Course Project Proposal

Using a recommender system to predict game sales

—

Lindsay Allen
Cpts 315
Washington State University
Mohammad Syed

INTRODUCTION

Deciding which approach to take to PC-based video game design can be difficult at times. Knowing which play styles and themes to follow can give valuable guidance when creating the game. For example, when designing obstacles throughout a level, I have wondered if games with combat or puzzle-based obstacles had higher overall sales. I found this task to be easily applied to numerous game types and styles, both with and without a novel-type storyline.

We set out to investigate the following questions:

Which game attributes correlate with higher sales?

Can we predict the success of a game based on these attributes?

As a software engineer specializing in game design, this project was the perfect opportunity for me to get an insight on what features work and which don't. I wanted to design a game that could be widely accepted, and I was struggling to decide which direction to take with certain levels.

One issue I faced was an imbalance in the number of attributes associated with high sales and the number of attributes associated with low sales. The amount of high selling games greatly outnumbered those considered to be low selling games. To account for the imbalance, we assigned a smaller support level to the high selling games and restricted the number of items considered during the comparison.

Our results showed a promising score for the proposed game. The attributes matched closer to high selling games, and both positive and negative attributes were summarized for possible game design modifications.

DATA MINING TASK

Input Data

To complete this task, we utilized a public dataset provided by Steam. As shown in Figure 1, it included the game ids, game titles, number of owners, and any attribute users have tagged to that game. We also provided a profile for the proposed game that included a game id, game title, a starting projected success score of 0, and a list of attributes describing its play style and atmosphere (Figure 2).

	appid	name	steamspy_tags	owners
	Unique identifier for each title	Title of app (game)	Semicolon delimited list of top steamspy game tags, similar to genres but community voted, e.g. action;adventure	Estimated number of owners. Contains lower and upper bound (like 20000-50000). May wish to take mid-point or lower
	160427.50 - 213500.00 Count: 234 10 1.07m	27033 unique values	Action;Indie;Casual 3% Action;Adventure... 3% Other (6421) 94%	0-20000 69% 20000-50000 11% Other (11) 20%
1	10	Counter-Strike	Action;FPS;Multiplayer	10000000-20000000
2	20	Team Fortress Classic	Action;FPS;Multiplayer	5000000-10000000
3	30	Day of Defeat	FPS;World War II;Multiplayer	5000000-10000000
4	40	Deathmatch Classic	Action;FPS;Multiplayer	5000000-10000000
5	50	Half-Life: Opposing Force	FPS;Action;Sci-fi	5000000-10000000

Figure 1. Small sample of input data taken from Steam dataset

```
class GameProfile:
    gameId = 123456789
    title = 'Proposed Game'
    attributes = ['Post-apocalyptic', 'RPG', 'First-Person', 'First Person Shooter', 'FPS', \
        'Third-Person', 'Third Person Shooter', 'Shooter', 'Violent', 'Action', 'Puzzle', \
        'Magic', 'Sci-fi', 'Scifi', 'Single Player', 'Adventure', 'Large Map', 'Open World', \
        'Horror', 'Free', 'Free To Play', 'Male Lead', 'Female Lead', 'Indie']
    projectedSuccessScore = 0
```

Figure 2. Game profile being tested

Output Data

The output data is an integer value representing the projected success of the game, and ranges between 1 and -1. 1 is considered a perfect score, and -1 is considered the worst possible score. Any attributes that negatively impacted this number are also displayed so the user can decide if they can be removed from the game design.

Questions

Which game attributes correlate with higher sales?

It's valuable to find distinct aspects of successful games. This information is helpful during the design process and can guide major decisions along the way.

Can we predict the success of a game based on these attributes?

Attributes help describe different aspects of the game, such as play style and atmosphere. A summary of popular attributes can give an interesting insight on what patterns to follow when designing each level.

Challenges

- How to even out data sets so one isn't overshadowing the other
- How to measure the importance of each attribute when calculating final classification

TECHNICAL APPROACH

In order to find the most commonly occurring attributes in games with high sales, we used the apriori algorithm to determine frequent item sets. Attributes occurring in these frequent sets were considered significant and used when determining the success of the proposed game. Using frequent sets rather than individually counting each attribute's occurrence keeps the set of attributes relevant to each other and reduces the overall number of attributes we need to consider. In order to classify a game as high or low selling, we separated them based on total sales. Those of at least 5 million in sales with greater positive than negative ratings were considered high selling. Games with less than 1 million sales and greater negative ratings than positive were considered low selling.

The proposed game begins with a projected success of 0. Each common attribute with high selling games will increase this projected success. Alternatively, common attributes with low selling games will decrease this projected success. The range for the projected score is between 1 and -1, with 1 being ideal and -1 being the worst possible outcome.

In order to keep the range between 1 and -1, we calculated the weight of each based on the number of occurrences. We first summed the occurrences of each attribute, $\sum X_{occurrence}$, and then divided each single attribute's occurrence by this value, i.e:

$$\text{Attribute weight} = Y_{occurrence} / \sum X_{occurrence}$$

This weight is added to the projected success score if it occurs in the proposed game attributes. Thus, a game containing all high sales attributes will incur a score of 1. The weight for low sales attributes are calculated the same way, but the weight is a negative value rather than positive. Attributes unaccounted for do not affect the projected success score.

The method we used to calculate the attribute weight can be affected if the amount of data is imbalanced between the high and low sales sets. In order to maintain this balance, we assigned the low selling games a support level of 2 and a confidence score of 0.3. We then assigned the high selling games a support level of 3 and a confidence score of 0.7.

If an attribute is found in both high and lows sales sets, the number for occurrences are first compared. If the attribute has a higher occurrence in one set, the total number of occurrences are reduced by the number of occurrences in the other set, and that attribute is then removed from the other set. For example, the attribute 'Action' is common in both sets, but is associated with a greater occurrence in the high sales set as shown below:

High Sales: Low Sales:

'Action' : 45 'Action': 15

The number of occurrences in the high sales set will be reduced by 15, and the attribute will be removed from the low sales set entirely. Attributes with equal occurrence in high and low sales sets will be removed from both.

The following pseudo code illustrates the approach taken when completing the data mining task:

- Classify games as high or low selling
- Find frequent attribute pairs in classified games
- Measure the number of occurrences for each attribute
- Calculate the weight for each attribute
- Calibrate the projected success score with common attributes

EVALUATION METHODOLOGY

The dataset is publicly available though kaggle, and I found it easy to work with. It was well-documented on the source website which made extracting relevant information simple. We also utilized the Pandas Python library to convert the csv file into a Dataframe object. This allowed us to easily traverse the profile for each game.

The output was evaluated as a numerical representation of the proposed game's similarities to high selling and low selling games. The starting projected success score is neutral at 0, and is manipulated with each attribute match.

RESULTS AND DISCUSSION

The result is a scaled value referred to as the projected success score. Any attributes that negatively affected the projected success score are displayed to the user along with their weight values to determine if they should be removed from the game design. Below is an example output for the proposed game:

```
**********************************RESULTS***********************************

Success! the projected score is: 0.12411347517730498

----------------------------------------------------------------------------

            Strengths:
Attribute: Action | Positive Impact: 0.1702127659574468
Attribute: Open World | Positive Impact: 0.11170212765957446
Attribute: FPS | Positive Impact: 0.09042553191489362
Attribute: RPG | Positive Impact: 0.0851063829787234


            Suggested Design Modifications:
Attribute: Indie | Negative Impact: -0.2857142857142857
Attribute: Violent | Negative Impact: -0.047619047619047616



----------------------------------------------------------------------------
Title: Proposed Game
Game ID: 123456789
Attributes:
Post-apocalyptic | RPG | First-Person | First Person Shooter | FPS |
Third-Person | Third Person Shooter | Shooter | Violent | Action |
Puzzle | Magic | Sci-fi | Scifi | Single Player |
Adventure | Large Map | Open World | Horror | Free |
Free To Play | Male Lead | Female Lead | Indie |
----------------------------------------------------------------------------
```

I wanted to utilize more of the information available in the dataset, so I decided to use the ratings submitted by users as well. This required the game to not only meet the minimum requirement of sales, but to also have more positive ratings than negative to be considered a high selling game. Conversely, the low selling games are upper bounded by a much smaller sale value and required the game to have received more negative ratings than positive. This removed games with low sales due to lack of exposure rather than poor game play.

As far as changes made along the way, the support value for the high and low sales were initially the same. I encountered an issue with the sets being radically different sizes which imbalanced the weight value per attribute.

LESSONS LEARNED

I was able to take away some useful feedback from the results. I used this to modify my game design and improve the projected success score. For example, the 'Violent' attribute of my game had reduced the projected success score by 0.048. Replacing the violent aspects of the game didn't greatly change the game design and helped improve the score.

Looking back, I think I could have utilized more of the data associated with the games. Some games have not received many tags, so it wasn't a very specific way to categorize that game. Those with single broad tags didn't adequately describe the aspects of the game the user found negative for low selling games.

I think it would have been helpful to add more variation to the final game classification of successful or unsuccessful. The current design classifies a game as successful so long and the projected score is greater than 0. I think it would be helpful to break this down further, and maybe make games above 0.4 considered successful and those between 0 and 0.4 as not a failure of a design, but one that needs a few slight adjustments.

ACKNOWLEDGMENTS

I. Steam Dataset

https://www.kaggle.com/nikdavis/steam-store-games