Capstone I Project Proposal

BoardGameGeek is an online community of board game players. Classic games, such as “Monopoly”, are listed; however, a myriad of lesser known games such as “Council of 4” are also available. Information is provided for each game including the game’s ratings, number of players, description, type, category, etc. Some games have links to video tutorials on how to play. Within this community, you can buy and trade games from other users or find links to websites where you can purchase them.

Membership to this community is free, however, donations from users are encouraged. There are also ads on the margins of the webpages. One way to increase both these forms of revenue is to increase traffic to the website. A recommender system can help do this. It can also drive up sales of board games for which the website can charge a fee.

A recommender system predicts what rating a user would give an item or product. This information would then be used to suggest the higher rated items to the user for sales or other purposes. Recommender systems are used by many companies such as Spotify, Pandora, Netflix, and YouTube to create music or video playlists. Facebook, Instagram, and Twitter use recommender systems to suggest content and ads to their users. Amazon and eBay use it to suggest items that the user may want to purchase. In this capacity is how a recommender system can help BoardGameGeek increase traffic and sales. A user may not be aware of a game that they would be interested in. The recommender system would predict that the user may like the game and bring it to the user’s attention. This increases the likelihood that the game will be purchased by the user. It also increases the likelihood that the user would return to the website for future board game purchases.

In order to build an accurate recommender system, data is required on all the board games listed on the website. Information about the users, the games, the games’ ratings, etc. is provided by BoardGameGeek’s API. Using Python code, the data will be downloaded from the

API and wrangled into a dataset that can be used for analysis. This will be accomplished using the “xml.etree.ElementTree” Python tool. Later, sklearn tools will be used to build the recommender system from the dataset. The code and markup used to create the recommender system will be available on GitHub.