# Introduction

## Introduce the problem

A common feature of most ecommerce websites is a recommendation system that can link a consumer to similar items in hopes of them finding a product and increasing the chances of a purchase. Accurately predicting what a person might find useful can significantly affect the purchase decisions of a consumer and drive-up revenue per transactions.

## Justify why it is important/useful to solve this problem

One of the simplest solutions to finding similar products is to implement a system that utilizes categorical information to find a list of products that are most like the original item. This provides an extremely fast method for retrieving recommendations, with the trad-off or requiring a significant amount of data preparation and a constant pipeline of categorized information for each new product. Another option is utilized the previous sales data of each user in your system to find similar purchases and base your recommendations utilizing that data. Using this method becomes more challenging when dealing with new products that may have had a limited sales life.

## How would you pitch this problem to a group of stakeholders to gain buy-in to proceed?

This project proposes to utilize natural language processing as a method to overcome some of the issues with using categorical data and purchase history. We can utilize machine learning to find similarities in product descriptions and user reviews to aid in the process of finding like products. The first can find these relationships without a necessary structure to its data and utilizes product descriptions from vendors. The latter can take established products and draw from the opinions of reviewers to find likenesses. Using these methods, our recommendations can transcend products that are not categorically similar, like different forms of media.

## Explain where you obtained your data

To prototype the model for this project, I used a dataset of a variety of board games and reviews for them. The data is a collection of entries on the boardgame review site Boardgame Geek and can be found on Kaggle. The idea is to build the model to fit board game products and gradually expand to other forms of media.

<https://www.kaggle.com/datasets/jvanelteren/boardgamegeek-reviews>

# Milestones

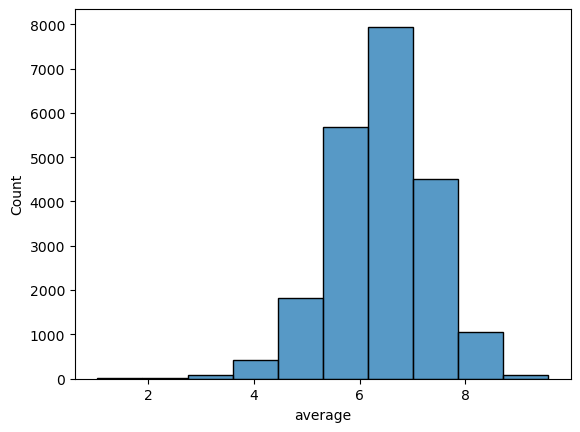
## Exploratory Data Analysis

One of the initial goals of our exploratory data analysis was to find any significant correlations between user ratings for board games, and any other feature in the data set. Categorical information provided no significant correlations and many of the positive correlations found were linked to the ownership or desire to own the game (presence in wish list and wanting lists). The most significant correlation with user rating that I was able to find was in relation to the number of comments in reviews. Board games tend to attract more positive comments than negative.

Graphical user interface

Description automatically generated

Another goal of our EDA was to find a center point to act as delineation for whether users find a game to be favorable or negative. Upon examination, the average of 6.417127 was determined as the center for user reviews. This meant that games scoring above that number would be considered a positive review.



This also brings up one of the first opportunities for improvement that can be implemented in this model. As we saw a strong correlation between the number of reviews and the average user rating, we can determine that the average user rating would increase with the number of reviews. A method that can be implemented in the future to offset this increase in averages is to apply a moving average in relation to the number of reviews and adjust whether a game should be considered positively reviewed.



## Data preparation

Much of the data preparation step was completed in the milestone for EDA as it was necessary to complete that step of the project. Most of the preparation was to trim down the 88 columns of information to the most important ones for the project. In the case of our NLP project, the most important factors to remain are the user comments, ratings, and product descriptions.

Most of the work done during this milestone was preparing the comments and descriptions for our model. This involved the filtering of stop words, tokenizing each entry and stemming the words. This allowed for the comments to be properly vectorized and crate our bag of words matrices for both the comments and descriptions.

One additional preparation I made was to create dummy variables for each of the genre categories, incase I had enough time, I wanted to implement a system that would cross verify them and provide recommendations based on genre similarity.

## Model building and evaluation

This project revolved around providing two different models that could be implemented at different steps in production for an ecommerce platform. The first was to create a fast categorization method that can be implemented on a per request basis, and in our first model we utilized a k nearest neighbor model to implement this system. This model forgoes our NLP strategy and uses the user ratings of like games to find similarities. Essentially if a significant number of used enjoyed two games, they would appear in our recommendations.

To complete this I leaned heavily on online resources to guide me through the process of building this model. To improve space and performance, I mapped indices for both users and game IDs to a sparse matrix, a space saving matrix for when most of the entries are going to be zeros. This allowed me to implement even more user data without a significant sacrifice to performance. After that the process was simple, I created a method to implement a kNN model and fit it to our matrix, then found out nearest neighbors and returned them as a list.

The second method for discovering our recommendations did utilize our prepared product descriptions with a few differences. Instead of stemming our descriptions I used a lemmatizer to prepare the descriptions for our model. This would hopefully allow for more context to be better understood. This was my first attempt at using lemmatization so the function for processing each entry borrowed heavily from online sources, and I ended up using an unprepared version of the product descriptions for the entry data. I also took this opportunity to implement the TFIDF vectorizer for these entries. Fitting the model to the game entries was much easier than with our previous model. I was able to implement a cosine similarity model to retrieve similarity scores compared to each other entry and come up with the top ten results for any game input.

# Conclusion

## What does the analysis/model building tell you?

There is some difference between what each of the models recommend. The first model based on user reviews will return games that are ranked similarly or share similar levels of engagement, while the second model returns recommendations with more mechanical and thematic similarities. One example is the game Ticket to Ride, in our first model we receive recommendations for games of similar popularity and skill levels, and the second returns and assortment of spinoff games to the original.

## Is this model ready to be deployed?

There are still several changes that I would probably make to the model before deployment. The model would still need to be expanded to other type of products to ensure its effectiveness. One this that the model ignores is the overall user ranking for the games, this kind of additional metric can be utilized to ensure higher quality products can be implemented.

## What are your recommendations?

We do see a significant difference between our A and B models so implementations at different points in development of our site would see benefits for either one. Our first model performs roughly 400% faster than our second model, so for situations where speed is important, like rendering a page, it will likely always prioritize over our second.

## What are some of the potential challenges or additional opportunities that still need to be explored?

One of the largest opportunities that I can see is that this seems to be a case that can be further developed using a collaborative filtering approach. There could also be a more streamlined approach with data preparation, the inclusion of a pipeline for adding additional entries would be required before deployment. Another significant challenge was the size of the datasets, which contained millions of user comments and ratings. The CSR matrix did help alleviate some of the issues with this, but I think that there may be a benefit to talking a smaller average of samples for recommendation purposes. Kind of like how the central-limit theorem takes an average of multiple sample averages.