Meeple for the People

PROPOSAL TO BUILD A BOARD GAME RECOMMENDER SYSTEM

Fehir, Aaron

Business Problem: Client is a boardgame café, which has noticed that their sales have been decreasing over the last 2 years due to increased competition. In order to combat this, they would like to implement a service that distinguishes them from the competition, a recommender system to suggest new games to their clientele based on what they've previously enjoyed. They believe that this will allow them to leverage some of their more obscure games and will lead to increased customer satisfaction, thereby slowing, or even stopping/reversing their diminishing amount of market share. Since they have a large university student population, they would like the prototype ready by September.

Which boardgame should we recommend to a customer based on what they've played and enjoyed? Ask customers how much they enjoyed the game they played, see if this goes up after introduction of the App.

Analytic Problem

Build a predictive model which takes games the client has previously enjoyed as input, and outputs one or more games which have similar characteristics ie, build a content-based recommender system. Get feedback about how the recommendation was received. The response to these recommendations will be measured against customers satisfaction with a game picked randomly prior to the release of this software. If the average reception score (measured out of ten) increases by 1, the product is successful. Assessment will be done prior to launch by doing testing/training splits on BGG user data, to see if a player has enjoyed a game they would have been recommended.

Data Understanding and Methodology

Information will be extracted from boardgamegeek.com (the world's largest board game forum / database) using their in-built API: Xmlapi2, a summary of which can be found in the hyperlink located immediately below.

https://boardgamegeek.com/wiki/page/BGG_XML_API2

Database schema:

Database will be relational and constructed in MySQL, since it's open-source.

User information will be contained in Users, primary key is user_ id, each observation is a single user.

Boardgame information will be stored in Boardgames, primary key is boardgame_id, each observation is a single boardgame

The Users' interaction with various games will be stored in the Ratings and/or Collection table/s , which will contain keys linking it to both the User and the Boardgame. Primary key would bet ratings_id/collections_id, respectively. Each observation is the relationship between a user and boardgame.

Getting User Information:

Issue: User information is only obtainable via entering a specific username.

Solution: Get Usernames by scrapping the Plays table for the top 100 boardgames. Grab 50 users per board game (no duplicates), to create a list of users. Feed these user names iteratively into a scrape designed to gather information about a user's game collection and/or plays/ratings to link users to product.

Use Rcurl to import the xmls into R, use xml2 to convert to list, use combination of purr and stringer to recursively collect information about users up until a certain user count (10,000?), and then convert the pooled information into data-frames. Save dataframes into csv, upload csvs as tables in a relational database (MySQL). Updating would be as easy as rerunning the program.

Getting Boardgame Information:

Boardgame database is already made and cleaned, (based off of a scrape performed on BGG data by Baldassare (2017) who created it for the purposes of market segmentation). It could however be updated, and re-run it through the pre-established cleaning protocol. Probably useful, as I'm not currently sure how the boardgames are stored in the database within a user profile, might be THING id, rather than boardgame title or code, if so, this would need to be extracted and added to the boardgame table/dataframe as well.

UI:

Will run in shiny, will involve some sort of prompted rating system to get an initial sample from the user. Hosted on Shiny.io? Input will be a rating of the out of 10, 5, or 3, with some sort of input indicating "I haven't played this game". Haven't decided how to select which games will be used as prompts, or how the prompt will be ordered, but a likely starting point will be boardgames by number of ratings descending, though there are obvious biasing problems with such an approach, which will favor very popular games.

Recommender System:

Content-based system, as the more you go to the café, and the more data you give them, the better the recommendations become. This naturally encourages repeat customers. It also avoids some of the sparsity and privacy issues that would be potentially encountered via Collaborative Filtering. Though, the thought of building two separate systems, one that's collaborative for inexperienced customers, and one that's content-based for regulars is intriguing, though it may lie outside the scope of this project.