The Board Game Recommender

Creating a Board Game Recommendation Application for All Types of Players

By Otis Benbow

**Frame the Problem and Look at the Big Picture**

1. Define the objective of the project.

Board games are a certain type of tabletop game that often include pieces moved around a board under a specific set of rules indicated by the selected game. There are hundreds of types of games, whether it’s family games, dice games, strategy games, deck builders, worker placements, area controls, dungeon crawlers, RPG’s, ASG’s, and not to mention the hundreds more genres of each of these games whether it’s action, adventure, fantasy, farming, puzzles, PVP, PVE and many more!

Now, was that a little overwhelming or what?

To someone who doesn’t play many board games that alone might put you off from ever trying. Not knowing where to begin is one of the biggest barriers of entry to new board game players. I’m an advocate for getting new players involved in gaming to the point they can be truly immersed and have a great time.

So, deciding to do my capstone on board games was a no brainer. They’re my biggest hobby, evident through my collection, attending board game meetup groups, or hosting regular board game nights for my friends.

In this project, I aim to create a board game recommender that is perfect for the player trying to get into their new found board game hobby, and the player who has a collection of a hundred.

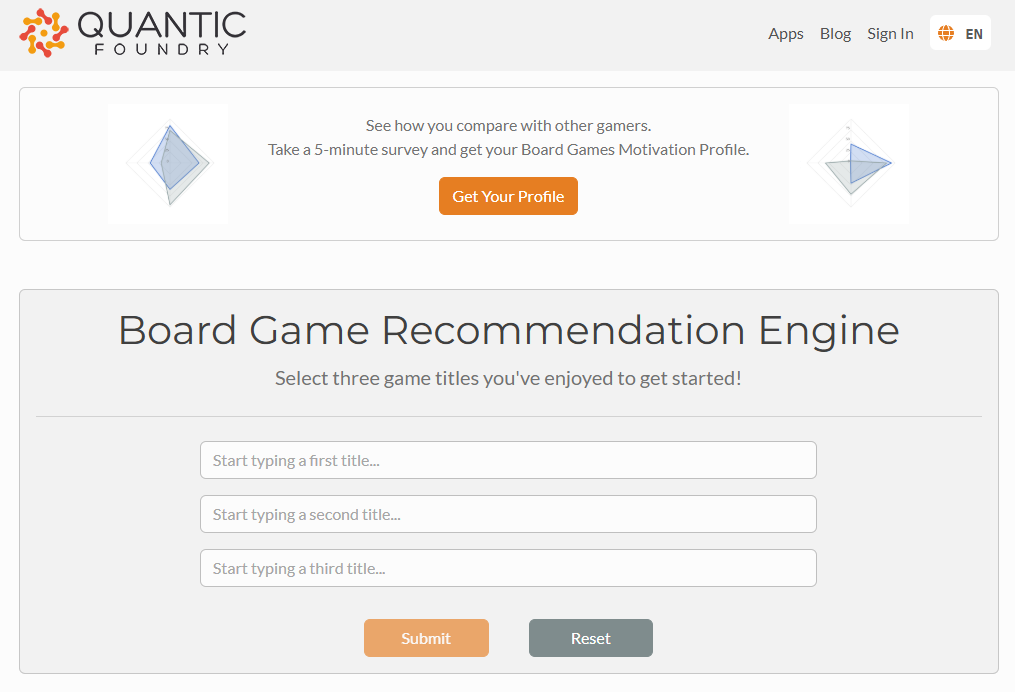
2. How will your solution be used?

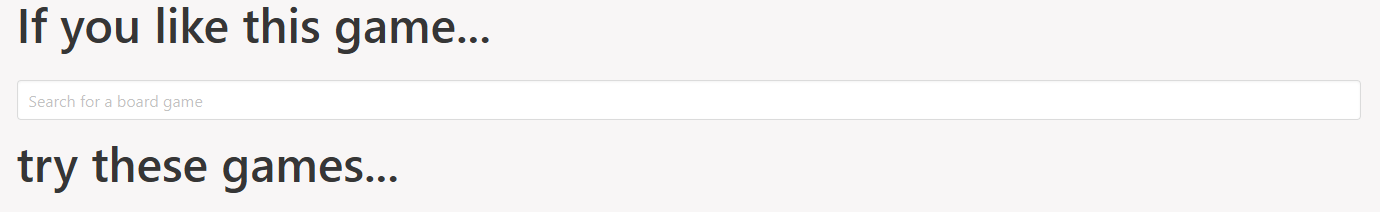
The result of this project can be used by anyone in need of board game recommendations. I aim to create an application that can recommend board games based off of a series of questions and inputs from the user such as player count, timing, genre, simplicity, etc. Which will in turn produce several board games best suited to that player. For help visualising, the app will be in an [Akinator](https://en.akinator.com/) in style of question on question until great options have been found.

I will use it when I’m wanting to find a new game to learn and play, new players can use it to explore board gaming and hopefully develop a passion for it, and experienced players could try it to learn new games suited to their preferred game styles.

3. What are the current solutions/workarounds (if any)?

Having to pick out the perfect game is a problem that all gamers struggle with, which has led to the creation of several board game recommendation sites. They will take some input and then produce a list of the best games suited to you as a player, generally, from the highest rated to lowest rated, according to user reviews of the games. Here are some examples of these sites:

Figure 1: [Quantic Foundry board game recommendation engine](https://apps.quanticfoundry.com/recommendations/tabletop/boardgame/)

Figure 2: [Try These Games Website](https://trythesegames.com/)

What do you notice about both of these recommenders?

They require you to have knowledge of existing board games to produce a recommendation for a new one! So, for a new gamer, it’s going to be difficult to branch out of the genre of games you already know. Take someone who has only played monopoly but wants to find an action-based board game to try out. Based simply on their knowledge of monopoly, they will NEVER be recommended an action game.

So, I intend to make a recommender that is capable of providing recommendations based off of the interests and requirements of a player, rather than their experience with similar games. This unique spin on a recommender is something I look forward to being able to use myself!

However, this doesn’t mean that I will completely disregard the concepts of those recommenders. Having the option to input a board game similar to one that you love would be a great addition, but not what I want to centre my concept around.

4. How should you frame this problem? Explain how/why that’s the case.

I will begin my project with how I’ve come to produce a board game recommender. I will go into my love of games, as well as my experience with getting new players into board games. I’ll make sure to highlight what makes my app unique, specifically on how it differs from board game recommendation websites. Not forgetting to mention how BGG lacks a clear filtering system, so my app is not made redundant by simply using BGG.

From there, I will move onto findings of the dataset and some interesting facts about the most common, and most popular areas in board games.

The main part of my presentation will be the app. I will demonstrate how it works. Attempting to use a real-world example of its effectiveness by trying it out in the previous days and coming back with my findings on how I found the whole experience from gameplay to learning time. This will be the main judgement of the effectiveness of my app, being how a player finds playing the game recommended to them.

I will take time to demonstrate how it works too using audience input. I will hope to find someone experienced with board games who can make assessment on how well it performed, as well as trying it out on a complete board game newbie.

I will finish on some future improvements to the app as well as how to access it in case they wanted to try it out!

5. How should performance be measured?

As mentioned in the previous section, the performance of the app will be judged on how much a player enjoys the recommendation provided for them. This is a lot more abstract than a performance metric like accuracy or RMSE, but is certainly the most effective way to assess.

However, there are ways to immediately determine success or failure in the creation phase. If something I have inputted, such as “I want a 4-player game”, gives a recommendation for a 2-player game, then that is an immediate failure. Another example is inputting the equivalent of “I don’t like action games”, and then being given an action game. That is also an immediate failure. So repetitive and diverse testing of the bot upon creation is a great way to immediately assess the validity of the app.

6. Is the performance measure aligned with the objective?

Using the experience playing the recommended game as my main performance measure certainly aligns with my objective for this project. I want to produce a board game recommender that actually recommends games people will want to play. This is very focused on the human experience, and therefore needs to be judged by a human. It’s very difficult to tell how a game is going to perform until actually trying it out. I’m sure anyone can think of a game they saw that looked awesome, only to try it out and it be very boring. As well as a game they saw and disregarded, only for it to become one of their favourites upon giving it a go.

7. What would be the minimum performance needed to reach the objective?

The most basic version of my project is a recommendation system in python without the creation of an app. It will take inputs and then return a list of the recommended board games and the respective information for those games.

The first measure of performance I will need to hit is ensuring that the model actually produces relevant board games to what has been input. As mentioned before, you have 4 players, but get recommended a 2-player game.

The second is how I personally find the gameplay of the board game I will have been recommended. I will take time to try it out before the presentation and bring anecdotal evidence of it either working/not working (hopefully it works!)

The third will be asking board game players I know to give the recommendation system a go, and see if they might know any of the games recommended. If they say “yes”, and that they enjoy those games, then that is a positive. If they say “yes”, but they didn’t enjoy those games, then that is a negative. If they say “no”, give them the opportunity to try out one of the games. While this would likely not be ready before the presentation, I want my recommender to be useful after this presentation so it still very much matters.

8. Is human expertise available?

I believe I have sufficient knowledge in the realm of board games to make this project a success. As mentioned, it’s one of my main hobbies and I love to learn about new games and trying them out which makes this a perfect fit for me. Additionally, I’m in a number of social groups that are focused around board games, and have lots of friends who love gaming as much as I do. So I’m able to take ideas and inspirations, as well as have them give my project a go.

My expertise in this area will allow me to make immediate judgements on the ability of the recommender as I will be able to recognise many of the games, as well as knowing where to look if I need any assistance or insight.

The only area that I will need to improve upon is my knowledge of creating the application. However, despite this, I have access to so many resources online, the teachings of the academy, and the tutors who can give me advice if anything starts to confuse me or I run into a roadblock.

9. What are the (not so) rough steps to approach this project?

**Step 1:** I will need to gather as much board game data as possible. I will ideally be using [BoardGameGeek](https://boardgamegeek.com/) (BGG) to retrieve the specific information for each game. BGG provides up-to-date data on games as well as reviews and rankings to quantify the popularity and experience a board game provides. They have several sections that allow users to browse games, categories, artists, and trends as well as access forums and find the cheapest place to purchase that game.

To retrieve that data, there are a few options I will be exploring.

The first is the [BGG XML API2](https://boardgamegeek.com/wiki/page/BGG_XML_API2), An API that includes the names, ids, ranks, average rating, and a few unknowns for all of the games in the database. However, there are challenges in the form of accessibility. As well as the data not being relevant enough to what I’m trying to achieve in this project, since I require data on areas such as player counts, genres, etc.

The second is web scraping, this seems much more feasible to retrieve the specific information I am looking for. On BGG they possess a list of all the games within the database. Here is how the website looks.

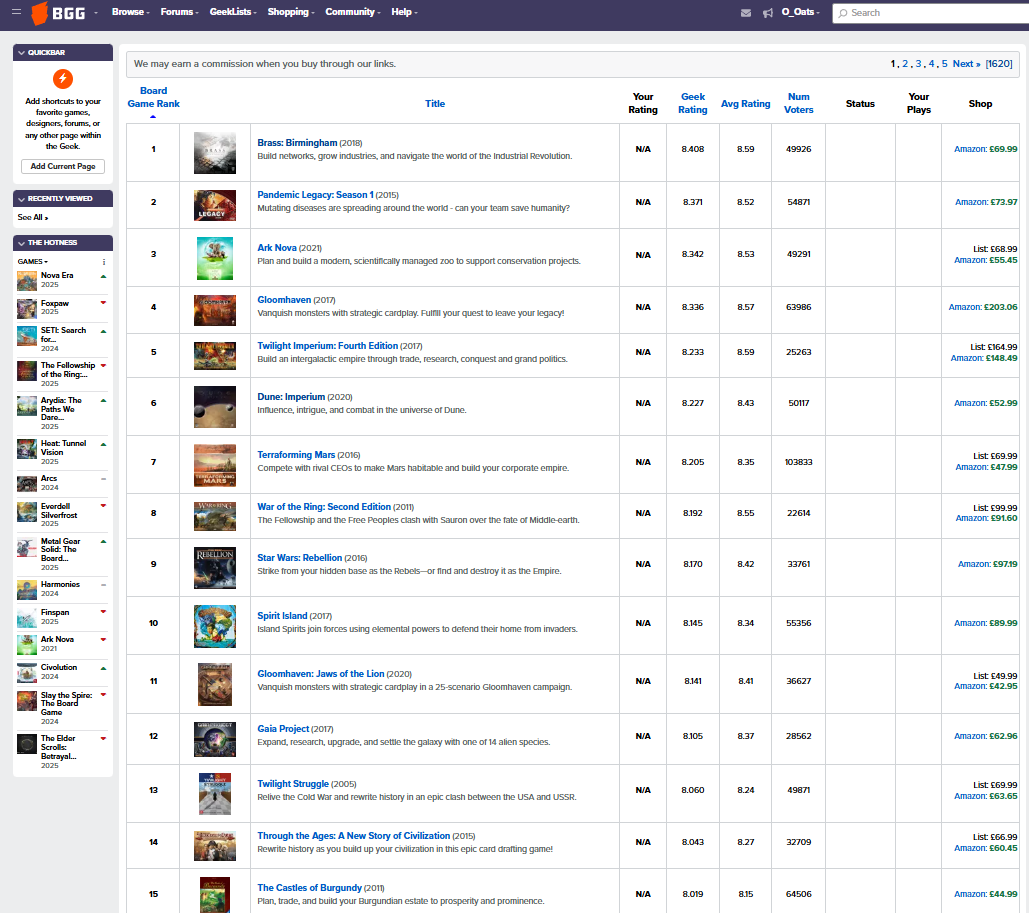
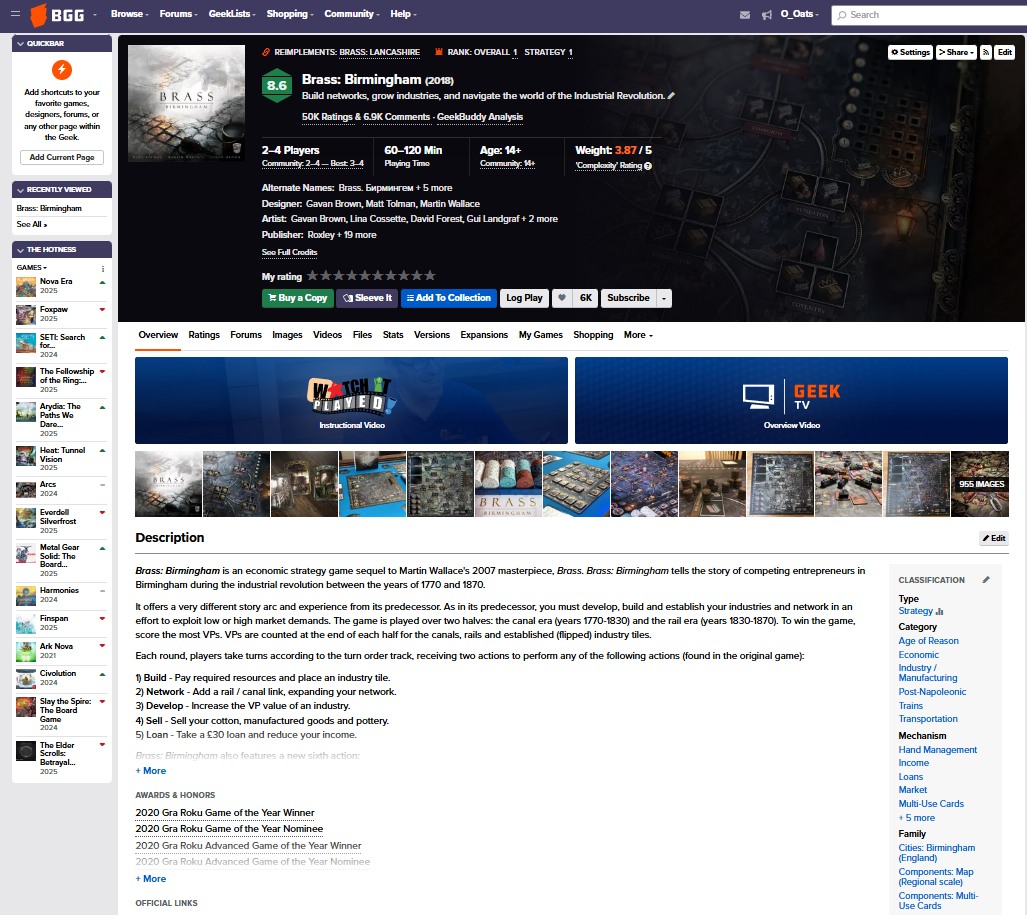


Figure 3: [BoardGameGeeks Browse: All Boardgames](https://boardgamegeek.com/browse/boardgame/page/1)

There are 1620 pages of up to 100 board games on each.

That is ALOT of potential data.

From here you can find the name, release date, a brief description, rating, number of voters, and different pricings. However, this is still not the information I am specifically looking for. So, let’s look further by selecting the top game, Brass: Birmingham (A personal favourite of mine).

Figure 4: [BoardGameGeeks Brass: Birmingham](https://boardgamegeek.com/boardgame/224517/brass-birmingham)

In the top section in black we can find the name, release date, BGG rating, minimum-maximum player counts for the game, minimum-maximum time for completion (excluding rule learning), age rating, and more. Below that is a much more expanded description of the game to get a better feel for how it runs. And lastly, in the bottom right, there is a section titled “Classification”, this contains information on the type of game, different genres, and different mechanisms of the game. There is so much more, but I have highlighted the potential for this method.

As mentioned before, there is so much data that can be extracted here. Potentially too much data, and it might be more realistic to only scrape a certain number of pages. There are two main reasons why I may take this approach. The first is that computational time will be greatly reduced, but the more important reasons is that it may overfill my app with unnecessary, unreleased, or just generally poor-quality games. But deciding what to keep and what to exclude will be something I will consider if I decide to go through with this method.

The final option, and definitely the easiest, is to scour the web further. It is possible someone has already made a dataset that contains all the information I currently desire. So, using their data might just save time. That being said, I want to actively challenge myself on this project, and even if I found the perfect dataset, getting practice using the tools taught to me in the academy is what I want to present.

Now that that chunk is out of the way...

**Step 2:** Transform the data into a csv format

**Step 3:** Clean the data. Essentially, I need to ensure that all data is in the same format, there are no mistakes such as whitespace, typos, etc, and that there are no missing values.

**Step 4:** Using pandas in python to get a feel for how the data looks and works. Specifically, player counts, play times, and most importantly, all the genre information.

**Step 5:** I’m going to summarise the research and creation of the app into this one step. It is likely that I will do further research to figure out how to make an app of this style. I want to make it almost like how the game [Akinator](https://en.akinator.com/) works. So, researching Akinator might be a good place to start here.

**Step 6:** Ensuring that it works effectively, I will provide it to other gamers to test who understand and know a plethora of games to see how good they believe the recommendations are, as well as completely newbies to gaming to see if the selection is appropriate for them.

10. List the assumptions you (or others) have made so far.

The biggest assumption I’m making is that the data will be fairly accessible. Obviously, I’ve provided examples on how I can extract the data, but for web scraping (as an example), understanding the HTML might be far trickier than I expect.

Additionally, I’m assuming that I can learn how to create this Akinator style chatbot in the given time frame. While I have knowledge in the area, actually applying it myself, specifically with coding, is completely new to me. If it is too tricky, I may need to simplify my design, but hopefully not at the expense of quality.