AFKAdventurers: Recommendation Systems for Games on Steam

A Project Report

Presented to

DATA-240-11

Spring, 2024

By

Abdul Sohail Ahmed (016769610)

Bhavik Patel (016800550)

Poojan Gagrani (016795285)

Smeet Sheth (016786133)

May 16, 2024

Motivation

Steam is one of the most famous digital platforms that was developed by Valve Corporation in 2003; and is mainly used for publishing, downloading, and purchasing various video games. It contains a vast set of games ranging from AAA titles to independent games developed by new unknown developers. The main set of motivations of this project was to enhance user satisfaction by providing tailored recommendations based on user preference and playing history, improving engagement of users to the platform, increasing sales of the games that are recommended to the users, and supporting new incoming users by suggesting games that may closely align with their taste. These motivations helped us to develop a recommendation system that suggests games that will be preferred mostly by both new and existing users.

Background

According to the records, in 2022, the worth of the gaming industry surpassed the combined worth of the film and music industry by reaching a total cost of approximately $220 billion. One of the major reasons was the popularity of the Steam platform which was attracting a lot of people to buy games by providing a single comprehensive platform where users can navigate and download the games that they want. But, the major problem is the presence of abundant options in games that confuse users regarding which game to select over another. Most of the initially provided recommendation systems focus on the popularity and sales of the games and don't take into account user preferences. Lack of personalization in the process of game recommendations could result in a loss of interest and missing games that could have been preferred more by the users. To address this issue, the decision was made to create two recommendation systems, one based on user playing history that helps to gain an idea about user preferences to generate more personalized recommendations, and the second one for the new users to select the category and genre of their preference and recommend appropriate games.

Literature Review

Kang et al. (2017) talk about the importance of game reviews in the decision-making process of consumers. In this study, the community data is analyzed to determine factors influencing the rate of helpfulness of game reviews. The Open Source Game Review dataset is extracted from Steam and tested against two models namely CART and ANN. CART emerged as the most accurate model with an SSE of 1213.269. Cheuque et al. (2019) proposed a recommendation system based on user preferences where the three datasets consist of Australian users' purchase history, playtime, characteristics of games, and their social interactions. Three advanced models are used FM, DeepNN, and DeepFM where ALS is used as the baseline and DeepNN emerges as the best model in terms of MAP, Novelty, and Diversity. Lomanto et al. (2023) suggested the usage of Collaborative Filtering with SVD and Pearson Correlation to generate user-preferred recommendations. The dataset is taken from Kaggle which contains details of 12,393 users, and 185,673 games based on their interactions. SVD outperformed Pearson Correlation in terms of MAE, and RMSE. Batra et al. (2023) also proposed the usage of interaction between users and games to create a recommendation system where the dataset is taken from Steam Video Game and Bundle Data. The ALS model is trained in three different ways, one using playtime, the other using playtime and reviews, and the third using playtime and recommendations. Incorporating sentiments helped improve the model's performance. The proposed system is better in the way that it provides recommendations for both new and existing users using Collaborative filtering with BPR, Content-Based Filtering, Agglomerative, and DBSCAN Clustering. This enhances the user experience and provides more personalized suggestions even if the user is new.

Methodology

The creation of any model requires a lot of steps like data extraction, data storage, data cleaning, data transformation, exploratory data analysis, modeling, model evaluation, and generation of results. The same goes for the Steam Recommendation System. Initially, the data were retrieved from Steam Web API and steam spy API which was made easier by getting a thorough understanding of API methods using a tool called Postman. This helped to understand which methods are required for the problem and only the details of those methods were extracted using the Python scripts. The extracted data were stored in CSV files which were further processed by performing data cleaning and data transformation on training data which then fitted to validation and test dataset. The split was kept to be 80-10-10 for both of the models. After the transformed data was obtained exploratory data analysis was performed on it to get an understanding of the final data before it was provided for modeling. The training set was supplied to different models like Collaborative Filtering with Bayesian Personalized Ranking (BPR), Content-Based Filtering, Hierarchical Agglomerative Clustering, and DBSCAN Clustering. Finally, the models were evaluated on different metrics the ones using the user preference data (Collaborative Filtering with BPR and Content-Based Filtering) were evaluated on Precision, Recall, F1 score, and AUC whereas the ones using game preference data (Hierarchical and DBSCAN Clustering) were evaluated on Silhouette Score. The final step was the generation of recommendations in both approaches. The methodology followed in the project is shown in Figure 1.

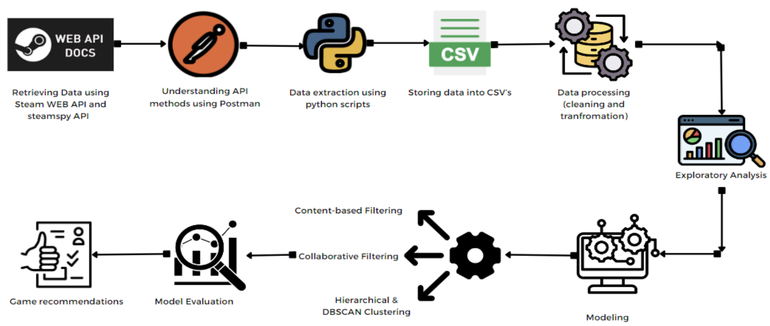

Fig.1. Methodology of the Project

User Preference Recommendation System

In this system, after the data was extracted and stored in CSV files, data processing was performed involving operations like column renaming, dropping irrelevant columns, filling missing values in country names, and correcting the ones that were misspelled or abbreviated. Then, an initial EDA was performed and it was observed that playtime was right-skewed which is one of the most important features in our recommendation system so it was decided to perform a log transformation to bring it closer to the normal distribution. The EDA-like distribution of ratings across games and top 10 games based on playtime was created to get an understanding of the final data. The modeling was performed using the training set of the final data. In this approach, two models were used: Collaborative Filtering with BPR and Content-based Filtering. Collaborative Filtering with BPR is useful when the requirement is to deal with the implicit feedback data like log transformed playtime in the current project. Collaborative Filtering makes the predictions of the interest of the user based on the information collected about the preferences of various users. It assumes that if the user has similar preferences currently then they are highly likely to select the same items (which is a game in this case) in the future. BPR is used in the project because the focus was to determine the pairwise ranking of items and not the ratings of each item and this was the reason the Collaborative Filtering was paired with BPR and used. BPR takes into consideration both interaction and non-interaction with items and places the items that were interacted with higher than the ones that were not. This made the combination of both useful as the goal was to determine top N recommendations. The model was trained on a different set of factors ranging from 10 to 100, a learning rate varying from 0.001 to 0.1, regularization varying from 0.01 to 1.0, and several iterations varying from 50 to 500 to determine the optimal set of parameters that will provide the best recommendations. Content-based Filtering is another model explored in this approach to understand if just considering the items that users have liked in the past to provide recommendations affects the quality of recommendations or not. This method focuses on creating user and item profiles and then performing cosine similarity to determine how closely an item matches the user's preferences. However, it lacks the concept of personalized recommendation as the focus is only to consider the items that users have liked in the past and not the implicit feedback factors like playtime. In this model parameters like random state of 42, feature scaling of Min-Max, categorical encoding of One-Hot, and similarity metrics of Cosine are used. Figure 2 illustrates the model workflow in this approach.

Game Preference Recommendation System

The game preference dataset consisted of 21 different columns and 837 instances. As all the columns were not important for the modeling, unnecessary columns were dropped resulting in the final 10 columns. To prepare data, all the null values and duplicates were checked. Numerical columns 'average_forever', 'average_2weeks', 'median_forever', 'median_2weeks', 'ccu', 'price', 'ratings', 'owners_average', 'MutualPlayerCount' were transformed and scaled. Categorical columns 'genre' and 'categories' were encoded using one hot encoding. Distributions for different numerical columns were checked and transformations were performed based on the values on the training set. The 'ccu' column had multiple '0's, so the first log transformation was applied. As the resultant transformation was not satisfactory, Box-Cox transformation was applied which gave somewhat better results and was able to transform the data. Finally, Quantile transformation was applied to the column which gave a normal distribution with multiple extreme outliers.

As our model is user-centric, it will first ask for input from the user 'Enter the Category you like' and 'Enter the Genre you like'. The dataset is filtered into three main categories i.e. Single Player, Multi-Player, or both, and 23 different unique genres. Once the input from the user is collected it is mapped to the main dataframe. The instances which align with the user's input are filtered out. For hierarchical clustering, clusters are made by using the Euclidean metric and Ward's linkage method which helped minimize the cluster variance and resulted in the formation of well-defined clusters. The range of clusters was given from 2-15 and the optimal number of clusters was decided based on the highest silhouette score. A similarity score based on the 'average_forever_qt_normal', 'ccu_qt_normal', and 'ratings_qt_normal' columns would be assigned to every game in the chosen cluster using Cosine Similarity. Similarity score is assigned by comparing the features of the game with the user's input. The resulting games are sorted based on their scores and recommendations are generated based on the games having the highest similarity scores out of which top 5 games are displayed as an output.

For DBSCAN all the preprocessing part was the same as done in hierarchical clustering. Min sample range was given from 3-10. This range would decide the number of neighboring points for a point to be considered as a core point while creating the clusters. The Epsilon range was from 0.5 - 0.95. This range is used to determine the maximum distance between two samples for them to be considered in the same neighborhood. The best parameter is chosen based on the value of the silhouette score for every combination to know how accurately the given data point fits into the assigned cluster as compared to the other points. The similarity score was calculated using Cosine Similarity and the recommendations were generated in a similar way to that of Hierarchical clustering.

Experiments and Results

User Preference Recommendation System

The user preference data is used to create a recommendation system using two models but there is a need to determine if the recommendations created are user-personalized or not. To determine the first model Collaborative Filtering with BPR which was trained with a number of latent factors as 50, learning rate as 0.01, L2 regularization as 0.1, and number of iterations as 100 was tested on the test data to determine the Precision, Recall and F1 score and finally determine the recommendations created. Some of the evaluation metrics values were like Precision@10 was 0.089, Recall@10 was 0.298, and ROC Curve value was 0.16 but the main issue was the recommendations that were getting generated having the pattern of repetitions that is the games that were getting recommended had the same names for different users this means that the recommendations were not user-specific rather it was more general. The recommendations obtained from this system are provided in Figure 2.

```
Recommended Games for User ID 76561198897560804: ['Counter-Strike: Global Offensive', 'Left 4 Dead 2', 'Portal 2', "Garry's Mod", 'Half-Life 2']
Recommended Games for User ID 76561199481524497: ['Counter-Strike: Global Offensive', 'Left 4 Dead 2', 'Portal 2', 'Half-Life 2', "Garry's Mod"]
Recommended Games for User ID 76561198015413217: ['Counter-Strike: Global Offensive', 'Left 4 Dead 2', 'Portal 2', 'Half-Life 2', "Garry's Mod"]
Recommended Games for User ID 76561198015685843: ['Counter-Strike: Global Offensive', 'Left 4 Dead 2', 'Portal 2', 'Half-Life 2', "Garry's Mod"]
Recommended Games for User ID 76561198020809111: ['Counter-Strike: Global Offensive', 'Left 4 Dead 2', 'Portal 2', 'Half-Life 2', "Garry's Mod"]
```

Fig.2. Generated Recommendations from Initial Parameters (First)

To create more user-specific recommendations it was decided to perform a grid search to determine the best set of parameters on which the model should be trained. The best parameters were found to be a number of latent factors at 50, the learning rate at 0.01, L2 regularization at 0.1, and the number of iterations at 50. The model trained on these parameters and it was found that it obtained the scores of 0.081, 0.271, 0.125, and 0.217 against evaluation metrics Precision@10, Recall@10, F1 Score@10, and ROC curve value respectively. Figure 2 illustrates the Precision, Recall, and F1 score values obtained against different K and ROC curves

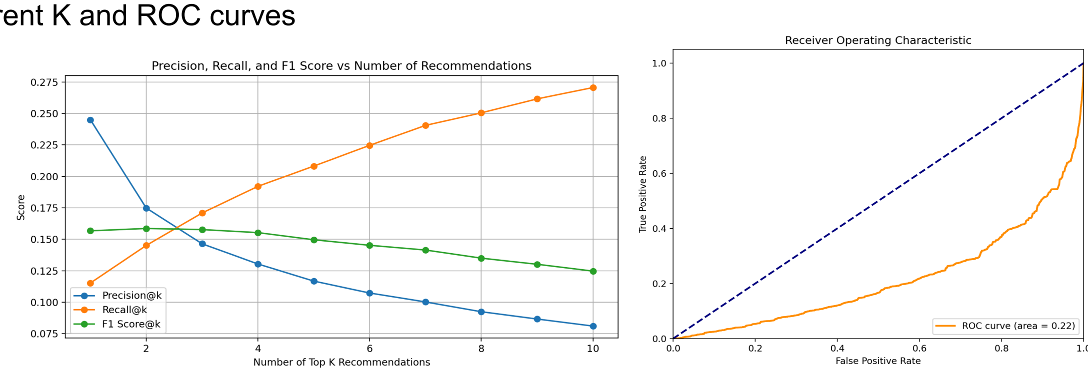

Fig.2. Plot of model performance against different Evaluation Metrics (First)

This model was then used to create recommendations for each user and it was found that the recommendations that were getting generated were more user-specific. Figure 3 shows the generated recommendations.

```
Recommended Games for User ID 76561198897560804: ['Counter-Strike: Global Offensive', 'Portal 2', 'Saints Row: The Third', 'Tomb Raider', 'Middle-earth: Shadow of War']

Recommended Games for User ID 76561199481524497: ['Counter-Strike: Global Offensive', 'Terraria', 'Portal 2', 'Left 4 Dead 2', 'Borderlands 2']

Recommended Games for User ID 76561198015413217: ['Counter-Strike: Global Offensive', 'Terraria', 'Portal 2', "Garry's Mod", 'Tomb Raider']

Recommended Games for User ID 76561198015685843: ['Counter-Strike: Global Offensive', 'Tomb Raider', 'Middle-earth: Shadow of War', "Don't Starve Together", 'Batman: Arkham Cit
y - Game of the Year Edition']

Recommended Games for User ID 76561198020809111: ['Counter-Strike: Global Offensive', 'Terraria', 'Left 4 Dead 2', 'Borderlands 2', 'Saints Row: The Third']
```

Fig.3. Generated Recommendations from Best set of Parameters (First)

Only the best-performing model was included in the final model to ensure that this model can generate effective user-specific recommendations.

The second model Content-Based Filtering was trained on user and item profiles and then a cosine similarity matrix was created between them to determine user-item interactions. Then the value of ‘N’ in top N recommendations is provided where the top N values were selected from the cosine similarity matrix and the model was evaluated against evaluation metrics Precision@K, Recall@K, F1 score@K, and AUC@K values. The values obtained were 0.017 (K=10), 0.003 (K=10), 0.005 (K=10), 0.58 (K=5) (optimal). Figure 7 shows the values obtained against different evaluation metrics. The number of recommendations were selected to be generated as 5 because the AUC curve value is maximum other than what it had at K=1 at only this value and after this, the value decreases and after K=10 it stabilizes around 0.54. K=1 not selected because the generated recommendation number would be very low and not that useful.

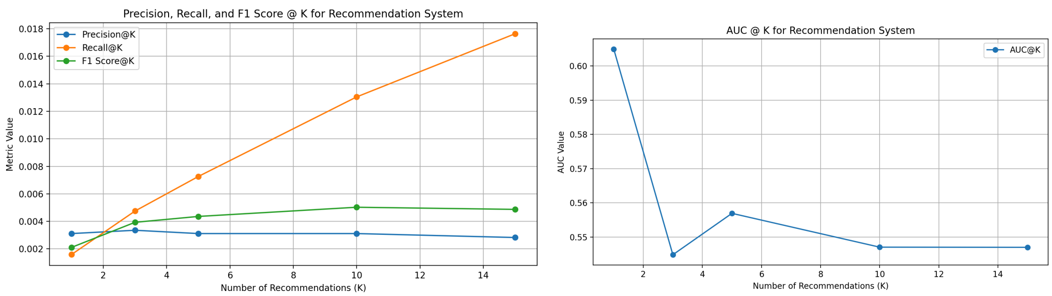

Fig.4. Plot of model performance against different Evaluation Metrics (Second)

With selected K value the recommendations are generated from the second model which can be seen in Figure 5.

```
User ID 76561198259264314 Recommendations: ['Filthy Hands', 'FarSky', 'The Clockwork Man']
User ID 76561198076737490 Recommendations: ['OH MY GOD, LOOK AT THIS KNIGHT', 'Orcs Must Die! 2', 'The Talos Principle 2']
User ID 76561198260424824 Recommendations: ['Dungeon Siege III', 'Lonely Mountains: Downhill', 'Metro Exodus']
User ID 76561198058454559 Recommendations: ['Hellsplit: Arena', 'Toribash', 'Tetris Effect: Connected']
User ID 76561197979019316 Recommendations: ['Zero Caliber VR', 'Monster Loves You!', 'Peglin']
```

Fig.5. Generated Recommendations (Second)

Game Preference Recommendation System

With a silhouette score of 0.439, which indicates a reasonably strong structure within the clusters, three were found to be the ideal number of clusters for hierarchical clustering. According to this metric, games inside a cluster are more comparable to one another than they are to games outside of it. The model's capacity to generalize over previously untested data is supported by the silhouette scores of 0.392 and 0.431 for the validation and test datasets, respectively. On the other hand, the DBSCAN algorithm produced the best silhouette score of 0.4229416337361661, which was optimized for an epsilon value of 0.9000000000000002 and a minimum sample size of 5. Even while this score is not as high as Hierarchical Clustering, it still shows that dense areas in the game data are fairly separated. On the test and validation datasets, however, silhouette scores drastically decreased to 0.12611428126902907 and 0.18467174416374785, respectively. These findings imply that the DBSCAN model might not have generalized as well as the Hierarchical model, possibly as a result of its sensitivity to the distribution of density in the data and parameter choices. Furthermore, 14 groups were found using the DBSCAN algorithm (not including noise), suggesting a more precise clustering that may be more responsive to particular game features than the more general classifications generated by Hierarchical Clustering. Comparing the results it was found that Hierarchical Clustering performed better than DBSCAN given our use case and was able to generate better, more reliable, and accurate results

Discussion

The project presents the creation of two recommendation systems. One is based on user preference where it uses the concept of Collaborative Filtering with BPR and Content-Based Filtering to provide a recommendation system that provides more personalized recommendations that help to improve user experience and engagement in the platform. Second is the creation of a recommendation system using Agglomerative and DBSCAN Clustering which helps to create more cohesive clusters that allow new incoming users to the platform to provide categories and genres of the games they usually like to play and the system will generate the most appropriate games that users might like. This will allow the new users to feel welcome and also the recommendations generated are more user-focused rather than some random generations. This clustering-based recommendation system will also allow the existing users to explore new games by providing the asked details like category and genre and get the recommendations of the most appropriate games specific to those details.

Future Improvements

In the future, the plan is to optimize the parameters of the used models by either performing experimentation with different sets of hyperparameters or extensive fine-tuning to enhance the performance of the system. Bringing in more advanced models like Artificial Neural Network (ANN), Deep Neural Network (DNN), etc to improve the recommendations by capturing more complex user behaviors and preferences. Integrating in-game behavior, community, and social media platform data to provide more context and deep insights about the data to provide more user-preferred recommendations. Targeted marketing could also be incorporated based on the user gaming preferences so that user engagement with that game can be improved. The concept of the feedback mechanism could also be included so that the idea about the correctness of the recommendations can be obtained so that the system can be modified in such a way that it provides better recommendations that are more suitable to the user's needs.

References

Batra, S., Sharma, V., Sun, Y., Wang, X., & Wang, Y. (2023). Steam Recommendation System. https://doi.org/10.48550/arXiv.2305.04890

Cheuque, G., Guzmán, J. L., & Parra, D. (2019). Recommender Systems for Online Video Game Platforms: the Case of STEAM. *WWW '19: Companion Proceedings of the 2019 World Wide Web Conference*. https://doi.org/10.1145/3308560.3316457

Kang, H. N., Yong, H. R., & Hwang, H. S. (2017). A study of analyzing on online game reviews using a data mining approach: Steam community data. *International Journal of Innovation, Management and Technology, 8*(2), 90-94. https://doi.org/10.18178/ijimt.2017.8.2.709

Lomanto, W. E., Andrian, V., Achmad, S., & Sutoyo, R. (2023). Collaborative Filtering for Steam Games Recommendation. *International Conference on Cybernetics and Intelligent System.* https://doi.org/10.1109/icoris60118.2023.10352243

Abbreviations

- CART - Classification and Regression Tree
- ANN - Artificial Neural Network
- SSE - Sum of Squared Errors
- ALS - Alternating Least Squares
- MAP - Mean Average Precision
- SVD - Singular Value Decomposition
- MAE - Mean Absolute Error
- RMSE - Root Mean Squared Error
- DBSCAN - Density-Based Spatial Clustering of Applications with Noise