**Steam Game Bundles Recommendation System**

Group members: Qiongqiong Lin, Zhuo Chen, Kevin Chiang

**1. Description of the project**

We are going to build this project for an online platform called Steam. Steam is an online store that sells PC games to its users. Games can be viewed as items, and the players can be viewed as users. To stimulate its users to purchase more games, Steam offers a type of selling method called game-bundle. A game-bundle ties several games together, yet the price of a game-bundle is always less than the net price of these games. Our task is to decide what bundle will interest the target user the most and generate a personalized bundle for each user.

**2. New and Innovative about the Project**

In terms of objective, our game recommendation system which does not aim at recommending individual games but the bundle games will generate more revenue for Steam than the general game recommendation system. Bundles are often designed to increase the sales of unpopular games. Users can buy their desired games at a lower price from a bundle, but the total cost indeed increases because a bundle contains multiple games. Hence, recommending a correct bundle would both increase the purchase intention of the user and increase the overall sales.

In terms of the approach we used, since our dataset does not have numeric ratings we choose to use python package VADER[1] to implement text analysis and translate the content of a review into a numeric rating to further support our analysis. Besides, we will design a personalized bundle which maximizes the occurrence of desired games for the target user and thus maximizing the purchase intention for the user.

**3. Literature Review**

Our model is built upon the collaborative filtering recommendation system learned in class and Python’s VADER sentiment analysis package [1]. There are also three papers using the same dataset that we used in this project. [2, 3, 4].

Among three papers, there is one paper addressing the same topic as ours which is generating and personalizing bundle recommendations on Steam [2], by Professor Julian McAuley at the University of California, San Diego. In [2], they build upon their system on a latent factor model using Bayesian Personalized Ranking(BPR) [5]. In the paper, they define three basic notations. Let U be a set of users, and for each user in U, let I be a set of items and B be a set of bundles. The BPR [5] algorithm will rank items that have not been bought by a user using the items that are already owned by the user. Then, for each user, a bundle BPR model is trained based on the ranking in order to find out the item preferences of the user. Once done training, the bundle BPR model will match the corresponding bundles of user’s preferences from the existing bundle dataset. [2].

In [2], the bundle datasets provided by Steam are used as part of the training process, which is not part of this project. Since the definition of bundles in Steam is simply a set of games published by the same game company, whereas we define bundles as a set of games that satisfy or attract the users the most. Thus, we think the bundle dataset is not compatible for our discussion in this project, and we decided to not use it. In [2], this is not mentioned, and so we think the results in [2] are not as accurate as what is expected in the hypothesis.

**4. Dataset**

We obtained our data from Professor Julian McAuley at the University of California, San Diego (<https://cseweb.ucsd.edu/~jmcauley/datasets.html#steam_data>). We used two datasets to build the model. The first dataset contains Australia users reviews on the Steam platform, and the second contains the content of the game. The review dataset contains 58,431 reviews, 25,458 users, and 3,682 games. The game dataset has tags describing content of games, and we can tag 86.7% of the game in the reviews dataset

**5. Hypotheses**

* The review ratings created by the VADER package are reasonable.
* Users opinions would agree with similar user groups.

**6. Evaluation Approach**

For the evaluation step, we computed the VADER score for each game review and used the VADER score as our ground truth. The reason we used VADER score as ground truth instead of recommendation tags is we need to calculate game ratings based on user reviews. We need information from text and the review text data can be transformed into sentiment score, value from -1 to 1 (from most negative to most positive). VADER not only tells us about the positivity and negativity of the text but also tells us about how positive or negative a sentiment is. Hence, in order to have accurate predicted review ratings, we need to apply the VADER package instead of simply classifying whether a game is recommended or not.

We excluded the text not in English, computed the score for each non-English text data, and then checked if the scores match the sentiment. To prove VADER score can reasonably reflect the sentiment of a user, we compared VADER scores with corresponding recommendation tags. We found that VADER works well on game data, most of the positive VADER scores match with recommend tags and most of the negative VADER scores match with non-recommend tags and the total accuracy of VADER scores matching with recommendation tags is 80.7%. Therefore, we decided to use VADER, as it gives us a better understanding of user evaluation; reasonably reflects a sentiment of a user and we could use the score as review ratings. The recommender systems could utilize these rating information as the review rating, and make better estimates.

After we decided to use the VADER score as our ground truth and before evaluating our models, we needed to split the training set and testing set for evaluation. There are a total 48,035 English reviews. We used Scikit-learn to randomly assign 90% of the reviews as a training set and 10% of the reviews as a testing set.

For user based method, we used grid search with K ∈ {10, 50, 100} and S ∈ {Pearson Similarity, Centred Cosine Similarity, Jaccard Similarity}, and have top 3 models as below:

|  |  |  |  |
| --- | --- | --- | --- |
| K (Neighbors) | S (Similarity Function) | Training MSE | Testing MSE |
| 100 | Pearson Similarity | 0.126 | 0.361 |
| 10 | Pearson Similarity | 0.128 | 0.361 |
| 100 | Centred Cosine Similarity | 0.084 | 0.382 |

The best model for the user based method is the model with 100 neighbors and using Pearson correlation coefficient as the similarity measure. The testing MSE is 0.361.

For content based method, we applied LSH method and used grid search with b ∈ {10, 15, 20, 25, 30}, r ∈ {2, 3, 4, 5, 10, 20}, and bucket size ∈ {2\*\*13, 2\*\*14, 2\*\*15, 2\*\*16}, and have top 3 models as below:

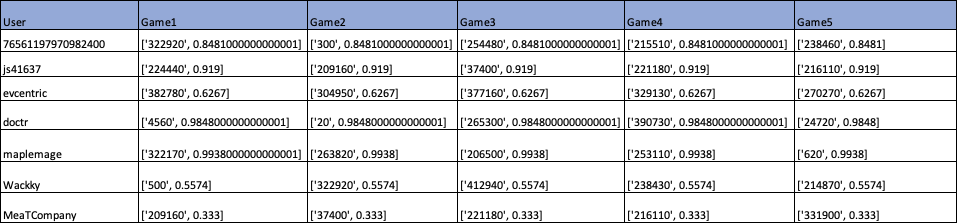
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| b(bands) | r (rows) | Bucket Size | Training MSE | Testing MSE |
| 15 | 5 | 2\*\*13 | 0.06 | 0.105 |
| 20 | 3 | 2\*\*15 | 0.08 | 0.115 |
| 30 | 2 | 2\*\*16 | 0.13 | 0.207 |

The best model for content based method is the model with 15 bands and each bands has 5 rows and bucket size is 2\*\*13. The testing MSE is 0.105.

**7. Result**

After investigating our hypotheses, we found ratings created by the VADER package are reasonable. However, the user based model having higher test MSE shows that the user opinions do not strongly agree with similar user groups, which is contrary to our second hypothesis. Hence, instead of using user based collaborative filtering, we decided to use content based filtering models to build the recommendation system.

Our final result is to generate a bundle with 5 games for each user, therefore, we choose the five games with highest ratings as a bundle for each user by using the best model we chose above.



**8. Conclusions**

In this study, we built user based collaborative filtering and content based filtering models by extracting the information from the review text. We tuned their corresponding parameters by using grid search methods. For this dataset (Australia users reviews on steam), we decided to use content based filtering model as our final model because its test MSE is significantly lower than user based collaborative filtering model. We applied MinHash and LSH methods to speed up the recommendation system, so that it can instantly recommend the top 5 games as a customized bundle.

**9. Lessons learned**

**9.1 The importance of evaluation process**

The first thing we learn in this project is the importance of evaluation. At first, we aimed at coming up with a solution to the problem we designed regardless of legitimate proof of the correctness and performance of the model. We thought that if the model can give us some results that make sense to us, we should be good. Thus, we brainstormed a lot of ideas and possible solutions for the problem we designed. However, in the midterm, we realized that the evaluation is an essential part of a project. When we consider whether a model could give us what we expect, we also need to design an evaluation method that can help us analyze if our thought is correct or not. As a result, during the middle stage of the project, we abandoned a lot of ideas and thoughts, and spent a long time on how we can design our model such that it can fulfill our expectations and meanwhile can be evaluated in a correct way. Thus, we learned from this project that the means of evaluation should always be a factor in composing the possible model or solution.

**9.2 Grouping bundles of games is way more complicated than we thought**

Unlike other real life products, games are digital products, which could be very hard to deal with. In real life, a bundle could consist of things that are a complement of each other, like pens and notebooks. For game data, there is no such concept of complement. Moreover, the games are hard to categorize because each game can fall into certain different categories. We tried to cluster the games by transforming these categories into vectors in Euclidean space, but the results are not as good as we expected. One of the possible reasons might be that game data is a high dimensional data, meaning that there are a lot more features to consider when we try to do analysis.

**10. Future work**

There are certainly more explorations of how to determine bundles that we can do in the future. Now we only used the sentiment analysis score from VADER to determine the top ratings of each user and then form a bundle. We thought of using clustering algorithms such as finding K nearest neighbors to form the bundle once we predict the top game for the users. We did not take this in the end due to we can not find a good evaluation method for this. Furthermore, there are more features of the game data that we did not consider in this project such as the playtime, genre, and price. If we could make use of those features, it will certainly increase the performance of our model.

**11. References**

[1] C. J. Hutto and E. Gilbert, “Vader: A parsimonious rule-based model for sentiment analysis of social media text,” in Eighth International Conference on Weblogs and Social Media (ICWSM-14).

[2] A. Pathak, K. Gupta, and J. McAuley, “Generating and personalizing bundle recommendations on Steam”, *SIGIR*, 2017.

[3] M. Wan and J. McAuley, “Item recommendation on monotonic behavior chains”, *RecSys*, 2018.

[4] W. C. Kan and J. McAuley, “Self-attentive sequential recommendation”, *ICDM*, 2018

[5] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-ieme, “BPR: bayesian personalized ranking from implicit feedback”, *In UAI*, 2009.