An Investigation Into the Prices of Steam Store Apps*

SP20 DSC190A00 Final Project

Jiayi Wu

University of California, San Diego

Yucheng Peng

University of California, San Diego
yup074@ucsd.edu

Zhaoyi Guo

University of California, San Diego
zhg061@ucsd.com

INTRODUCTION

Founded by Value in 2003, Steam is one of the most popular video game distribution services online, selling over 30, 000 games in 2019 and generating more than $4.3 billion in sales. With such a massive amount of games, the prices of these games also became so diverse and varsatile that it could range from free to over $400. As a result, price is one of the most important factors that could determine whether the users will play this game or not. Thus, it would be useful to develop a model that could predict the prices of Steam games.

Previous attempts at Steam games predictions are all focusing on the success of Steam games. That is, the number of audiences one game has. This way, they can help the games creators and developers to better support and fund their games. However, they did it by only predicting the number of audience of the games. If we focus on this topic from a financial angle, we can actually build a model that could help games creators and developers make more profit.

In this paper, we are going to examine the various factors that contribute to the differences in pricing of video games on Steam using machine learning. We will first identify a dataset, and perform an exploratory data analysis for it. Then, we will develop a model to attack this task, and we will optimize our model. Later, we will do research over Steam Games, and we will try other works that are related to the Steam Games datasets. In the end, we will do a final evaluation of our model performance.

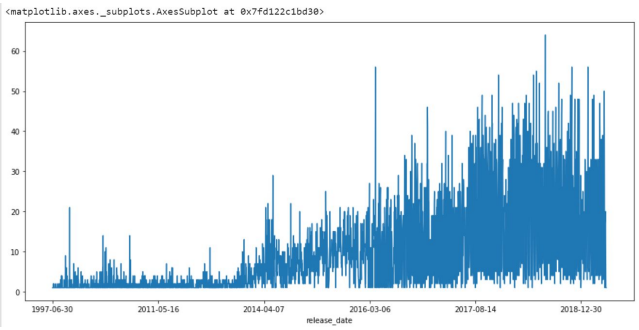
1 Dataset

We extracted our dataset from Kaggle (https://www.kaggle.com/nikdavis/steam-store-games?select=steam_requirements_data.csv).
This Steam Store Games dataset is a clean data set collected from Steam Store and STeamSpy APIs. It covers a bunch of detailed information of 27075 games released before May 2019. The entire dataset consists of 6 separated tables:
steam.csv
steam_media_data.csv
steam_requirements_data.csv
steam_description_data
steamspy_tag_data.csv
steam_suppoer_info.csv.

All of them have a column named appId as an identification for different games. In this analysis, we only utilized the data covered in the first three tables with features including screenshots', 'movies', 'nvidia', 'windows_x', '64bit', 'memory', "storage', 'mean_memory_storage', 'release_date', 'english', "developer', 'publisher', 'required_age', 'achievements', 'positive_ratings', 'negative_ratings', 'average_playtime', 'median_playtime', 'owners', 'windows_y', 'mac', 'linux', and TF-IDF vectorized 'category', 'genre'. We then constructed our model based on the processed features from them.

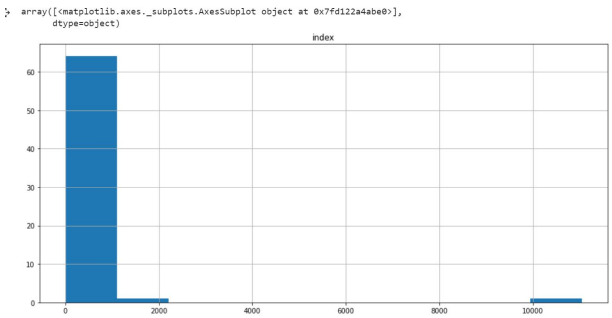
Exploratory Data Analysis

We have information on categorical and numerical properties, textual description, advertising media, system requirements, tags, and technical support for over 27, 000 games, totaling 400 different features. However, most are tags, which are heavily sparse, and some are apparently irrelevant to the price. Therefore, we drop a few datasets and irrelevant features.



Distribution of video games by release date

There are a few interesting findings from the datasets. For example, the release date indicates that Steam is experiencing an exponential growth in the number of video games being released: over half of the games were released in the last three years.



distribution of publishers by number of games

In addition, almost all games are in the language of English and on the platform of Windows. Tags like "Action", "Indie", "Adventure", "Multiplayer", "Singleplayer" are among the most popular game tags on Steam. Moreover, 95% of the developers and publishers are individuals who released or published less than 4 games in total, demonstrating a vast amount of sparse values if taken into the raw model.

2. Predictive Task

Our main task in this report is to predict the prices of each game based on the various processed features and achieve a rmse as close to zero as possible. Since we didn't actually have a test set of data, we trained our model on the entire data set and tried to come up with a concise prediction. In order to prevent overfitting, we used cross_val_socres with cv = 5 to find out the mean score for all runs of cross validation. Our baseline model only includes the numerical features from steam.csv to get a RMSE of 7.32. We then made some improvements to the baseline model by extracting and engineering more numerical features from steam_media_data.csv, and steam_requrement_data.csv, and categorical data from steam.csv. After adding these features, we observed a drop about 1 of the cross_val_scores of our model.

3. Model

We chose to use LightGBM as our baseline and final model. LightGBM is widely used on Kaggle. It trains rapidly, and most importantly it outperforms all other models for most of the cases. There is no reason to use other inferior models as baselines just for the sake of having them.

Though the datasets supply many features, as indicated in our Exploratory Data Analysis, most of them are unusable if not processed. We thought about collapsing features into categories and tags that are more general, such that "massively_multiplayer" into "multiplayer". However, it would require manual labelling for over 300 labels or using pre-trained word embeddings such as BERT to cluster these tags. Due to the limited time frame, we did not include this part of processing in our project but it can definitely be included in future.

Instead of collapsing features into categories and tags, we, on the other hand, included extracted information such as how much advertising information each game has, whether it has a support website, and most significantly the minimum system requirement each game could run on. We hypothesize that games with higher minimum system requirements, like newer Windows versions, have more RAM, more storage, and require discrete graphics cards like Nvidia GeForce. We based our hypothesis on the fact that newer games tend to have better graphics and bigger size which are reasonably correlated to the price due to developmental efforts and such. As a result, our hypothesis improves our model.

nvidia	count	mean	std	min	25%	50%	75%	max
0	19872.0	5.339506	6.563177	0.0	1.69	3.99	6.99	303.99
1	7190.0	8.120976	10.428071	0.0	2.09	5.79	10.99	421.99

Game with description with "Nvidia" price distribution

In addition, if we were to fine tune the model, the most straight-forward way is to GridSearchCV on parameters such as n_estimator, max_depth, num_leaves and so forth. However, our final model's performance is about 5.66 and our main objective is not to have perfect prediction. Thus, it is unnecessary to add GridSearchCV at this stage.

4. Literature

Our datasets have been studied by others before. However, their studies are different from the task we've been working on. They used these datasets to predict the success of Steam Games. That is, the average number of concurrent players in the first two months after release. In order to attack this task, the author used Chi squared to compute the importance of all attributes. This way, they could filter unimportant features. In terms of models, they used RPART, GLM, Random Forest, SVM, and Naive Bayes, and compared their accuracy, precision, recall and F1 scores. As a result, SVM gave them the highest correlation, while Random Forest gave them the highest precision and recall scores. Also, features like GPU and tags are highly important in their models. [1]

Rather than focusing on the success of Steam Stores, we are working on a brand-new task, which is to predict the prices of the Steam Store apps. Also, instead of using Random Forest and SVM, we used a gradient boosting model: LightGBM regressor. LightGBM regressor is a tree based algorithm, which is relatively new, but it is popular because it is sensitive to overfitting, and able to handle large datasets. Therefore, the major novelties of our work are that we focus on a different task of predicting the prices of Steam Games, and that we use LightGBM to build our model.

```
LGBMRegressor(boosting_type='gbdt', class_weight=None, colsample_bytree=1.0,
              importance_type='split', learning_rate=0.1, max_depth=-1,
              min_child_samples=20, min_child_weight=0.001, min_split_gain=0.0,
              n_estimators=100, n_jobs=-1, num_leaves=31, objective=None,
              random_state=None, reg_alpha=0.0, reg_lambda=0.0, silent=True,
              subsample=1.0, subsample_for_bin=200000, subsample_freq=0)
```

The default parameters for our model

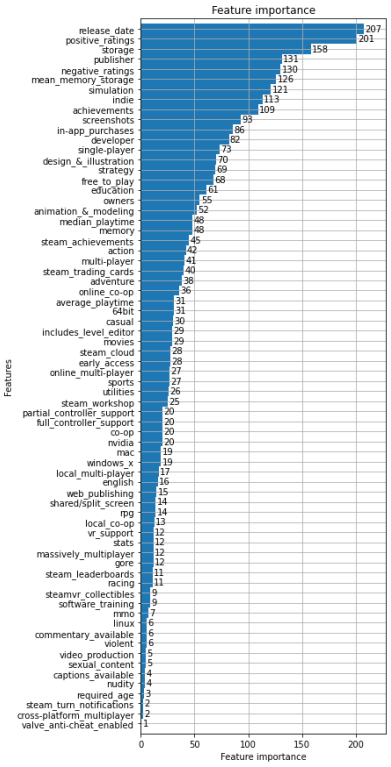
5. Results

Though our model beats the baseline, we are unsure if the performance is good due to the lack of an explicit expectation or a counterpart. Nevertheless, the improvement is verified and confirmed with multiple iterations of cross validation. We contribute the improvements to text mining and in-depth feature engineering as well as some background knowledge of video games and distribution platforms. The model could be further improved if genres, categories, or tags are included because they are likely key determining factors too.

	CV1	CV2	CV3	CV4	CV5
Base	-6.974	-7.071	-8.257	-8.191	-6.137
Final	-5.666	-6.056	-7.197	-7.416	-5.134

Some features such as number of trailers and advertising images are not as useful compared to other features like release date and hardware requirements.

According to the feature significance, having achievements in games and more screenshots and better graphics (bigger size and higher minimum system requirements) seem to contribute the greatest to video game prices. A highly priced game often correlates to more critical comments of both positives and negatives.



CONCLUSION

In this paper, we developed a model and examined the various factors that contribute to the differences in pricing of Steam video games. First of all, we extracted our dataset from Kaggle and did exploratory data analysis. By analyzing the datasets, we realized that many datasets and features are irrelevant, so we filtered some of the information from these datasets. Also, we had many interesting and useful findings, such as popular tags and release dates which we took for future considerations.

After we did exploratory data analysis, we identified a predictive task based on our dataset. That is, predict the prices of each game based on the various process features. Then, we decided to evaluate different models in this task by calculating their RMSE scores. Also, the baseline model we wanted to compare with is the RMSE score of 7.32. Our based model is appropriate because it includes all the important numerical

features, and it was processed with cross validation to prevent overfitting.

With the predictive task in mind, we started to train our model. The model that we proposed to attack this task is LightGBM. The reason we chose this model is that it is sensitive to overfitting, and it is able to handle large datasets. In order to optimize our model, we designed many features for our model by extracting large, sparse information into binary features, and categorical features. The reason that we didn't collapse features into categories and tags is that there were over 300 labels, and manually labeling them would be challenging in our limited time frame. Also, we didn't add any 3rd-party libs, since our model has already outperformed the baseline model.

By researching tasks related to Steam games, we found out that our datasets have been studied before. However, they used the datasets to predict the number of players for a game. In one of the reports, the author did feature selections, and used multiple models to compare their accuracy, precision, recall and F1 scores. As a result, models SVM and Random Forest have the best performances. By comparing their work to our work, the major novelty of our work is that we are working on a brand-new task: predicting the prices of Steam Games. Also, we used a LightGBM regressor to build our model, which is more advanced, and efficient.

As a result, our model outperformed the baseline model, and the gap we made from the baseline model is significant. However, not all features are designed to be effective, because some of the features have low importance when it comes to training the model.

It might be a bit more accurate if we were to tune the hyperparameter of the model. However, the increase would be rather limited due to the fact that the quality of features determines the lower bound of how good the model can eventually perform. Also, since the RMSE is relatively low, it is unnecessary to fine tune the model to a point where it overfits.

Ablation Study

The definition of ablation study "has been adopted to describe a procedure where certain parts of the network are removed, in order to gain a better understanding of the network's behaviour." [Long] Though increasing features from 12 to 80 does not increase the RMSE by a lot, removing key features will be detrimental.

Case Study

We examined the top 20 games with the highest percent of errors. All of them are priced at an abnormal level: all are at least $60 and some are even above $100. Reasonably, pricey games are unlikely to be attractive to gamers; almost all of them have no average playtime, meaning no one is buying them. At the same time, the minimum system requirements of those games are among the highest, which also explains why the lack of popularity. They can be seen as outliers and, since most games are relatively inexpensive, those games are predicted at a much lower price than they actually are.

REFERENCES

[1] Michal Trneny. (2017) *Machine Learning for Predicting Success of Video Games.* Masaryk University. https://is.muni.cz/th/k2c5b/diploma_thesis_trneny.pdf [accessed 8 Jun 2020].

[2] Robert Long. *What is an ablation study? And is there a systematic way to perform it?* Stack Exchange. https://stats.stackexchange.com/questions/380040/what-is-an-ablation-study-and-is-there-a-systematic-way-to-perform-it