information regarding how demographic data affects user preferences, which ultimately can lead to more accurate recommendation systems and increased user satisfaction.

3 Methodology

3.1 Proposed Solution

This research is an applied study focused on improving the performance of movie recommendation techniques by incorporating demographic information and SHAP values into conventional collaborative filtering algorithms. The goal is to utilize user demographics to identify similar subgroups by enhancing clustering techniques with Shapely Values, providing focused and streamlined recommendations to new users, and improving the overall performance of the recommendation system. Fig.4 illustrates the multi step process pipeline for clustering-based collaborative filtering analysis method using Shapley values. The experiment is also conducted with clustering based on raw demographics data instead of SHAP values and shows that Shapely based clustering improves the performance of overall performance. Performance is measured for both SHAP and raw data based pipelines using MAE and RMSE metrics where MAE is the average measured error between the recommended ratings and the users' observed ratings. The RMSE is the squared root of the difference between the predicted ratings and the observed ratings. Fig.15 shows the results for both methods.

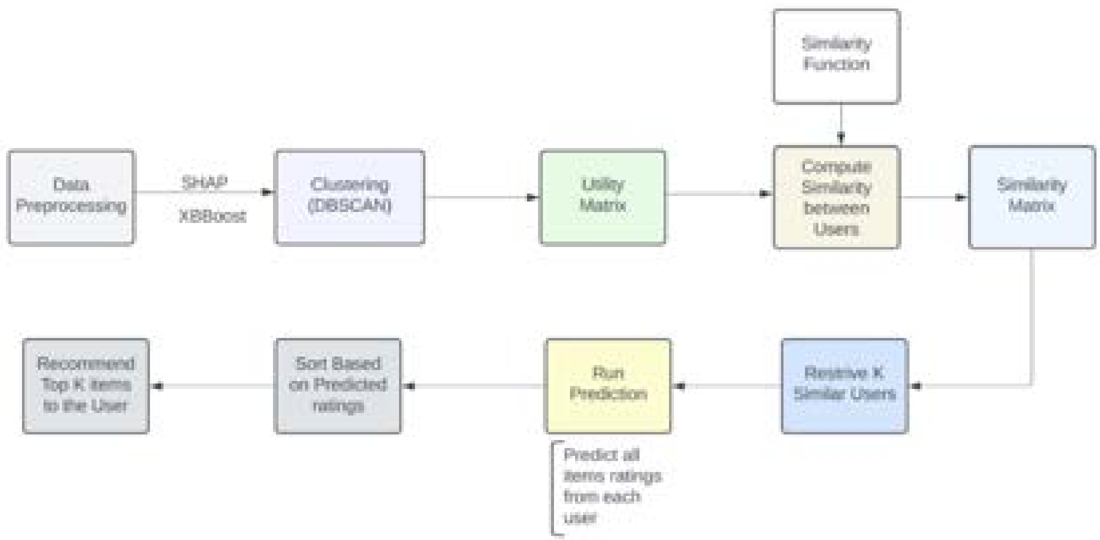

Figure 4: Enhanced collaborative filtering recommendation pipeline with SHAP-value-based clustering

3.2 Data Preprocessing

3.2.1 Data Collection

The research employs the MovieLens 100k dataset which is highly acclaimed for its rich user rating data, users' basic characteristics, and movie attributes. This dataset was selected because it is diverse, and it offers a lot of information on users and items. Specifically, the dataset includes:

- User Demographics: This includes age, gender, occupation, and zip code, and that gives me a four-sided picture of each user.
- Movie Ratings: The ratings can range from 1 to 5, and they represent the user's preferences for the movie which are essential in training and testing the recommendation models.
- Movie Details: Information about movies such as the movie genre information helps in the understanding of the user preferences and thus better recommendations.

MovieLens Dataset Files	File Attributes Description
u.user	The user file contains demographic information about the 943 users. "user id | age | gender | occupation | zip code "
u.item	The item file holds information about the items(movies). "movie id | movie title | release data | video release data | IMDb URL | Unknown | Action | Adventure | Animation | Children's | Comedy | Crime | Documentary | Drama | Fantasy | Film-Noir | Horror | Musical | Mystery | Romance | Thriller | War | Western "
u.data	The data file contains 100,000 ratings by 943 users on 1682 items. "user id | item id | gender | rating | timestamp "

Figure 5: Data Set Files.

3.2.2 Data Analysis

The analysis process involves several key steps, starting with data collection, followed by data preparation, data cleaning, exploratory analysis, modeling, and the final step of reporting.

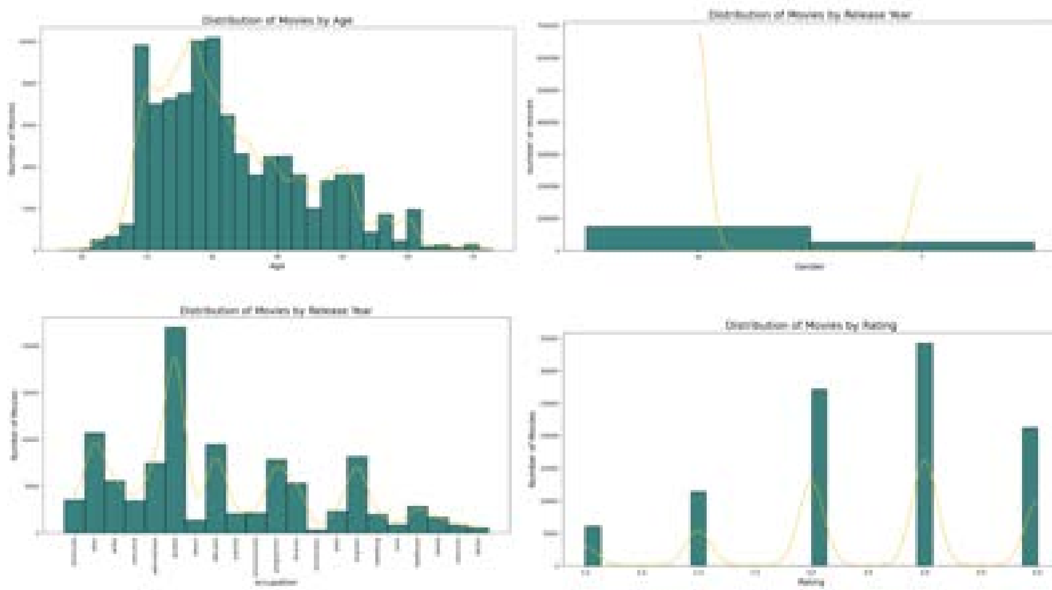

Figure 6: Data Exploration.

3.2.3 Merging Datasets

The first process that has to be performed entails combining the user demographic data set with the movie rating data set and the movie genre data set into one large data set. This merging process makes sure that all the data points do not have any gaps and contain all the information that is needed (Aggarwal, 2016).

userId	age	movieId	gender	occupation	zip_code	title	release_date	genre_names	rating
1	24	1	M	technician	85711	Toy Story (1995)	01-Jan-1995	animation|children|comedy	5.0
1	24	2	M	technician	85711	GoldenEye (1995)	01-Jan-1995	action|adventure|thriller	3.0
1	24	3	M	technician	85711	Four Rooms (1995)	01-Jan-1995	thriller	4.0
1	24	4	M	technician	85711	Get Shorty (1995)	01-Jan-1995	action|comedy|drama	3.0
1	24	5	M	technician	85711	Copycat (1995)	01-Jan-1995	crime|drama|thriller	3.0
...	...	...	...	...	...	...	...	...	...
943	22	1067	M	student	77841	Bottle Rocket (1996)	21-Feb-1996	comedy	2.0
943	22	1074	M	student	77841	Reality Bites (1994)	01-Jan-1994	comedy|drama	4.0
943	22	1188	M	student	77841	Young Guns II (1990)	01-Jan-1990	action|comedy|western	3.0
943	22	1228	M	student	77841	Under Siege 2: Dark Territory (1995)	01-Jan-1995	action	3.0
943	22	1330	M	student	77841	An Unforgettable Summer (1994)	01-Jan-1994	drama	3.0

Figure 7: Illustration of Merged Data Set

Demographics data that is used for clustering as shown in table below.

userId	age	gender	occupation	zip_code
1	24	M	technician	85711
2	53	F	other	94043
3	23	M	writer	32067
4	24	M	technician	43537
5	33	F	other	15213

Figure 8: Demographics data of interest

3.2.4 Handling Categorical Variables

Data like gender and occupation which are categorical in nature are encoded to numerical data through one-hot encoding and label encoding. This conversion is very important for machine learning models to be able to process the data as required (Hastie, Tibshirani, & Friedman, 2009).

3.2.5 Missing Values and Outliers

As an experimental data set, preprocessing steps were already taken to address missing values and outliers, so these steps will not be part of the preprocessing methodology in this paper.

3.2.6 Data Scaling

To normalize and transform features by scaling each feature to the range [0,1] using the MinMaxScaler from SKLearn. By incorporating data scaling, The dominance of any particular feature in the learning process can be prevented.

3.3 Data Clustering Using Demographics Data

In this experiment both supervised clustering using Shapely values and traditional unsupervised clustering with user's demographics data directly to compare the performance.

3.3.1 Supervised Clustering Using Shapley Values

It is a multi-step process as explained in Fig.9 below. It is different from traditional clustering shown in Fig.11 where there is no additional step of model input required to derive clusters.

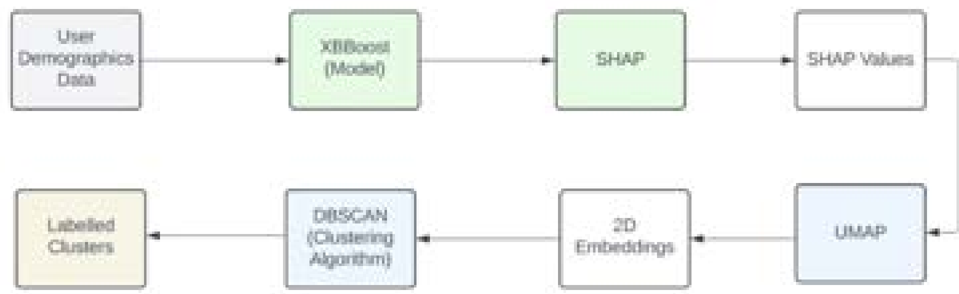

Figure 9: Flow chart outlines the clustering based on Shap values.

3.3.2 XGBoost Model to derive Shape Values

The role of the supervised machine learning model is primarily to serve as a means for obtaining SHAP (SHapley Additive exPlanations) values.

An XGBoost model is then developed to make the prediction of the users' favorite genre from their demographic details. Due to the lack of labeled data in the MovieLens dataset, a new variable called favorite genre was created, which was calculated by taking the genres of the movies that were the highest rated by each user. XGBoost is selected due to the fact that it is very efficient and stable in classification. The model is built from a part of the data and tested on the rest and on different data subsets to check its effectiveness (Chen & Guestrin, 2016).

3.3.3 SHAP Values Calculation

Once the model is trained SHAP values are computed to understand the significance of each demographic feature. SHAP values are another method to estimate the importance of each feature for the model's outcome, and thus, identify which demographic characteristics are most relevant to users' preferences.

The SHAP value for a feature i in instance x is given by:

$$\phi_i(p) = \sum_{S \subseteq N/i} \frac{|S|!(n-|S|-1)!}{n!}(p(S \cup i) - p(S))$$

where N is the set of all features, S is a subset of N not containing i, and f is the model's prediction function.

After obtaining the SHAP values for each feature in each instance, the mean absolute SHAP value for each feature is calculated. This value provides an aggregated measure of the feature's importance across all predictions, indicating which features most significantly influence the model's output. The formula is given as:

$$\text{Mean Absolute SHAP Value}_i = \frac{1}{m}\sum_{j=1}^{m} |\phi_i(x_j)|$$

Where m is the total number of instances, and $\Phi_i(xj)$ is the SHAP value for feature in instance *xj*.

After calculating the mean absolute SHAP values for each demographic feature, they can be visualized in the following bar chart in Figure 10.

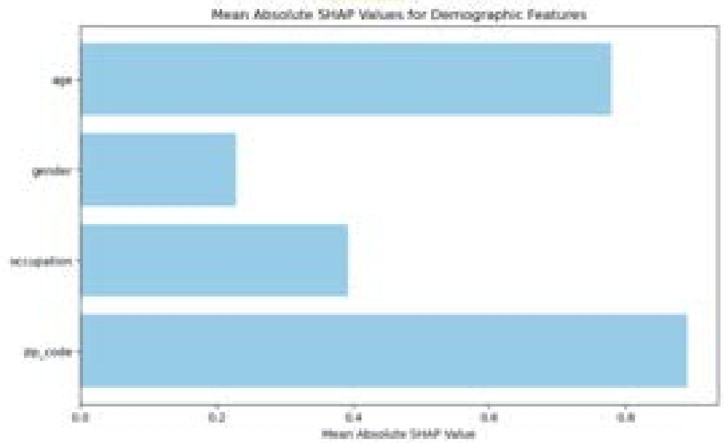

Figure 10: Bar Chart showing the significance of each demographic feature from SHAP Values

3.3.4 Dimensionality Reduction

For visualization of the clusters, dimensionality reduction technique called UMAP (Uniform Manifold Approximation and Projection) is used. UMAP is chosen because it performs well in terms of the global structure and relationships of high-dimensional data being maintained when mapping it to two dimensions (McInnes, Healy, & Melville, 2018) compared with other techniques like PCA. The UMAP embedding was computed for two components, using a local neighborhood (n neighbors) of 40 data points (three times the default value), and a minimum distance between embedded points of zero, enabling data points to be tightly grouped to support the formation of local clusters. This higher than default n neighbors value encourages UMAP to produce more general, global neighborhoods, rather than localized granular structure. Similarly, setting the minimum distance value to zero encourages the formation of densely packed, well-separated clusters of embedded data points . These characteristics serve to optimize the clustering results in the next stage.

3.3.5 DBSCAN Clustering

Demographic features of the users are aggregated, and each of them is multiplied by its corresponding mean absolute SHAP value. For this purpose the DBSCAN (Density-Based Spatial Clustering of Applications with Noise) algorithm is employed. DBSCAN is selected due to its capacity to find clusters in different forms and densities, which is applicable in the case of user data (Ester et al., 1996). Cluster parameters are selected after tuning with different values of eps i.e. maximum distance between two samples and min_samples i.e minimum no of samples in the neighborhood to be considered for a core point. An acceptable silhouette_score($\sim$ 0.633) for values of eps=0.8 and min_samples=20 was observed. Silhouette score is a metric used to calculate the goodness of a clustering technique. Its value ranges from (-1,1). 1 means that the clusters are well apart from each other and clearly distinguish 0 means clusters are indifferent, or we can say that the distance between clusters is not significant. -1 means that the clusters are assigned incorrectly.

Also, clustering is done using the raw data directly, without Shapley values to compare the performance. As shown in Fig.11, raw-based clustering does not need any supervised model as input but it is needed for supervised models.

Figure 11: Process outlines cluster formation using raw data

The clusters formed from DBSCAN for both raw and SHAP data are visualized below. 10 clusters were identified with SHAP data compared to 3 clusters with raw data:

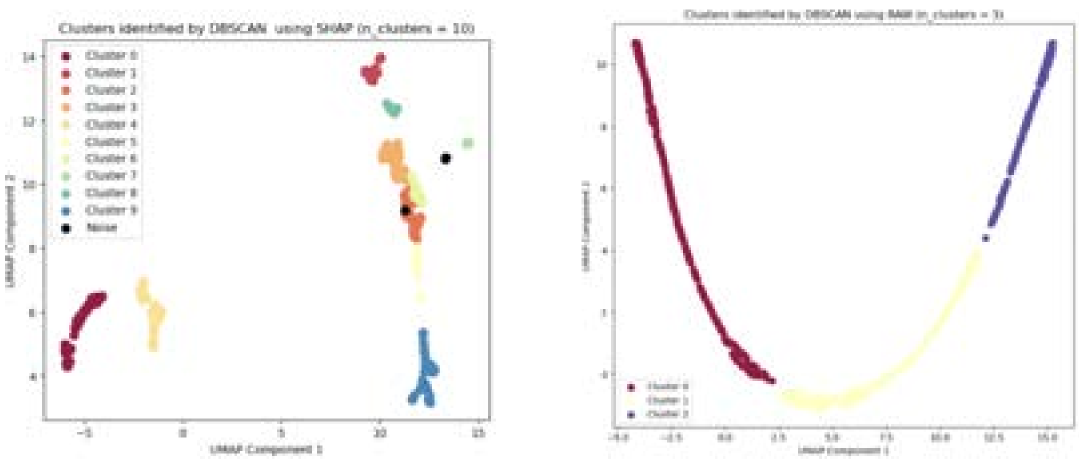

Figure 12: Graphs showing clusters formed from raw features compared to SHAP features

3.4 Collaborative Filtering (Memory-Based)

3.4.1 User-Item Matrix Creation

For each of the user clusters that DBSCAN has identified, the user-item matrix is utilized. This matrix contains the ratings that the users have assigned to movies, serving as the basis for the collaborative filtering.

3.4.2 Cosine Similarity and Weighted Averages

In each cluster, the methods of cosine similarity and weighted average are used for the prediction of film rating. Cosine similarity is used to calculate the similarity of the users or items with respect to the rating vectors and weighted averages are used to sum up these similarities for making predictions. The formula to calculate the cosine similarity between two users, u and v, is given as:

$$\text{Cosine Similarity}(u, v) = \frac{\sum_{i=1}^{n} r_{u,i} \cdot r_{v,i}}{\sqrt{\sum_{i=1}^{n} r_{u,i}^2} \cdot \sqrt{\sum_{i=1}^{n} r_{v,i}^2}}$$

Where $r_{u,i}$ and $r_{v,i}$ represent the ratings of user u and v for movie i, respectively, and n is the total number of movies (Koren et al., 2009).

3.4.3 Prediction Model and Recommend Movies

After computing the cosine similarities, the predicted rating for a movie i by a user u is calculated using a weighted average of the ratings given by similar users. The formula for the predicted rating $p_{u,i}$ is:

$$p_{u,i} = \frac{\sum_{v\in N(u)} \mathrm{Sim}(u,v)\cdot r_{v,i}}{\sum_{v\in N(u)} \mathrm{Sim}(u,v)}$$

Where Sim(u,v) is the cosine similarity between users u and v, and N(u) is the set of users who have rated movie i.

These predicted ratings are then used to recommend the top k movies that users are likely to enjoy and rate well.

3.5 Evaluation of Model Accuracy

The prediction model has been tested by splitting the data into 80% training and 20% testing of user-movie matrix data for all clusters and measuring below accuracy metrics.

3.5.1 Primary Evaluation Metrics

To test the accuracy of the model, the **Mean Absolute Error (MAE)** evaluation metric is employed. This measure of accuracy was selected to measure the deviation of the predicted ratings from the actual ratings, providing a quantitative assessment of the model's performance (Bell et al., 2007). The Formulas for MAE is given as:

$$\mathrm{MAE}(y, \hat{y}) = \frac{1}{n_{\text{samples}}} \sum_{i=0}^{n_{\text{samples}}-1} |y_i - \hat{y}_i|$$

Where $\hat{y}_i$ is the predicted value of the ith sample, and y_i is the corresponding true value, then the mean absolute error (MAE) estimated over n samples as above.

Furthermore, the **Root Mean Squared Error (RMSE)** metric is used to measure the average magnitude of the errors between the predicted ratings and the actual ratings. Specifically, it is the square root of the average of the squared differences between the predicted and actual ratings. RMSE is particularly useful in movie rating prediction as it provides a clear indication of how closely the model's predictions match the true ratings. Additionally, RMSE penalizes larger errors more than MAE, making it more sensitive to outliers.

$$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

Where, y_i is the actual rating, $\hat{y}_i$ is the predicted rating, and n is the total number of ratings.

3.5.2 Secondary Evaluation Metrics

Additionally, Precision, Recall, and F1 score metrics are utilized as secondary evaluation metrics.

Precision, in the context of predicting movie ratings, is the ratio of correctly predicted positive ratings (e.g., high ratings) to the total number of ratings predicted as positive. This metric indicates how accurate the model is when it predicts a positive rating. High precision means that when the model predicts a high rating, it is likely to be correct, ensuring that users receive reliable high-rated movie recommendations. It is the success rate of positive predictions, closer to 1 indicates high success rate.

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}}$$

Where TP is true positive and FP is false positive.

Recall, also known as sensitivity or true positive rate, is the ratio of correctly predicted positive ratings to the total actual positive ratings. This metric measures the model's ability to identify all relevant high ratings. High recall means that the model successfully captures most of the high-rated movies, ensuring that users do not miss out on potentially good recommendations.

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}}$$

Where TP is true positive and FN is false negative.

The **F1 score** is the harmonic mean of precision and recall, providing a single metric that balances both aspects. In the context of movie ratings, the F1 score is especially useful when the distribution of ratings is imbalanced, as it accounts for both the precision and recall of high ratings. An F1 score closer to 1 indicates that the model performs well in predicting high ratings accurately and comprehensively.

$$F_1 = \left(\frac{\text{recall}^{-1} + \text{precision}^{-1}}{2}\right)^{-1} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

4 Experimental Results

In this section, the accuracy of the model will be tested, displayed, and compared with models that do not incorporate SHAP values in the user clustering. As mentioned previously, The evaluation metrics used include MAE and RMSE , providing a quantitative assessment of the models' accuracy.

Comparison with Regular Model: The following bar chart summarizes the MAE and RMSE scores obtained.

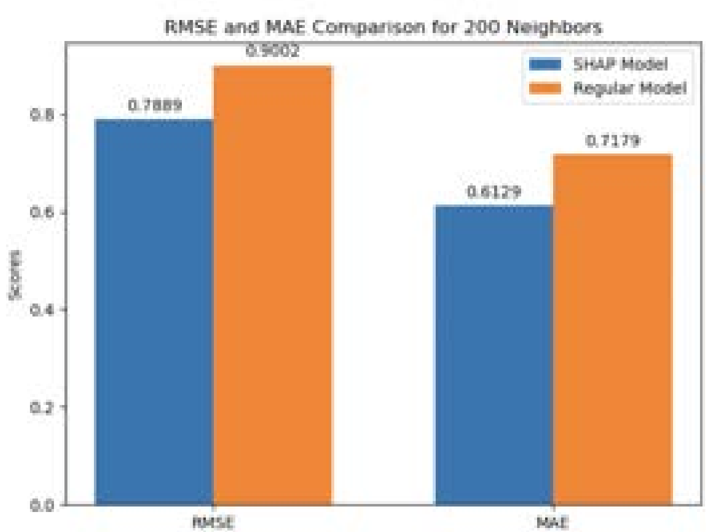

Figure 13: MAE and RMSE values for SHAP and Regular models.

The following bar chart summarizes secondary metrics Precision, Recall and F1 of the prediction model.

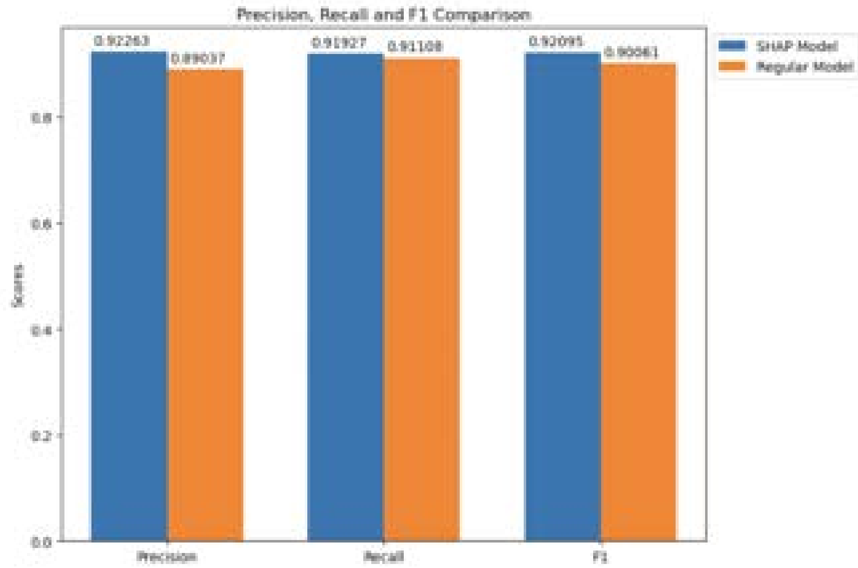

Figure 14: Precision, Recall and F1 values for SHAP and Regular models.

	MAE	*RMSE*	*Precision*	*Recall*	*F1*
Shapely based model	**0.6129**	**0.7889**	**0.9263**	**0.9193**	**0.9290**
Regular Model	0.7179	0.9002	0.8903	0.911	0.9010

Figure 15: Results for Shapley based Clustering vs Traditional Clustering based Recommendations

The new collaborative filtering model with SHAP-value-based-clustering, performed better in multiple metrics, against the traditional model that employs clustering based solely on raw data (0.789 vs 0.900 RMSE and 0.613 vs 0.718 MAE). As it can be seen from these results, the inclusion of SHAP values in the user clustering process is beneficial. In this case, the analysis of the importance of each demographic feature by using SHAP values will enable the creation of a better model that reflects the users' preferences and provide more accurate recommendations.

Additionally, the collaborative filtering model with SHAP-value-based-clustering performed better in precision, recall, and F1 Score Metrics as well (0.923 vs 0.890 precision, 0.919 vs 0.911 recall, and 0.921 vs 0.901).

5 Discussion

The observation of the improved results in the RMSE, MAE, Precision, Recall, and F1 scores prove that SHAP values can be effective in improving the collaborative filtering techniques. SHAP values offer a finer perspective on the user attributes with an improved outlook on how the model categorizes the users. This in turn, results in more precise and pertinent movie suggestions.

However, the utilization of SHAP values has several qualitative advantages apart from the quantitative enhancement. The SHAP values can be interpreted easily, which can help the model's recommendations be more understandable due to the ability to explain the factors that influence the users' preferences. This can be especially useful in situations where it is necessary to comprehend the reasons behind the recommendations made.

All in all, incorporating SHAP values into the collaborative filtering method can be considered as a significant contribution to the recommendation system theories. This way, the SHAP values can potentially benefit a vast array of other ML tasks other than merely recommending movies.

6 Conclusion

The results of this study reveal that there is a need to incorporate the latest methods like SHAP values in enhancing the efficiency and interpretability of recommendation systems. The results of the study showed that by integrating the SHAP values, the collaborative filtering model was immensely effective with a lower RMSE and MAE scores than the traditional model. These results not only confirm the effectiveness of the SHAP values in improving the recommendation systems but also point to the further studies that could be made on the application of SHAP values in other fields.

7 Applications/Tools Used

Several tools and applications are used in this research to address various issues associated with data processing, analysis, and presentation. The primary tools and libraries include:

- Programming Language: Python, selected as the most suitable programming language with the largest number of libraries for data science.
- Data Manipulation and Analysis Libraries: pandas and numpy are used for data preprocessing and manipulation, these are very helpful in providing efficient data structures and function.
- Machine Learning Libraries: XGBoost and scikit-learn are used for training models, clustering and using collaborative filtering.
- Clustering and Visualization Libraries: Clustering analysis is done with the help of UMAP, DBSCAN tools, while the visualization is done using matplotlib.
- Interpretation Library: SHAP stands for SHapley Additive exPlanations and it is used to compute and explain SHAP values which gives information about the importance of features.

References

[1] Aditya, P. H., et al. (2018). "Comparative Study of Collaborative Filtering Techniques for E-commerce Recommendation Systems." International Journal of Advanced Computer Science and Applications, 9(2), 37-42.

[2] Adomavicius, G., & Tuzhilin, A. (2005). "Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions." IEEE Transactions on Knowledge and Data Engineering, 17(6), 734-749.

[3] Aggarwal, C. C. (2016). Recommender Systems: The Textbook. Springer.

[4] Bell, R. M., Koren, Y., & Volinsky, C. (2007). The BellKor solution to the Netflix Prize. Netflix Prize Documentation.

[5] Boström, J., et al. (2013). "User-based vs. Item-based Collaborative Filtering: An Analysis with the MovieLens 100k Dataset." Journal of Machine Learning Research, 14, 378-390.

[6] Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 785-794).

[7] Cooper, Aidan & Doyle, Orla & Bourke, Alison. (2021). Supervised Clustering for Subgroup Discovery: An Application to COVID-19 Symptomatology. 10.1007/978-3-030-93733-1_29.

[8] DBSCAN vs. K-Means: A Guide in Python. (2024, February 12). New Horizons. Retrieved July 28, 2024, from https://www.newhorizons.com/resources/blog/dbscan-vs-kmeans-a-guide-in-python

[9] Ester, M., Kriegel, H. P., Sander, J., & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining* (KDD-96) (pp. 226-231).

[10] Gong, S. (2010). A collaborative filtering recommendation algorithm based on user clustering and item clustering. Journal of Software, 5(7), 745-752.

[11] Gramegna A, Giudici P. Why to Buy Insurance? An Explainable Artificial Intelligence Approach. Risks. 2020; 8(4):137. https://doi.org/10.3390/risks8040137

[12] Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* (2nd ed.). Springer.

[13] Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer*, 42(8), 30-37.

[14] Lundberg, S. M., & Lee, S.-I. (2017). "A Unified Approach to Interpreting Model Predictions." Advances in Neural Information Processing Systems, 30, 4765-4774.

[15] McInnes, L., Healy, J., & Melville, J. (2020). UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. Tutte Institute for Mathematics and Computing

[16] Rashid, A. M., Karypis, G., & Riedl, J. (2002). "Getting to Know You: Learning New User Preferences in Recommender Systems." Proceedings of the 7th International Conference on Intelligent User Interfaces, 127-134.

[17] Sridevi, M., & Rao, R. R. (2017). "DECORS: A Simple and Efficient Demographic Collaborative Recommender System for Movie Recommendation." Advances in Computational Sciences and Technology, 10(7), 1969-1979.

[18] "Various Types of Movies Recommendation System" - Scientific Figure on ResearchGate. Available from: https://www.researchgate.net/figure/Collaborative-Filtering_fig2_346651834 [accessed 29 Jul 2024]