Predicting Board Game Sales

By: Ryan Solava

Abstract

This project uses Data from Board Game Geek to predict the number of uses who own a game as a proxy for number of games sold. Webscraping is used to obtain the data. Linear regression and random forest regression are used to predict the number owned. From those models, we are able to find the most influencial features.

Design

Board games are a large and growing industry, and so determining what factors affect board games sales would be valuable. We consider this project from the prospective of a board game publisher. Determing what factors influence a board games success would be useful to help determine what board games to publish. Predicting the number of copies sold for a potential game, would help determine how many resource should be put toward that game.

Data

- Board Game Geek Using Selenium and BeautifulSoup data for each game was obtained. One row of our data corresponds to a game. There were over 21,000 games included in our data. The target is the number of BGG users who own the game. Features for each game include:

-Year published
-Rating
-Complexity or weight
-Recommended player age
-Game duration
-Number of players
-Categories (genre)
-Mechanics
-Publishers
-Designers

- Sales data from the publishers GMT Games This dataset gives the estimated number of copies sold for the 15 games published by GMT games. By combining this with the BGG dataset above, we can compare actual sales data with the number of BGG users who own the game.

Algorithms

I cleaned and preprocessed the data by removing outliers, imputing missing values, and converting categorical data to numeric data with one hot encoding. Then I performed typical EDA analysis.

Ordinary least squares linear regression was used first on only the numeric data. This model was improved by removing outliers, including categorical data, removing unnecessary and unpredictive featuers, and adding polynomial terms when appropriate. While each of these steps improved the model, the model still only achieved an R-squared of .283. The model was herteroskadistic with higher predicted values having larger errors. Also, many of the features seemed to have non-linear relationships.

To handle these issues, I also modelled the problem with a random forest regressor. This had the advantage of handling non-linear relationships and some outliers. After tuning the parameters, this model achieved an R-squared of .459. There seemed to be some uncaptured variance, which could be accounted for by irreducible error or perhaps features that are not in our dataset.

Tools

- **BeautifulSoup** and **Selenium** to webscrape and obtain data

- **statsmodels*** to do the linear regression
- **scikitlearn**, specifically their **RandomForestRegressor**

Communication

The slides and presentation are the main way I communicated my results. Also, my code and visualizations are posted to my personal GitHub