CSE 158 Assignment 2

Steam Game Type Predictor

Gan Kang Fengyuan Wu

December 2, 2021

Abstract

The problem of not having fitting video game to play during vacation and break has troubled us for so long. This often results in a lot of unwanted game purchases and that causes loss to both gamer and game publishers on the online video game platforms **Steam**. To solve fitting game recommendation problem, we have tried out a few commonly known methods of the potential factor model and discovered why a particular model is or isn't fit for this problem. Prediction of fit for a gaped populated dataset has been utilized in this process. The potential factors for gamer, video game type pairs that correspond to their real interest are learnt from the past game purchase and review data. The outcome for a game, video game type pair is predicted based on the difference between gamer reviews and video game and Multiple algorithms have been implemented for evaluating the performance on the **Australian user** dataset.

Keywords

Deep learning, Personalization, Steam, Recommendation System, Games

1 Introduction

During the national wide quarantine caused by COVID-19, we have more free time under work/study from home pattern of life. Playing video game is one of the most common way to spend those time. In the case of game purchases, the biggest issue is there are too many choice such that gamer usually need abundant of time to read reviews and compare across games to find the most fitted one. Price and popularity would also influence the decision of gamer. To alleviate this problem, recommender system has been developed to provide various of suggestions to gamer purchasing a fitting type of game based on the reviews and personal preference. Personalization of the game choosing is a really complex problem that involve with multiple aspects. There are plenty of sources of variance within the genre of games between different publishers. There are gamer who want to purchase a game that can play with his friend and a single-player game for himself. Thus, there are multiple types of game purchases under same account. To help gamer find his type of game, the project aim to give a prediction on a gamers' reviews, video game type pair. Section 1 focus on dataset and analysis on it. Section 2 aimed on prediction task that we explored on the dataset. Section 3 describe our model and explain our decision on to use the model we proposed. Section 4 discussed the literature on how we compare the strength and weakness of existing models. Section 5 briefs the whole project and brings conclusion.

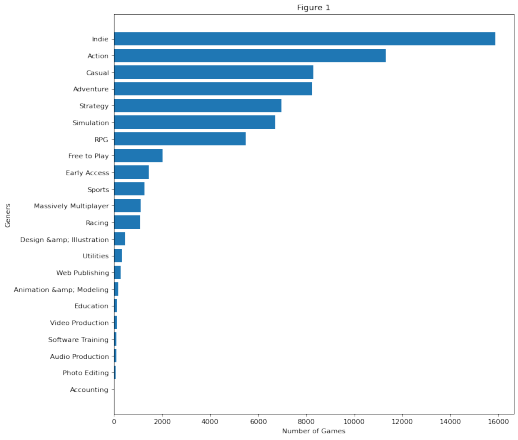
1.1 Dataset Choice

We are using dataset form Julian McAuley's paper about Steam Video Game and Bundle Data[1]. we are using steam_games which

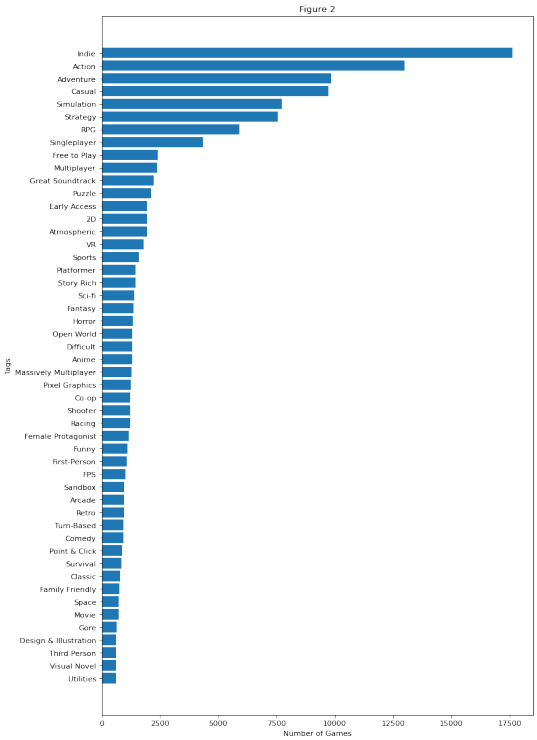
contain steam game metadata and Australian user reviews which contain user reviews in Australian. Given that the whole dataset of user review contain 1.3GB of data, our computational resources couldn't afford that intensive work. So we decide to use smaller dataset with reviews and game information. The smaller dataset also contain 50,000+. So using smaller dataset won't harm the model we discussed in the paper significantly. However, the model we are using would be more accurate if we use larger dataset since the pattern of reviews could overlap more with same type of game.

1.2 Dataset Anlysis

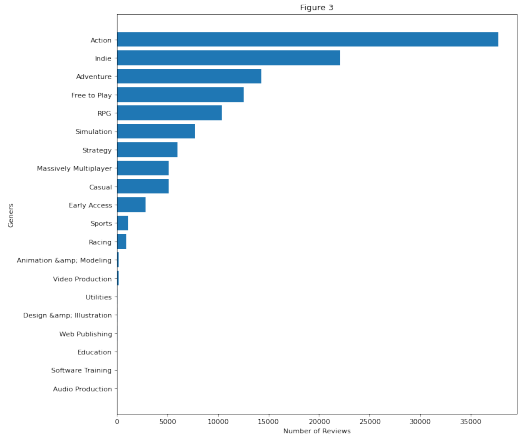
The dataset we are using is from `https://cseweb.ucsd.edu/~jmcauley/datasets.html#steam_data`, and it is extracted from `https://store.steampowered.com`. It's one of the most famous online video game store in the world. The distribution of genres and game numbers are shown below (Given the size of the data, we only displayed top 50 genres and types in the graph below. The actual training and prediction used whole analysed dataset).



From this figure, we can see the Indie, Action, and Adventure types of game are most popular compare to other genres. Our assumption is that those genres of game are more time consuming and attracting to gamer, they have appropriate difficulties and rewards for common gamer.

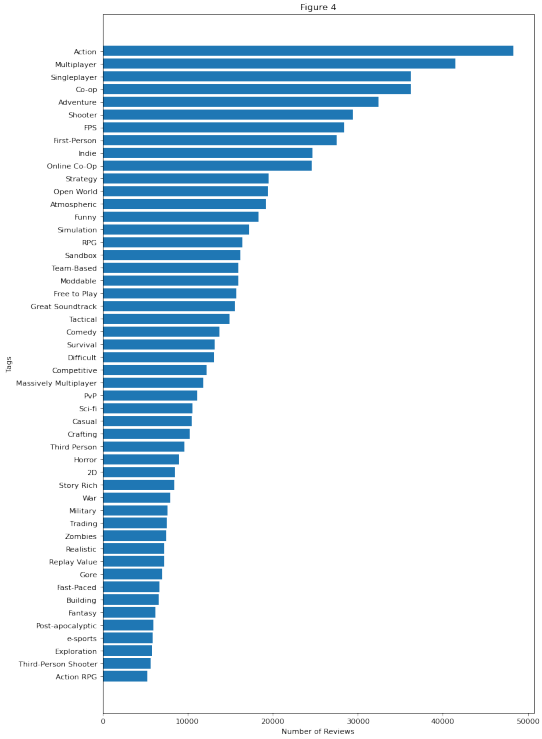


In Figure 2, there are 50 top tags among all games in Steam, The Indie is still in top 1. We are not showing all tags in the graph due to the space issue. This graph could show that the number of Indie is largest tags among all other tags. This is reasonable as Indie only indicate this game is not developed by huge company. There can be a great overlap between Indie game and other tags. And the common fact is that Indie game do have largest amount of developer, which theoretically has largest number. The relation between reviews and game types and be show better in the following figure.



If we compare Figure 1 and Figure 3, we can find that the relation between game number

and review number from different game genres is not positive. And since the reviews are written by gamers, Action genres has more population base than Indie.



So does the Tag, the game tags that theoretically have more players would have more reviews. Like Multiplayer and Co-op tag, these games need huge number of players to make the game playable. And players who play those type of games tend to invite their friend to join them, which further increase the player numbers.

2 Predictive Task

For this project, we identify the predictive task to be "Given a game and its corresponding information, what game tags should it be classified into". By using this predictor, people can automatically generate and correct games tags based on the review of the game. We are not predicting genres since the genres data are highly unbalanced compare with tag data. We afraid this would affect the performance of model negatively when applying NLP.

The validity of the predictions will be measured by MSE. The MSE is defined as

$$\sum_{i=1}^{D}(x_i - y_i)^2$$

2.1 Identify Relevant Features

After analyzing the data set and before building real prediction, we predict there will be a strong relationship between the tags of the game the the review data of the game. Other game metadata like the game publish date and price won't have an significant connection to the type of the game. As a result, we want to do NLP on review datas and using regression predictor to predict the result.

2.2 Data Pre-processing

We first process the dataset by dropping the invalid entries. There are 5317 games have comments but don't have corespondent game metadata in steam_games.json. There are two entries in steam_games.json don't have tag informations. After processing the combining the the game metadata information with review information, we got an data size of 53,986 entries.
For the Tags information, we convert it into one-hot encoded to save up memory.

3 Model

3.1 Model Selecting

The model we choose to used is Logistic Regression model. To be more specific, we predict all tags separately using LogisticRegression in sklearn and passing it into MultiOutputClassifier and calculate the MSE based on the overall prediction compare with the test dataset. We choose the logistic regression since we want to predict an binary output for each of the game tags. As a result, it will be more appropriate to use logistic regression instead of linear regression which predict an linear output.
The feature we extracted is the TF-IDF feature of the review since only using TF-IDF

would provide best result. We also consider following feature extraction:

- popularity of tag,
- the publish date of review,
- the publish date of game (N/A if the game is in early access),
- the existence of tag word in review,
- bag-of-words.

3.2 Model Performance

The baseline model is an naive predictor based on the popularity of Tags which predict all 5 most popular Tags to be true.

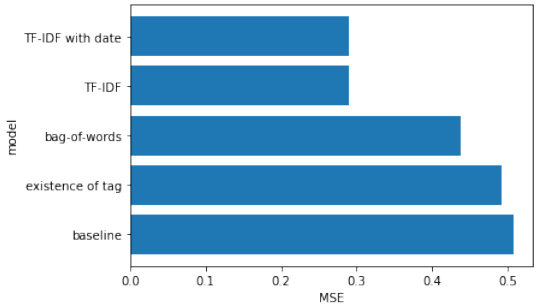
The existence of tag model split the review based on words and predict the Tags if that specific Tag exist in the review content.

The bag-of-words model use bag-of-words as feature which is the one-hot encode of the existence of the first 3500 most popular words among all reviews.

The TF-IDF model use TF-IDF feature. The review data is first being processed by removing the punctuation, stop words and also being stemmed.

The TF-IDF with date model is the same as the TF-IDF model and also add one-hot encoded year and month of the publish data of the review and game as the feature.

The overall model performance comparison is in the following figure.



The model this paper are using, TF-IDF, produce an MSE of 0.2886 compare with the baseline, 0.5078. After adding date feature into TF-IDF, the MSE goes slightly higher into 0.2898. As a result, we finally choose only using TF-IDF feature in our prediction model.

We also tested different C on TF-IDF logistic regression model. The best performance appear when C is lower or equals to 0.01. The output show in the following figure.

```
C = 1e-05; test MSE = 0.2886267527264634
C = 0.001; test MSE = 0.2886267527264634
C = 0.01; test MSE = 0.2886267527264634
C = 0.1; test MSE = 0.2890941464500334
C = 1; test MSE = 0.2893834854217672
C = 10; test MSE = 0.2895170264856443
C = 1000; test MSE = 0.28971733808146005
```

3.3 Known Issue

The primary issue of the model this paper are choosing is the complexity of TF-IDF and also applying logistic regression to predict each Tags one by one. An possible solution is using linear regression to predict and round prediction into 0 or 1, but MSE will raise significantly. We also failed to apply Factorization Machine when analyzing the performance since we can't let fastFM work on our WSL on windows computer. We also try to install it on AWS sever, but python keep exiting during calculating and we can't find out where the issue is.

4 Literature

We have used dataset from *Item recommendation on monotonic behavior chains*[1], It has 4 groups of data for games, reviews, users, and bundles. We used reviews from Australia and games in those dataset. There also have other people who did prediction on this dataset. They are *Item recommendation on monotonic behavior chains*[1], *Generating and personalizing bundle recommendations on Steam*[2], and *Self-attentive sequential recommendation*[3].

In [1], the authors are focusing on the recommendation system prediction based on

user interactions (purchase-play-reviews-recommend chain in Steam example). They has trained and tested their model on the Austrian Steam users' data. We have simplified the process and take part of chain to predict the type of game.

In [2], the authors have studied the relation between bundled items and user preference of purchases. The model they are using is Bayesian Personalized Ranking (BPR), which take implicit feedback to predict a ranking value that the user might want to interact with. In this model, the authors have developed a bundle BPR learning from item BPR. This BPR used bundle correlation to predict the similar bundle that the same user would prefer.

Beside, there are other datasets that been analyze for Steam platform as well.For instance, *Using Steam data to tell you if your game will sink or swim*[4]. This article has studied what factors affect the successfulness of a new published game. The result turns out that your previous game evaluations are important to your next game's success. And the gamer prefer to purchase high-budget game given that the price is high.

4.1 State-of-The-Art

Game type prediction is very important in the area of game recommendations. The tools that mainly used in this area is deep learning and neural networking. These tools use more dynamic and complex matrix and algorithms for feature extraction and prediction, so their result would be more accurate if more feature and data used.

Compare to deep learning and neural networking, traditional methods like linear regression would take less computational resources and have average performance. Like TF-IDF model we are using in this project, it is from sklearn package in Python. This package has other useful tools like linear regression and normalization.

The text analysis is also an important part in this area. NLP (Natural Language Processing) has multiple techniques to handle the reviews of users. Like Bag of words and BERT. The performance of those techniques are different for various of area predictions.

5 Conclusion

In this paper, we have compared the MSE of the different model mentioned in the Model section. The MSE of each model are also calculated and mentioned in the Model section.

The baseline model provide an MSE of 0.5078 since it only consider the top 5 popularity tags.The existence of tag model provide 0.4922 MSE. It provide better result but still can't comapre with TF-IDF since it didn't consider the other word feature inside the review. The best performance is being achieved by using logistic regression with C = 0.01 to predict existence of each tags using the TF-IDF feature from the comment of the game as it consider the overall words using feature inside the review. The temporal feature turns out not related to the tags of the game as adding it into TF-IDF would increase the error.

The TF-IDF model we are choosing provide a very decent result in predicting of tags of the game by analyzing the game comments. The MultiOutputClassifier from sklearn really help me to applying logistic regression on our data set as logistic regression can only output 1-d array. We also try to apply logistic regression to each tags separately before find MultiOutputClassifier. However, this approach prove to be too slow for total number of 339 separate tags. Applying all tags also increase the error of our model as those tags with lower popular have smaller review data set. Overall, we learned a lot in this assignment and also happy about the performance of our model.

References

[1] Mengting Wan, Julian McAuley. Item recommendation on monotonic behavior chains. *RecSys*, 2018.

[2] Apurva Pathak, Kshitiz Gupta, and Julian McAuley. Generating and personalizing bundle recommendations on steam. *SIGIR*, 2017.

[3] Wang-Cheng Kang, Julian McAuley. Self-attentive sequential recommendation. *ICDM*, 2018.

[4] Michal Trněný. Using steam data to tell you if your game will sink or swim. *Venture Beat*, 2017.