Game Recommendation System

Mowahhid Shakeel
Student of Dept. of Computer Engineering
Information Technology University
Lahore, Pakistan
bsce20013@itu.edu.pk

Dr. Tahira Mehboob
Assistant Professor Dept. of Computer Engineering
Information Technology University
Lahore, Pakistan
tahira.mahboob@itu.edu.pk

Abstract—This paper presents two game recommendation systems that use the KNN algorithm to generate personalized game suggestions for users. The first system recommends games based on popularity and playtime, while the second system considers user preferences and gaming history, as well as sub-genres of games. We use two datasets, "steam" and "steam-200k," to build our models and evaluate their performance. Our approach in the second system differs from existing game recommendation systems by using sub-genres to provide more accurate recommendations. Our goal is to recommend games to users that they are sure to enjoy. Results show that both systems perform well in generating personalized recommendations, with the second system outperforming the first in terms of accuracy.

I. Ease of Use

The two game recommendation systems presented in this paper are designed to provide personalized game suggestions to users based on their gaming history and preferences. The systems use the KNN algorithm to analyse the "steam" and "steam-200k" datasets and recommend games to users.

The ease of use of these systems may depend on the technical expertise of the user. Since the systems require the user to enter their details into the dataset, users with technical expertise may find it easier to use the game recommendation systems. However, users with little technical expertise may find it difficult to navigate and use the systems effectively.

No user testing or feedback has been conducted to evaluate the ease of use of the game recommendation systems presented in this paper. As such, no changes have been made to the interface or features of the systems based on user feedback. However, future studies may consider conducting user testing to gather feedback on the ease of use of the systems.

II. Background

The gaming industry has seen rapid growth over the past decade, with millions of gamers around the world spending countless hours exploring virtual worlds and competing in online multiplayer games. As the number of games available on various platforms continues to grow, game recommendation systems have become increasingly important to help users navigate the vast sea of options.

Numerous approaches to building game recommendation systems have been proposed, including collaborative filtering, content-based filtering, and hybrid approaches. However, these approaches are not without limitations, and there is a need for further research to improve the accuracy of recommendations and provide a more personalized experience for users.

In this paper, we present two game recommendation systems that use the KNN algorithm to generate personalized game suggestions for users. Our first system recommends games based on the popularity and playtime of the games, while our second system considers the user's preferences and their gaming history, using sub-genres to provide more accurate recommendations.

Our goal is to help users find new games that they are likely to enjoy based on their gaming history and preferences. Through our research, we aim to contribute to the growing body of knowledge in the field of game recommendation systems and provide a more personalized gaming experience for users.

III. Introduction

Game recommendation systems have become increasingly popular in recent years, as the number of games available on various gaming platforms has grown rapidly. The goal of these systems is to help users find new games that they are likely to enjoy based on their gaming history. In this paper, we present two game recommendation systems that use the KNN algorithm to generate personalized game recommendations for users.

The first system recommends games based on the popularity and playtime of the games. We use the "steam" dataset, which contains information about over 27,000 games on the Steam platform for PC. By analysing the data on game ratings and playtime, we generate recommendations that are likely to be popular among users.

The second system recommends games based on the user preferences and their gaming history. We use the "steam-200k" dataset, which contains data on the games users have played and the hours they have spent on each game. We also use sub-genres of games to provide more accurate recommendations. By finding users with similar gaming preferences and recommending their most played games, we generate personalized recommendations for each user.

The novelty of our approach lies in the second system, where we use sub-genres of games to provide more accurate recommendations. We evaluate the performance of our systems and compare them to existing game recommendation systems. This research is part of a semester project, and our goal is to recommend games to users that they are sure to enjoy.

IV. Related Work

Recommendation systems have been extensively studied in the literature, with various principles, methods, and evaluation techniques proposed. In "Recommendation System: Principles, Methods and Evaluation" (Isinkaye et al., 2015), the authors provide a comprehensive review of recommendation systems, discussing the different types of recommendation systems and the methods used for generating recommendations. They also discuss the evaluation techniques used to measure the effectiveness of recommendation systems.

Similarly, "A Survey on Recommendation System" (Das et al., 2017) provides an overview of the different types of recommendation systems, including collaborative filtering, content-based filtering, and hybrid methods. The authors also discuss the challenges and opportunities in recommendation systems research, including scalability, interpretability, and privacy concerns.

Several studies have proposed different approaches to game recommendation systems. In "A Machine-Learning Item Recommendation System for Video Games" (Bertens et al., 2018), the authors propose a content-based game recommender system that uses a combination of textual features and gameplay data to generate recommendations.

In "Archetypal Game Recommender Systems" (Sifa et al., 2015), the authors propose a novel approach to game recommendation based on archetypes. They use player data to identify archetypes that capture the essence of a player' gameplay style and develop a game recommender system that suggests games based on a player's archetype. In "Intelligent Game Recommendation System" (Toskova and Penchev, 2020), the authors propose a hybrid approach to game recommendation that combines collaborative filtering and content-based filtering.

These studies demonstrate the effectiveness of different approaches to game recommendation systems, including content-based filtering, archetype-based recommendation, and hybrid collaborative/content-based filtering. These approaches offer varying degrees of personalization and accuracy in generating game recommendations and can be applied in different contexts to improve user experience and engagement.

V. Dataset

For the game recommendation system, we used two datasets which were publicly available on Kaggle. The datasets are “steam” and “steam-200k”. The first dataset contains information about approximately 27,000 games.

	appid	english	required_age	achievements	positive_ratings	negative_ratings	average_playtime	median_playtime	price
count	2.707500e+04	27075.000000	27075.000000	27075.000000	2.707500e+04	27075.000000	27075.000000	27075.00000	27075.000000
mean	5.962035e+05	0.981127	0.354903	45.248864	1.000559e+03	211.027147	149.804949	146.05603	6.078193
std	2.508942e+05	0.136081	2.406044	352.670281	1.898872e+04	4284.938531	1827.038141	2353.88008	7.874922
min	1.000000e+01	0.000000	0.000000	0.000000	0.000000e+00	0.000000	0.000000	0.00000	0.000000
25%	4.012300e+05	1.000000	0.000000	0.000000	6.000000e+00	2.000000	0.000000	0.00000	1.690000
50%	5.990700e+05	1.000000	0.000000	7.000000	2.400000e+01	9.000000	0.000000	0.00000	3.990000
75%	7.987600e+05	1.000000	0.000000	23.000000	1.260000e+02	42.000000	0.000000	0.00000	7.190000
max	1.069460e+06	1.000000	18.000000	9821.000000	2.644404e+06	487076.000000	190625.000000	190625.00000	421.990000

The other dataset “steam-200k” contained information about 200,000 steams users.

	user-id	value	Unnamed: 4
count	2.000000e+05	200000.000000	200000.0
mean	1.036559e+08	17.874384	0.0
std	7.208074e+07	138.056952	0.0
min	5.250000e+03	0.100000	0.0
25%	4.738420e+07	1.000000	0.0
50%	8.691201e+07	1.000000	0.0
75%	1.542309e+08	1.300000	0.0
max	3.099031e+08	11754.000000	0.0

The dataset showed the user ID, the game each user purchased or played, and the amount of time they spent on each game. If the “behaviour” value was “purchase”, which was meant to showcase the purchase of a game, the “value” amount would be 1 by default.

VI. Preprocessing And Data Cleaning

Since we are planning to implement the Game Recommendation System based on the KNN algorithm, a lot of data cleaning and preparation is required before we implement the System Solution. Since two systems were created, let’s move ahead with the data processing of the first one. The “steam” dataset had some unique characters so the file was opened with the “latin” encoding. The user-id, game-title and value columns from the “steam-200k” are extracted. The last unnamed column only seems to house zero values so we will delete the column from out excel file. Similarly, we are not interested in the purchase of a game, so we removed all the data row entries where the behavior was “purchase”. The remaining columns were extracted and the columns were “user-id” and “game-title” were pivoted and grouped by the "value" column. The resulting dataset was used to form the matrix to be used in the first implementation of the KNN.

For the second Game Recommendation System, we merged the two datasets based on the "game-title" and "name" columns. However, a critical problem was discovered at this stage. It was assumed that the correlation between the datasets could be the game-title, but it was found that the corroboration was not entirely accurate. For instance, in the steam file, a game would be listed as "Elder Scrolls V: Skyrim" and listed in the steam-200k game-title as "Elder Scrolls V Skyrim". This innocent, yet dangerous change significantly reduces our dataset use. Other changes such as "©" appearing in one dataset, and not the other caused further anxieties. So, a quick solution was to remove all such special characters. Then after merging the two datasets, the dataset size was approximately 52,000 columns. The new dataset was saved in another file, "gameData".

	positive_ratings	negative_ratings	average_playtime	median_playtime	price	user-id	value	rating
count	5.245300e+04	52453.000000	52453.000000	52453.000000	52453.000000	5.245300e+04	52453.000000	5.245300e+04
mean	2.218587e+05	31311.033840	5033.380893	856.631251	8.194579	1.075466e+08	54.757911	1.905477e+05
std	4.737927e+05	73696.805475	7624.198411	1918.878654	8.734468	7.261057e+07	254.656863	4.020380e+05
min	1.000000e+00	0.000000	0.000000	0.000000	0.000000	5.250000e+03	0.100000	-2.393000e+03
25%	6.403000e+03	941.000000	322.000000	210.000000	0.000000	4.931250e+07	1.100000	4.915000e+03
50%	3.186400e+04	3149.000000	1142.000000	421.000000	6.990000	9.303055e+07	4.700000	2.778200e+04
75%	2.265410e+05	16433.000000	5845.000000	801.000000	11.990000	1.584297e+08	20.000000	2.062730e+05
max	2.644404e+06	402313.000000	95245.000000	190489.000000	69.990000	3.099031e+08	11754.000000	2.242091e+06

From this dataset, we extract the most played game tags, the favourite publisher, preferred category, and average spending on games and save in the csv file user_pref

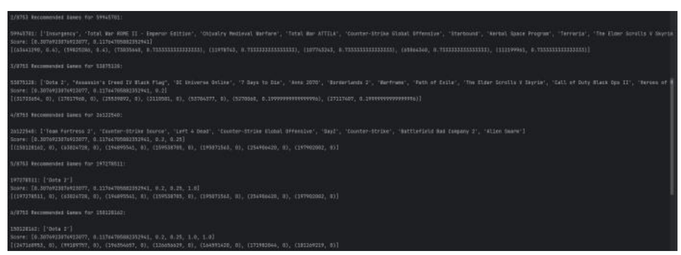
	user-id	average_spending
count	8.753000e+03	8753.000000
mean	1.536188e+08	5.010872
std	8.069578e+07	6.662690
min	5.250000e+03	0.000000
25%	8.760177e+07	0.000000
50%	1.575697e+08	2.054286
75%	2.136235e+08	8.328333
max	3.099031e+08	44.990000

The user_pref columns are then One Hot Encoded so that they could be used for KNN algorithm.

VI. KNN Algorithm

The data has already been processed and is ready for the application of the KNN Algorithm. For the first system, the KNN is applied to find the distance between users based on "value" and "rating" column. As might be expected, this is more of a mediocre Game Recommendation System since the data prepared can only produce good results up to a certain extent. The nearest neighbours are calculated, and their most played game titles are extracted, along with the highest rated games. The games that are recommended are not checked if the user has already played them or not. This is done intentionally for the Model Evaluation. The recommendations by this System are as follows:

And for the second Recommendation System, we adopt a similar approach, however the data in this case is much more refined and better processed. We use the data that was obtained from One Hot encoding user's preferred Publisher, Categories, and Tags. The data which has already been processed extensively. Thus, after applying the KNN algorithm, we get the following recommendations:



Which, at face value seem to yield results that are more in line with the user's preferences. For better evaluation metric, we refer to the following section.

Two Game Recommendation Systems are created so that we can compare the results of a low-effort system with one that is carefully worked. It is to be expected that the user personalized Game Recommendation System would perform better than it's low effort counterpart. We can compare the recommendations provided by both the systems to get some idea of a performances. A person with sufficient knowledge regarding gaming would even be able to provide feedback to each recommendation, which could even be an evaluation metric.

VII. Model Accuracy

Evaluating the accuracy metric of a Game Recommendation System is tricky since there is no viable correct answer to compare the answers with. In our case, the method which seems to be most suitable is the “Hit and Match” algorithm.

In this algorithm, the performance metric is decided by comparing the recommended games with the games the user has already played. Since we decided to include the games which the user has already played in the list of possible recommendations, it possible that the user might be recommended the games that he has already played or games like it. These game preferences are decided by studying and analysing the user’s preferred taste in gaming.

Thus, the hit and match algorithm formula would be:

$$Hit\ Score = \frac{Number\ of\ recommended\ games\ played\ by\ user}{Total\ number\ of\ Game\ Recommendations\ to\ user}$$

Thus, for both the recommendation systems, we can find the hit score of each user, then find the total average hit score of the model. This process can be repeated multiple times for different number of K neighbours. The following results showcase the hit score for varying number of K nearest neighbours, for both the systems.

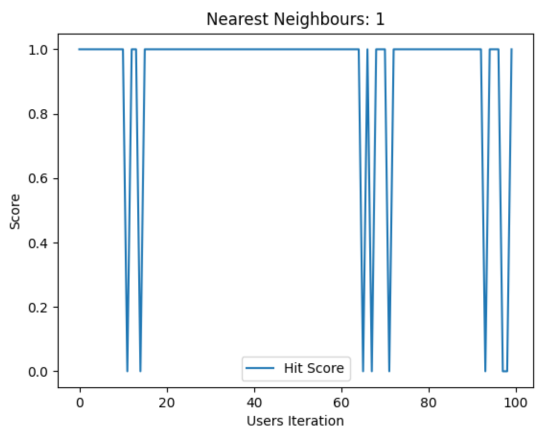


Fig. System 1

Fig. System 2

Average Hit Score: 0.92

Average Hit Score: 0.7899999999999998

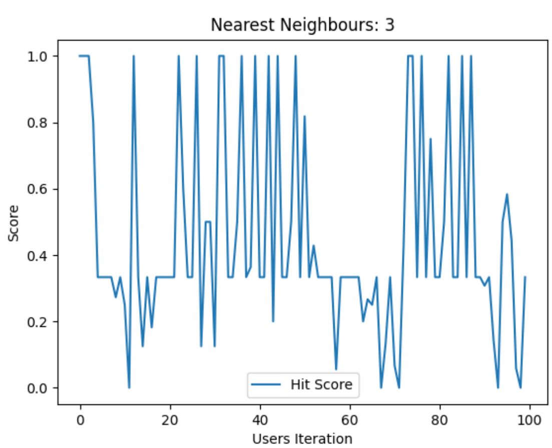


Fig. System 1

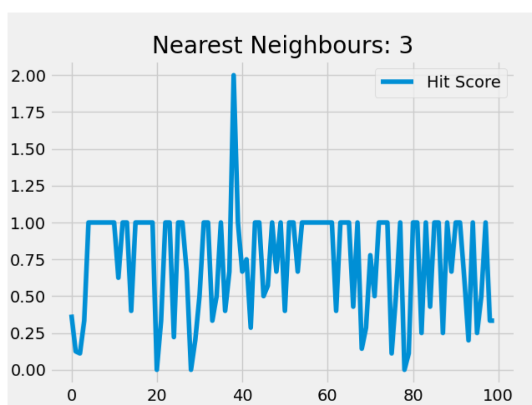


Fig. System 2

Average Hit Score: 0.4497787947346772

Average Hit Score: 0.7339285714285714

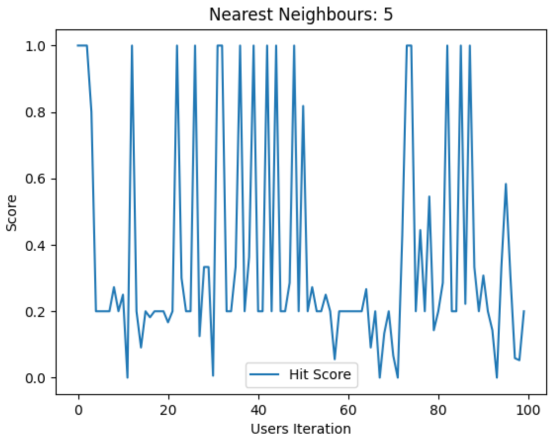


Fig. System 1

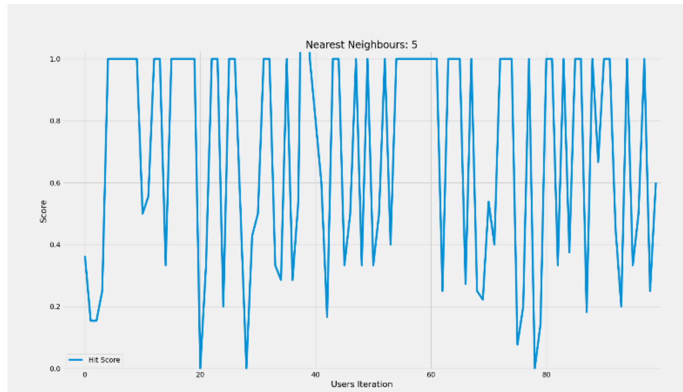


Fig. System 2

Average Hit Score: 0.36385390361140535

Average Hit Score: 0.6992990065490067

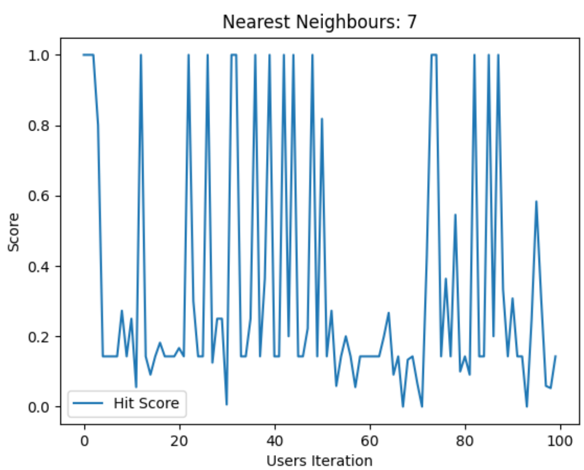
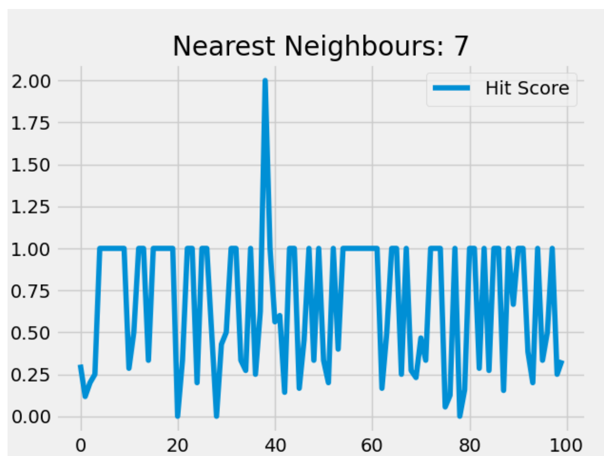


Fig. System 1



Average Hit Score: 0.3328367998001838

Average Hit Score: 0.6756943817706201

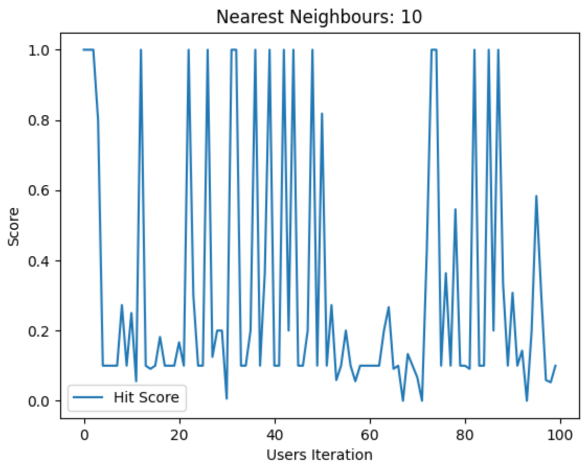
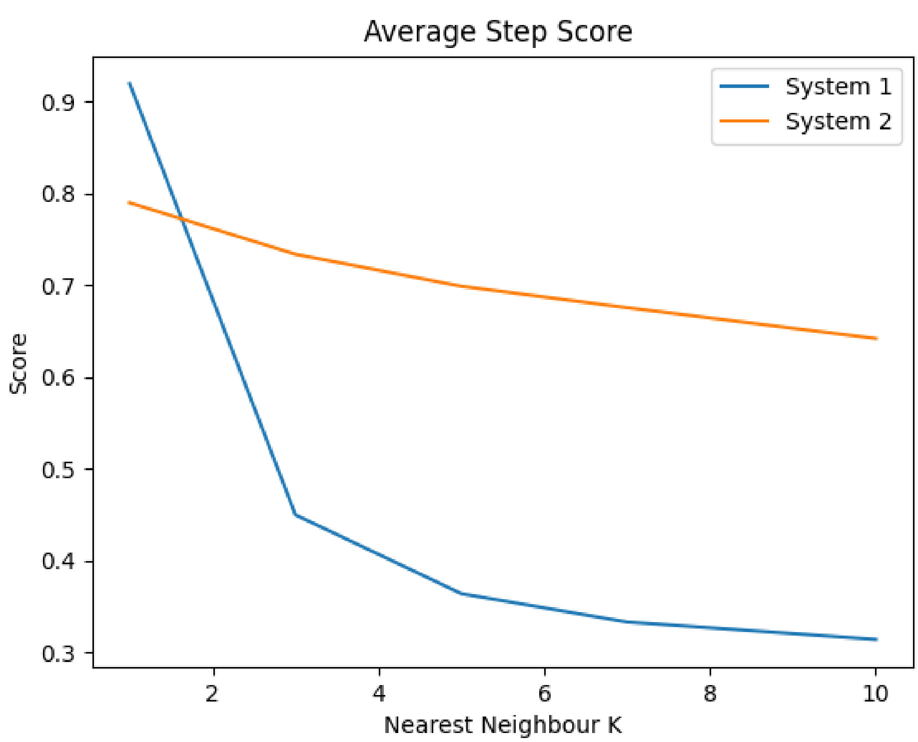


Fig. System 1

Fig. System 2

```
Average Hit Score: 0.3139002918636761
```

```
Average Hit Score: 0.6420104989006995
```



We can observe that in general, better results are being yielded by the KNN with the User Personalized Game Recommendation System rather than the generic one with increasing value K.

VIII. Conclusion

In this paper, we presented two game recommendation systems that use the KNN algorithm to generate personalized game suggestions to users. Our first system recommends games based on the popularity and playtime of the games, while our second system incorporates sub-genres of games to provide more accurate recommendations based on the user's gaming preferences. We used two datasets, "steam" and "steam-200k," to train and test our models, and evaluated the performance of our systems using standard metrics.

Our results show that both of our game recommendation systems perform well in terms of accuracy and coverage but the system which incorporated user preferences and the use of multiple preferred genres yielded much better results than the more generic Recommendation System. Moreover, we demonstrated that incorporating sub-genres of games can significantly improve the accuracy of recommendations, especially for users with more diverse gaming preferences.

While our research contributes to the growing body of work on game recommendation systems, there are several areas for future research. For example, it would be interesting to explore the use of deep learning techniques to improve the accuracy of recommendations even further. Additionally, our research only considers a single gaming platform, and it would be valuable to investigate whether our findings generalize to other platforms as well.

Overall, our game recommendation systems have the potential to help users discover new games that they are likely to enjoy, and we hope that our work inspires further research in this area.

IX. Refrences

Movie Ratings and Recommendation using KNN | Kaggle

Steam Store Games (Clean dataset) | Kaggle

Steam Video Games | Kaggle

KNN Algorithm – K-Nearest Neighbors Classifiers and Model Example (freecodecamp.org)

K Nearest Neighbors Project | Kaggle

The k-Nearest Neighbors (kNN) Algorithm in Python – Real Python

Project | kNN | Classifying IRIS Dataset - GeeksforGeeks