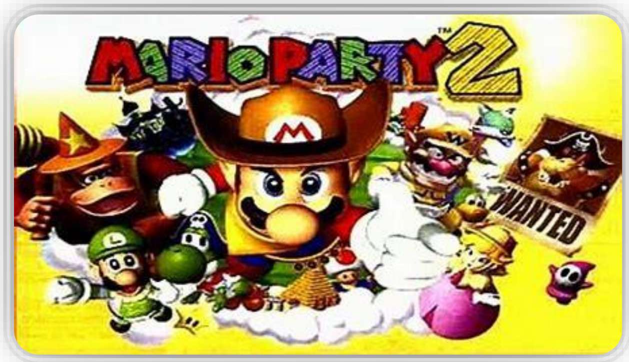
Recommendation System for Video Games

By Devesh Gokalgandhi

Currently, approximately 2.2 billion gamers worldwide exist (a third of the world's population!), with that number expected to rise to 2.7 by 2021. Often then not, a common issue that arises for gamers, with and without a rich gaming history, is deciding what to play next. Perhaps we want to enjoy games most similar to our previously played games, but how can we comprise a list of games to achieve that. Maybe be can look at other users' histories with games, and decipher whether we would also enjoy games played by these other users.



Just like with TV and movies, games exhibit qualities and concepts that databases have taken note of and stored, as well as, other user's ratings and reviews for these games, which can be utilized to decide which game to play next. Recommendation systems opt to alleviate this issue of deciding which games to play, by providing a list of games that the user will most likely enjoy based on previously played games and the history of similar users. The recommendation system is built on two main components: content based filtering and collaborative based filtering.

The idea behind content-based filtering is that the system will recommend games, who content is most like the target game. For example, if I enjoy the game Halo, I would be recommended Halo 2, because it shares the same characters, genre, concepts, etc., or I would be recommended Destiny and Destiny 2, because they share the same developer, Bungie. This filtering method relies on the NLP technique, TF-IDF, in which keywords and properties of the game are taken into consideration.

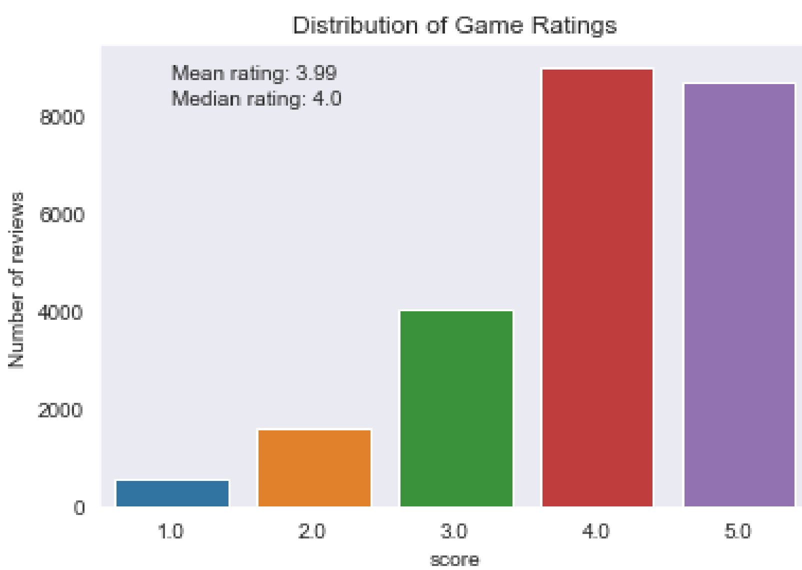
The idea behind collaborative-based filtering is that the system will predict ratings for unrated games based on the taste of similar users. For example, if I enjoy the game Halo (rating it 5 star), and you enjoy the games Halo and Mario Party (rating them a 5 star), I will most likely enjoy Mario Party. The filtering method will rely of model-based approaches, such as clustering algorithms and matrix factorization algorithms, to predict ratings for unrated games.

The Data

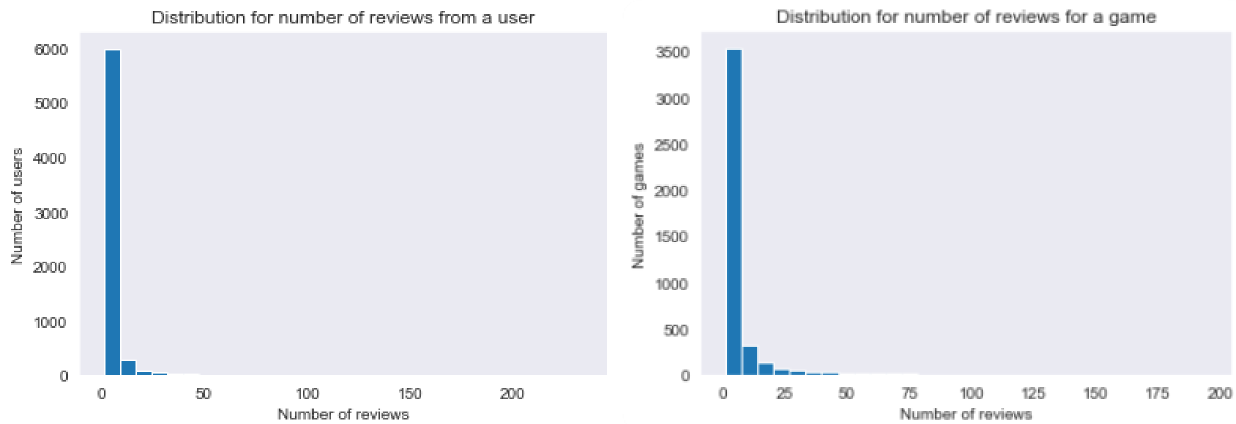
24023 game reviews were extracted from the API provided by the largest gaming database online, GiantBomb.

User ratings, on a scale of 0-5, are given from a total of 6561 users to 4223 games, ranging from July 2008 to September 2019. Database also provided game features: genres, themes, concepts, platforms, developers, publishers, developers, short and long game description. Luckily, an additional feature, “similar-games”, provides a list of games that GiantBomb deemed similar to the target game. This feature will be used to evaluate how the content-based filtering system recommends games.

Positivity and Sparsity



Looking at the distribution of game ratings for the entire dataset shows that most games are rated highly positive, with a lot of 4’s and 5’s. This could be a result of response bias or unwillingness to rate negatively, in which gamers may feel pressured to rate more popular games higher or show preferences towards their favorite games. This distribution will go on to affect the collaborative based filtering system, in which the system will tend to overpredict for negatively rated game reviews.

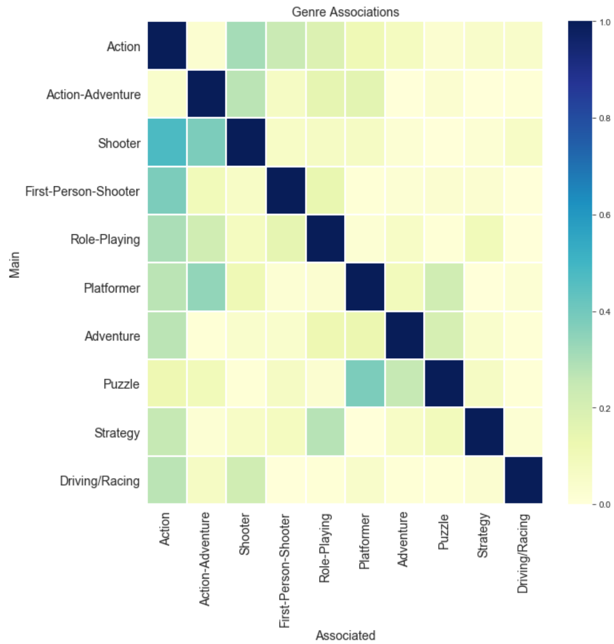


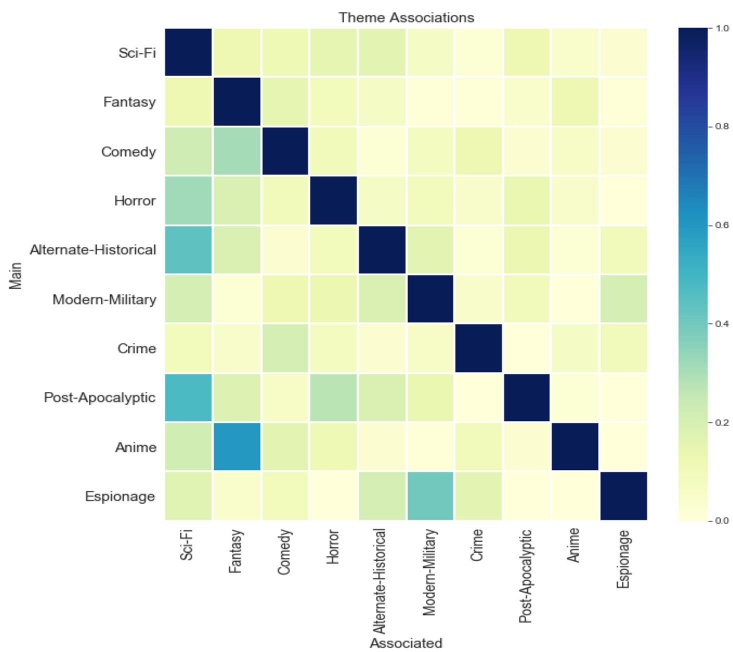
Looking at the distribution of the number of reviews per user, and per game, clearly this dataset is sparse. This is expected as not every gamer has played every game, and many users have only reviewed one game, and many games have only been reviewed only once (long tails in the plots). Knowing this:

1. We can expect, with the recommendations, that more popular games will tend to be recommended more, whereas games from the ‘long tail’ section might get ignored.
2. When querying a new video game to the system, the ‘cold start’ problem may occur, where unless a user has rated these new games, they won’t get recommended. The sparsity of the data adds on to that issue.

Action packed, Sci-Fi filled, Achievements unlocked

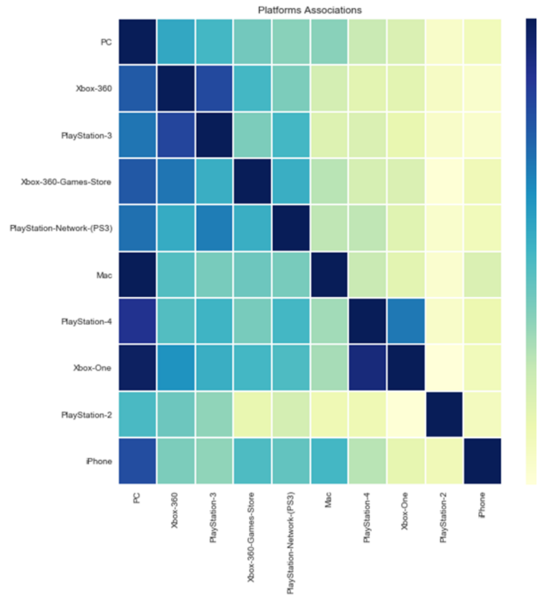
The majority of reviews deal with games of the Action and Action-Adventure genre, Sci-Fi and Fantasy theme, and contain Achievements and Polygonal 3D concepts (3D rendering of games). We expect the content-based filtering system assign less weight to these genres, themes, and concepts, as they are prevalent in most games, when determining how similar games are to each other. Association analysis for the most popular genres and themes:





Most games have a higher association with Action genre and Sci-Fi themes, because there two features are the most popular. Further associations exist between other popular and less popular features, and the system will utilize these associations to filter games with the same associations to make better recommendations.

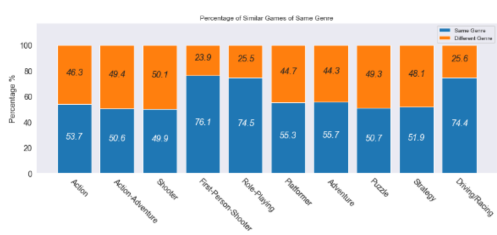
PC Rules All

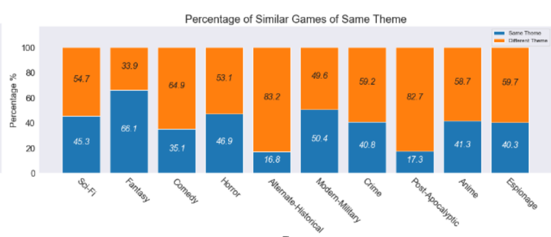


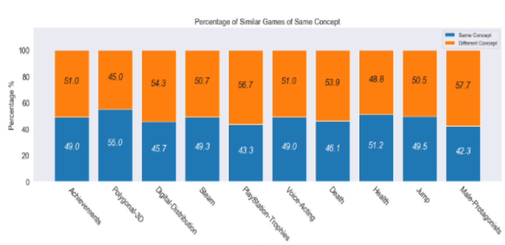
As displayed by the association analysis of the most popular platforms for gaming, all platforms heavily overlap with PC, except PS2. This can be attributed to the innovations in computer technology that allow higher graphics and processors to handle large games to be played on a laptop or desktop computer. We expect however the system to recommend console exclusive or unexclusive games for the games that are from the same platform/platforms, to capture any exclusivity.

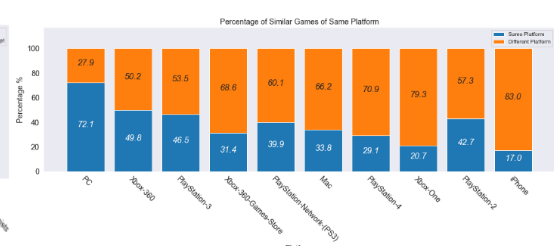
Publishers? Developers? Or Characters and Franchises?

Because the similar games feature will be used to evaluate how well the content-based system performs in recommending games, we take a look at what percentage of games, in the similar games list, are of the same feature (top ten most popular feature) as the target game:









Clearly a relatively high percentage of games are of the same genres, themes, concepts, and platforms, as their respective target game, while the rest of the features (publisher, developer, characters, and franchises) had an extremely low percentage. This indicates that, for the purposes of evaluating the content-based system, publishers/developers/characters/franchises will not benefit the system when recommending games to match the games in the similar games list. This does not mean that these features are useless, as they provide valuable information to match sequels/prequels/spin-offs of games that gamers enjoy, but this won't for our content-based system.

Content-Based Filtering

The NLP technique, TF-IDF was used to determine similarities between games, in which all the game's features (genre, concept, etc.) were aggregated into one single feature, otherwise known as "bag of words", and inputted into the TF-IDF method provided by package, scikit-learn. Initially, all of the features were used in the bag of words to make the recommendations, then were evaluated by the similar games list using the metrics for 10 recommended games:

1. Precision = # of games from the recommendations that are in the similar games list / 10
2. Recall = # of games from the recommendations that are in the similar games list / # of games in the similar games list
3. F1 = 2 * (precision * recall) / precision + recall)

Once the metrics for how well the system recommends games were measured, one game feature was removed from the bag of words, and the system was rebuilt and evaluated on how well it recommended games that were in the similar games list. If the system improved, the feature was kept out, otherwise, it was left in (similar to the concept of stepwise backward selection). The choice of which features were removed were based on the EDA of the similar games list, from the stacked bar plot shown above. The results were:

Table 1. Results of TF-IDF recommendations for each bag of words combination

Bag of Words includes:	Avg. Precision	Avg. Recall	Avg. F1
Every feature	0.087	0.191	0.103
Every feature except characters	0.088	0.193	0.104
Every feature except characters and game body*	0.109	0.206	0.122
Every feature except characters, game body, franchises	0.110	0.209	0.123
Every feature except characters, game body, franchises, publishers	0.111	0.211	0.125

* "Game body" refers to full length description of the game

The best combination of features, as the bag of words, were determined to be genres, themes, concepts, platform, developers, and short game description. The system was able to recommend, on average, at least one game, out of the 10 recommended games, that was in the similar games list, as well as, on average, recommending at least 20% of the games in the similar games list. An example of how well the system recommend games for the game Pac-Man:

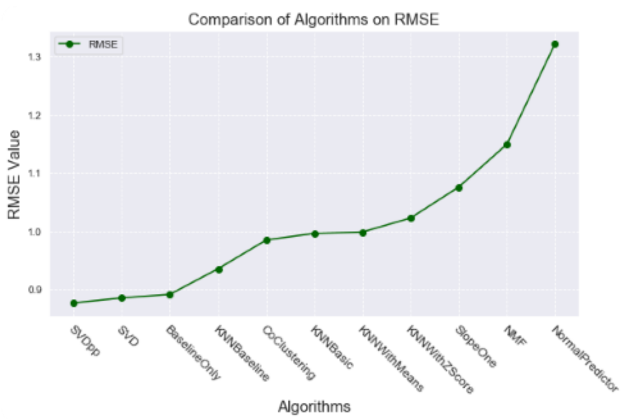
Similar Games list for Pac-Man
Lock-n-Chase
Lady Bug
Katamari Damacy
Spore
Mario Bros.
Dig Dug
Bit Boy!!
Bit Trip Void
The Legend of Zelda: Spirit Tracks
Jungler
Gubble
Monaco: What's Yours Is Mine
Grand Theft Auto
Wolfenstein-3D
Metal Gear
Galaxian

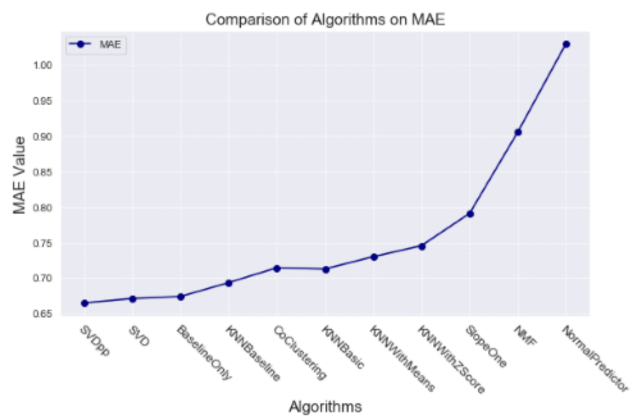
Recommended Games	In Similar Games list?
Dig Dug	Yes
Ms. Pac Man	No
Galaxian	Yes
Galaga	No
Centipede	No
Mario Bros.	Yes
Frogger	No
Pac Man Championship Edition	No
Tetris	No
Space Invaders	Yes

When we think of Pac-Man, and games like it, we think of the games that the recommendation system had outputted, such as classic arcade games Frogger, Centipede, Galaga, Tetris, etc, whose machines would be right next to Pac-Man in an arcade. The recommendation system did well in recommending 4 out 10 games in the target similar games list, but upon closer inspection of the similar games list for Pac-Man, games such as Wolfenstein 3-D and Grand Theft Auto don't seem to bear any similarities with a classic arcade 2D maze game such as Pac-Man. This shows that the similar games list is limited and contains flaws, and does not provide an objective list of similar games for their respective game.

Collaborative-Based Filtering

Using the package surprise, several clustering, matrix factorization, and basic algorithms were running on the dataset to determine which algorithm best predicted ratings for games. The results of the cross-validation of each algorithm, using the error metrics RMSE and MAE, is shown below:



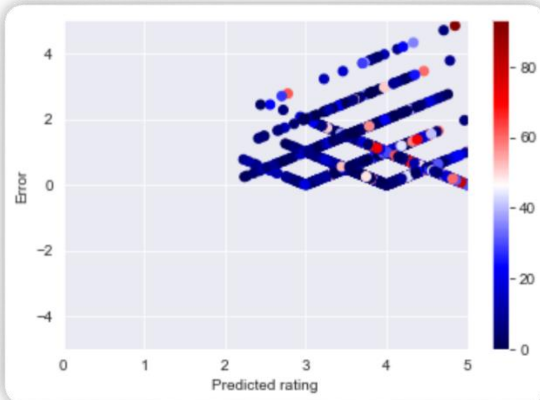


Clearly the matrix factorization algorithms SVD++ (captures implicit feedback) and SVD best produced ratings for each game and were chosen to fit a training set. GridSearch cross validation was used to determine the best parameters for both algorithms, and the model was evaluated using a test set, for which the results were:

Table 3. SVDpp and SVD evaluation of test data

Algorithm	RMSE	MAE
SVDpp	0.8682	0.6662
SVD	0.8708	0.6679

When we look at the predicted ratings vs. residuals (difference between actual rating and predicted) plot:



We see that the system overpredicts many ratings, and doesn't even predict for a score of 0, 1, or 2. This is a result of the uneven distribution of ratings in the dataset, in which it was skewed towards positive ratings (lots of 4's and 5's).

The way this system would recommend games, for each user, is to sort the games, all 4243 unique games, based on the highest predicted rating to lowest. Then we would print the first ten games of the sorted list, as the recommended games for that user.

Biggest Takeaways

- A relatively high percentage of games, in GiantBomb's similar games list, were of the same genre, theme, concept, platform, whereas an extremely low percentage of games were of the same character, franchise, publisher, and developer, when looking at the most popular values for these features.
- For content-based filtering, the best combination of features, when evaluated by the similar games list, was determined to be genres, themes, concepts, platforms, and short game summary. The system, on average of 1 game in similar games list, out of 10 recommendations and 20% of the games in similar games list. The features not used are not useless, since we don't have an objective measure for the content-based system's performance, as in general, they can be utilized to recommend sequels/prequels/spin-offs of the target game, but for the purposes of this project, they are not utilized.
- For collaborative-based filtering, the best algorithm that fit and predicted for the overall dataset was determined to be the matrix factorization algorithms, SVD++ and SVD. However, the system overpredicts ratings, as a result of the uneven distribution of the ratings (heavily skewed to positive ratings). In this case, content-based filtering is the preferable method to recommending the games.

To get more involved, check out my repository for this project, to get access to the data and IPython notebooks of how the wrangling, analysis and modeling took place. Hopefully, you, as an avid gamer, can use the recommendation system to choose what game to enjoy next, and contribute to the accuracy of the recommender system by adding your own ratings and reviews on GiantBomb's website.

Devesh Gokalgandhi is a student at Springboard's Data Science Career Track. His Linkedin and Github can be found here: https://www.linkedin.com/in/devesh-gokalgandhi-a20b3b123/, https://github.com/dgokalga/Springboard-Data-Science

---