Independent games, or indie games, have been booming in recent years and received much attention from both the customers and game developers. This paper explores different types of features extracted from the indie games’ metadata, including context features and text features, and relies on logistic regression and random forest algorithms to predict the popularity of indie games. The result shows the popularity prediction is feasible and logistic regression achieves higher performance. Some most indicative features (e.g. user-generated tags) are discussed to help game developers and publishers identify and avoid potentially unpopular game genres, thereby having larger possibilities to stand out in the market.

Headings:

Independent game

Indie game

Predictive analytics

Machine learning

Text mining

Popularity prediction
PREDICTING THE POPULARITY OF INDEPENDENT VIDEO GAMES ON THE STEAM PLATFORM

by
Ziyang Jiang

A Master’s paper submitted to the faculty of the School of Information and Library Science of the University of North Carolina at Chapel Hill in partial fulfillment of the requirements for the degree of Master of Science in Information Science.

Chapel Hill, North Carolina
April 2021

Approved by

_______________________________________
Yue Wang
# TABLE OF CONTENTS

1. *Introduction* ............................................................................................................. 2

2. *Literature Review* ........................................................................................................... 4
   2.1 Consumer Behavior ................................................................................................. 4
   2.2 Game Attributes ...................................................................................................... 5
   2.3 Independent Games ................................................................................................. 7
   2.4 Predictive Analysis ................................................................................................. 8

3. *Proposed Method* ........................................................................................................... 10
   3.1 Problem Definition .................................................................................................. 10
   3.2 Dataset Introduction ............................................................................................... 11
   3.3 Context Features ..................................................................................................... 13
   3.4 Textual Features ...................................................................................................... 17
   3.5 Machine Learning Algorithms ................................................................................. 17

4. *Results* .......................................................................................................................... 19
   4.1 Experiment Setup .................................................................................................... 19
   4.2 Evaluation Metrics ................................................................................................... 20
   4.3 Hyper Parameter Tuning ......................................................................................... 20
   4.4 Performance Evaluation .......................................................................................... 24
   4.5 Parameter Importance ............................................................................................ 25
      4.5.1 Logistic Regression ......................................................................................... 25
      4.5.2 Random Forest ............................................................................................... 28
   4.6 Error Analysis ........................................................................................................... 30

5. *Discussion* .................................................................................................................... 32
   5.1 Summary .................................................................................................................. 32
   5.2 Limitation ................................................................................................................ 33
   5.3 Implication ................................................................................................................. 33

6. *Conclusion* .................................................................................................................... 35

7. *References* ..................................................................................................................... 36
1. INTRODUCTION

The term independent, or indie as an abbreviation, comes from the film industry. It was originally used to stand for films produced outside the major film studio system. The game industry continues to use this term and inherit its spirit. Although it is a relatively new concept in the history of digital games, “indie” is now a ubiquitous designator for certain kinds of digital games and developers. (Parker, F., 2013, p. 1) However, more attention brings more competition. We can easily perceive the enthusiasm for game development from the rising number of game developers. For example, in 2018 there were hundreds of indie studios sprouting up all around Spain, and only 10 of them were actively turning a profit off their games. (Wright, 2018) Making their games stand out in an exponentially increasing market is a big challenge for every independent developer group.

Steam is an ideal platform for studying and analyzing indie games due to a large number of samples on it. According to SteamSpy, a web site for collecting Steam data, “The weekly trickle of games flowing into Steam has surged into a roaring river, with the number of games on the platform nearly doubling every year from 2014 to now — 1,772 that year, 2,964 in 2015, 4,207 in 2016, and 7,672 in 2017.” (Wright, 2018) One of the reasons for the explosion in new releases on Steam is Steam Greenlight, which allows
game developers, primarily indie developers, to present their games in advance on Steam, get feedback from players, and possibly start making a profit.

This paper explores different types of features extracted from the indie games’ metadata, and uses these features to create, test and validate a model to predict the popularity of indie games. Two machine learning algorithms are involved in this paper — logistic regression and random forest. The experiment results show logistic regression outperforms random forest, and both two models beat the majority guess. Then the parameter importance is further analyzed to find the most indicative features. The anticipated findings are possible to help game developers and publishers to improve the attraction of games, thereby stimulating customers to make a purchase.

This paper is organized as follows. Section 2 reviews related researches and introduces the indie game background. Section 3 talks about the dataset used in this paper, proposes the feature extraction method and machine learning algorithms. Section 4 discusses the experiments and results. Section 5 covers the implications and limitations of this paper. Section 6 makes the conclusion.
2. LITERATURE REVIEW

2.1 Consumer Behavior

Since we try to predict the sales volume of indie games, it is necessary to explore the researches on purchase behavior and consumption intention. Engel et al. (2002) treated purchase behavior as a type of psychological decision making. In order to satisfy the needs, consumers will search for the related information according to their experience and external environment. After the information is accumulated, consumers start the evaluation and consideration. Upon comparison and judgment, they make purchase decisions. Similarly, there are well-known five stages of a consumer buying process: (a) identify the problem (the recognition of the needs), (b) information search (find the possible option from various information sources), (c) evaluation of alternatives (compare with other products that can fulfill the requirements as well), (d) purchase decision (choose the product to buy or not), (e) post-purchase evaluation (assess if the product satisfies the needs). In the context of game consumption, the steps of purchase behavior keep the same, but with a more general need — entertainment. Given the virtual nature of the game, the product attributes and the information sources where publicize the games should be different from those physical goods. Therefore, it requires the survey of game attributes researches.
2.2 Game Attributes

Video games were born with controversies. Many types of research under the domain of video games focused on the negative effects they may had on human beings, such as gambling, violence, addiction. A wide variety of game attributes were exploited to quantify the scale of game attractiveness during these researches. Wood et al. (2004, p. 3) were the first to use a framework of structural features of video games in order to conduct a survey that determines what features are important to video game players. The main categories of structural features are listed below:

- **Sound**: the sound used in the game, including background music, sound effects, speaking characters, etc.
- **Graphics**: the graphical display of the game, such as the display resolution, art style, full motion video.
- **Background and settings**: the use of realistic settings and fantasy settings, the connection with other entertainment products (films, novels).
- **Duration of game**: the time of completing the game playing.
- **Rate of play**: refers to how quickly the players immerse themselves in the game and how quickly the gameplay advances.
- **Use of humor**.
- **Control options**: the sound, graphics, and skill settings of the game.
- **Game dynamics**: the overall game experience including exploring new areas, AI interactions, collecting things, elements of surprise, etc.
- **Winning and losing features**: the potential to gain or lose points and the saving functionality.
• Character development and customization.
• Brand assurance: refers to brand loyalty and/or celebrity endorsement.
• Multiplayer features: referring to various multi-player options, communication methods, building alliances, and beating other players.

King et al. (2007, p. 93) expanded the framework above by using a five-feature model of video game structural characteristics: (a) social features, (b) manipulation and control features, (c) narrative and identity features, (d) reward and punishment features, (e) presentation features. Each of the groups contained several sub-features, which introduced a couple of new features such as theme and genre features, in-game advertising features, leader board features, etc.

The researchers from the domain of human-computer interaction provided a different perspective of video games. González Sánchez et al. (2009, pp. 66-67) regarded video games as a special interactive system. Trying to evaluate the “usability” of video games, they considered not only functional values but also the specific properties of video games, which introduced the “playability” of video games. The Playability Model aimed to characterize the player experience (the corresponding word is “user experience” in HCI), which uses six facets to identify different game attributes and properties: (a) intrinsic playability (gameplay design), (b) mechanical playability (game engine, graphics, sounds, textures, lighting, etc.), (c) interactive playability (game interface, input method, interaction dialogue, etc.), (d) artistic playability (the quality of the artistic and aesthetic rendering in the game elements), (e) intrapersonal playability or personal playability.
(individual perceptions and feelings when playing the game), (f) interpersonal playability or social playability (the group awareness that arises when playing the game).

The researches above explored various aspects of video games from the game itself, but people may be affected by external environment factors. If trying to analyze the consumer behavior of buying games, the information sources could be the website selling the game or the recommendation from friends and other players. Macey et al. (2020) deemed: “This is a significant gap because the intention to play the game is also heavily influenced by factors outside the game, for example by a game trailer or social activities of spectators during game nights and get-togethers, which allows people to form new kind of social communities around the games.”

2.3 Independent Games

In general, independent games and mainstream games are two sides of the same coin. However, game studies have not provided a clear definition of the indie game. At the same time, indie games receive significant attention from the industry after the success of several well-known indie games, such as Limbo, Minecraft, Castle crashers. The Independent Games Festival, the largest annual gathering of the indie game sector, states the official rules of the candidates for the best indie game of the year should be the games created in the “indie spirit” by an independent game developer (Pérez Latorre, 2016, p. 17). Lipkin (2013, p. 8) considered the indie games as ideologies pitting political-economic dissent against major game publishers. Grayson (2012) indicated indie games on the one hand, “stood for freedom of expression and unbridled experimental spirit,”; on
the other hand, “became a for over-inflated egos and introspection with all the depth of a sun-dried puddle.” Lipkin (2013, p. 15) deemed that the development of indie style was limited to economics and its beliefs. “because 2D is cheaper to produce than 3D, indie games trend towards the 2D. Pixels are easier to program than circles or spheres, hence the voxel craze.”

2.4 Predictive Analysis

The predictive analysis for web content attracts a large amount of attention among machine learning researchers. Szabo and Huberman (2010, p. 85) observed that the log-transformed long-term popularity of a given content is strongly correlated with its early popularity. Based on this observation, they proposed a linear function where the future popularity (e.g., the number of views) is related to its early number by a constant factor. Pinto et al. (2013, pp. 368-369) tried to extend Szabo and Huberman’s model by considering regular intervals as the input parameters rather than a fixed date. In addition, they also assigned different weights to each sampling interval to capture the popularity evolution patterns of videos. Lee et al. (2010, p. 626) used a Cox proportional hazard regression model to fit the distribution of popularities of online discussions. They only used the number of comments and the number of posters during the first few days of the content to predict the lifetime of threads and the final number of comments.

Researches above relied heavily on the historical data of web content, which was usually acquired through crawling techniques. However, accessing view histories is difficult or expensive in practice for some types of items. Alternatively, many researches tried to
leverage other features to make the prediction. For instance, Gelli et al. (2015, p. 908) tent to predict the popularity of an image posted in a social network using four types of features: user features, object features, sentiment features and context features. Among these, object and sentiment features were extracted by the convolutional neural network models while user and context features were explanatory features such as tags, description, the mean image views of the user. In addition, they also conducted a qualitative analysis to determine what types of images were popular or unpopular in terms of sentiments and semantic metadata.

He et al. (2014, p. 234) focused on user comments rather than view history to predict item popularity. It is interesting to note that they did not use textual features but the timestamps and usernames of comments, which is based on the three observations: (a) if an item receives many recent comments, it is more likely to be popular in the next time step; (b) if the users commenting on an item are more influential, the item is more likely to receive more views in the future; (c) if an item is already popular, it is likely to garner more views in the future (He et al., 2014, p. 236). Since game popularity draws little attention among various web content, this paper tries to combine contextual features with the textual features extracted from the game overviews to make an exploratory prediction.
3. PROPOSED METHOD

3.1 Problem Definition

The purpose of this paper is to explore different types of features extracted from the overview of an indie game, and use these features to create, test and validate a model to predict the popularity of the game. Given the purpose, this paper seeks to address two research questions:

a) Can we use the information on game overviews to predict the popularity of Indie games?

b) For each part of the game overviews, what are the most indicative of the popularity of Indie games?

To be clear, two key terms in the research questions are elucidated below:

a) Indie games (independent games). Indie games can be explained in different forms in different contexts. In general, “independent” is regarded as the antonym of “mainstream”. In this paper, the indie game is defined under the economics or capitalism domain. In other words, indie games are those games with a limited budget in terms of development processes, human resources, and advertisement.

b) Game popularity. It is intuitive to define popularity as the sales volume of the game. However, the accurate number of game sales is usually hidden from the
general public. Game publishers may make the announcement when the sales reach some large numbers (e.g., one million, five million or ten million), but it is difficult to acquire the exact number during the time intervals. This situation might be especially worse for indie games, because most of them have relatively small sales numbers. Alternatively, this paper uses the number of owners of a game as the popularity of the game on Steam, which is one of major statistics displayed on Steam Spy — a Steam stats service website which “automatically gathers data from Steam user profiles, analyzes it and presents in simple, yet beautiful, manner.”

(About -, 2021)

Based on research purpose, it is clear that the supervised machine learning task is applied to solve the research problems in this paper. Accordingly, the independent variable should be the popularity of indie games, and the dependent variables are game features extracted from the introduction page on Steam. In addition, the machine learning task could be treated as a binary classification problem due to the nature of the dataset used in this paper.

3.2 Dataset Introduction

This paper uses a dataset open on Kaggle. The author crawled the data on Steam and Steam Spy using related APIs in May 2019, and made a preliminary data cleaning including removing non-game items (e.g., software). In summary, the dataset used in this paper has 27075 samples, of which 20288 have the user-generated tag “indie”. The class of each indie game, or the owner of a game, ranges from 0–20k to 20m–50m, which
contains 11 intervals in total. The class distribution of the samples is illustrated in Figure 3.1.

![Figure 3.1 The distribution of classes in the dataset](image)

Although it can be treated as a multiclassification task, the distribution of different classes is highly skewed. Many classes, especially those which consist of the relatively popular games, have too few samples to be statistically meaningful. According to the research about the sample size of machine learning problems, depending on the data set and sampling method, it took between 80 to 560 annotated samples to achieve mean average and root mean squared error below 0.01 (Figueroa et al., 2012, p. 1). Therefore, this paper will combine together the classes that have more than 20 thousand owners. In other words, the samples would only have binary labels, which are either less than or more than 20 thousand. After combing the classes, 27.7% of the indie games fall into the class “≥20k” (negative class). The features involving in the dataset could be grouped into
two categories. Section 3.3 discusses the context features, and Section 3.4 discusses the text features.

3.3 Context Features

Game context information such as genres, languages, prices, art styles is crucial for customers since they would not pay the game not fitting into their tastes. Steam has a standard layout that helps to show game context information. Figure 3.1 – 3.3 illustrates how different context sections are displayed on the Steam store (The game overview has a single page layout so Figure 3.1 – 3.3 are screenshots of different parts on the same page).

![Image of Steam game overview](image)

*Figure 3.2 The overview page of Skul: The Hero Slayer on Steam (top part)*

The top part of the page (Figure 3.2) is the thumbnail of the game, which contains game name, image/video slides, a short introduction, game review stats, release date, game...
developer and publisher, and user-generated game tags. Image and video slides should be critical for prediction since they reveal not only art styles but also the gameplay (e.g., the tactical aspect of the game). However, it is quite difficult to convert them into the features used in the machine learning models. Alternatively, this paper relies on user-generated tags to represent the art style and the gameplay, which consist of words and phrases (e.g., “pixel graphics” represents the art style; “action roguelike” represents the gameplay). They can be treated as a special version of text features, together with other context features to be put into the model.

The game review stats play an important role in the game purchase because it reflects the opinions from other players. Steam has 9 levels of ratings from overwhelmingly positive to overwhelmingly negative. Each level has a corresponding percentage range for positive ratings (e.g., overwhelmingly positive means 95-99% of the game reviews are positive). This paper uses the percentage as a numeric feature.
Figure 3.3 The overview page of Skul: The Hero Slayer on Steam (middle part)

The middle part of the page (Figure 3.3) has a two-column layout. The left column consists of different purchase options, and concise reviews from the media. The right column shows a list of context information, including some characteristics not covering in Figure 3.1, such as languages and playing platforms supported by the game. The game genres stated by Steam would not be considered in this paper because user-generated tags serve the same purpose in most cases. The Steam store may not be able to offer particularly accurate categories considering the organization of the game library.
However, the classification did by the players has not such limit and could be more useful for the prediction. For instance, “Action roguelike” can be regarded as the subclass of the action genre.

![Figure 3. 4 The overview page of Skul: The Hero Slayer on Steam (bottom part)](image)
The bottom part of the page (Figure 3.4) contains the game introduction written by the developers or publishers in detail. The introduction would be used as the corpus for textual features.

3.4 Textual Features

This paper uses the bag-of-words model to convert the corpus into text features, and only uses unigrams without considering parts of speech. To filter out the most and least frequent words, this paper will only consider the words appearing in less than 95 percent of the samples and at least two samples. Rather than word counts, this paper uses the TF-IDF value for each textual feature. TF-IDF is a measure that balances word relevance between one document and the entire corpus, by multiplies term frequency (TF) and inverse document frequency (IDF). Term frequency, in the simplest way, is the raw count of a term appearing in a document. Inverse document frequency is calculated as:

\[ idf(t) = \log \frac{1+n}{1+df(t)} + 1, \]

where \( n \) is the number of documents in the document set, and \( df(t) \) is the number of documents in the document set that contains the term.

3.5 Machine Learning Algorithms

Given the classification task for popularity prediction, the machine learning algorithms involved in this paper are logistic regression and random forest, which perform fairly well in various domains.
Logistic regression uses a logistic function to model the probability of a binary dependent variable. The logarithm of the odds is used to map the probability values from \((0, 1)\) to \((-\infty, +\infty)\). If we denote the probability of being the positive class for a sample as \(p\), the logistic function could be written as:

\[
\log \frac{p}{1-p} = \beta_0 + \sum_{i=1}^{n} \beta_i x_i,
\]

where \(x_i\) represents the independent variable and \(\beta_i\) represents the corresponding coefficient.

Random forest is a machine learning model that consists of a number of decision trees, where leaves stand for class labels and branches represent groups of features. For each decision tree, only a random subset of the features is taken into account when splitting a node. In addition, the bagging technique is used when selecting samples for building the tree. Based on random features and random samples, random forest has less risk of overfitting than a single decision tree, resulting in better performance. When combing multiple decisions, there are two common voting metrics: hard voting and soft voting. Hard voting is the majority vote of all decision trees while soft voting is the mean predicted class probabilities of all decision trees. Specifically, the class probability of a single tree is the fraction of samples of the same class in a leaf.
4. RESULTS

4.1 Experiment Setup

Although the dataset was already preprocessed with basic data cleaning, it needs other preliminary modifications to be able to use for building the machine learning model, such as converting the class labels, merging context features and text features together, etc. Thus, this paper uses Python as the major programming language to conduct the experiment, from the feature extraction to the final model building and testing. Python has a number of high-quality libraries that support predictive analytics fairly well. This paper uses Pandas to store the dataset and experiment results, Scikit Learn to acquire the hands-on machine learning related APIs, and Matplotlib to visualize the model performance.

According to the dataset, context features consist of 372 different user-generated tags, the language support (English or not), the price, and the proportion of positive reviews. After filtering out 5% of the most frequent words and the words appeared only in one sample, 29984 text features are incorporated into the model
4.2 Evaluation Metrics

This paper mainly uses two different evaluation metrics: accuracy and F-score. Accuracy is widely used in classification tasks to measure how well the model correctly identifies the unseen samples. The formula of accuracy in binary classification is:

$$Accurac\ y = \frac{TP+TN}{TP+TN+FP+FN},$$

where TP is the number of samples that are correctly predicted as positive (True positive); TN is the number of samples that are correctly predicted as negative (True negative); FP is the number of samples that are erroneously predicted as positive (False positive); FN is the number of samples that are erroneously predicted as negative (False negative).

F-score takes both precision and recall of the class into account. We could use a parameter $\beta$ to represent the importance of recall. The formula is shown below:

$$F_\beta = (1 + \beta^2) \cdot \frac{\text{precision} \cdot \text{recall}}{(\beta^2 \cdot \text{precision}) + \text{recall}}$$

In general, the value of $\beta$ is 1, which means the recall is as important as the precision.

4.3 Hyper Parameter Tuning

The dataset is split into the training set and the test set with the ratio 8:2. To avoid overfitting, the training set is used for hyper parameter tuning as well. This paper adopts a 3-fold cross-validation and the grid search technique, in order to find the optimal
combination given the sets of parameters. F-score would be used as the evaluation metrics to take both precision and recall into consideration.

The hyper parameters of random forest tuned are listed below:

- `n_estimators`: the number of trees in the forest.
- `criterion`: the function to measure the quality of a split.
- `max_depth`: the maximum depth of the tree.
- `min_samples_split`: the minimum number of samples required to split an internal node.
- `min_samples_leaf`: the minimum number of samples required to be at a leaf node.
- `max_features`: the number of features to consider when looking for the best split.

Figure 4.1 – 4.6 show the tuning result for each parameter, meanwhile the other parameters are set by default values. The best combination is listed in Table 4.1, and the corresponding F-score reaches 0.8514, while the default combination has the value 0.8498.
Figure 4.1 criterion

Figure 4.2 min_samples_leaf

Figure 4.3 min_samples_split

Figure 4.4 n_estimators

Figure 4.5 max_depth

Figure 4.6 max_features

criterion | entropy
--- | ---
min_samples_split | 2
min_samples_leaf | 1
n_estimators | 50
max_depth | None
max_features | Sqrt (about 174)
F-score | 0.8514

Table 4.1 The best combination of random forest parameters
The hyper parameters of logistic regression tuned are listed below:

- tol: tolerance for stopping criteria.
- C: inverse of regularization strength.

Figure 3.7 shows these two parameters’ tuning results. When the tolerance is less than 10, the F-scores keep almost unchanged for any given C, while they grow or drop drastically with the value of 100. The best combination is listed in Table 4.2, and the corresponding F-score is 0.8683. As a comparison, the default combination’s F-score is 0.8678.

![Figure 4.7 The tolerance and C tuning result](image)

<table>
<thead>
<tr>
<th>C</th>
<th>0.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tolerance</td>
<td>0.001</td>
</tr>
<tr>
<td>F-score</td>
<td>0.8683</td>
</tr>
</tbody>
</table>

*Table 4.2 The best combination of logistic regression parameters*
4.4 Performance Evaluation

With the optimal parameter combinations, both logistic regression and random forest algorithms are evaluated by accuracy and F-score (Table 4.3). This paper also digs into the confusion matrix of each algorithm and compares their prediction details.

<table>
<thead>
<tr>
<th>Model</th>
<th>Accuracy</th>
<th>F-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline (majority guess)</td>
<td>0.7227</td>
<td>0.8390</td>
</tr>
<tr>
<td>Random Forest</td>
<td>0.7797</td>
<td>0.8670</td>
</tr>
<tr>
<td>Logistic Regression</td>
<td>0.8021</td>
<td>0.8736</td>
</tr>
</tbody>
</table>

*Table 4.3 The performance of two algorithms compared with the baseline.*

Both logistic regression and random forest predict the popularity of indie games better than the majority guess. The confusion matrixes of each algorithm (Figure 4.8) show the differences in their predicting behavior. Random forest tends to predict the sample as the class of less than 20 thousand owners, which makes it close to the majority guess. While it has a higher recall — only 47 samples in the positive class (less than 20 thousand owners) are predicted as negative, more negative samples predicted positively cause low precision.
4.5 Parameter Importance

This paper also delves into the parameter importance in order to find which features have the most correlation with the popularity of indie games.

4.5.1 Logistic Regression

For logistic regression, the parameter importance refers to the coefficient of the features in the decision function. Table 4.4 shows the top 30 features correlated with the positive class (<20k), and Table 4.5 shows top 30 features correlated with the negative class. Features with the label “T” belong to context features.
From these two tables, we can observe that context features (user-generated tags in most cases) have a large proportion of both positive and negative classes, which indicates merely relying on text features would increase the difficulty of the prediction task. In other words, if we assume the user-generated tags represent the mechanical and artistic aspects of the game pretty well, users would consider these features heavily rather than reading the detailed game introduction. Some tags might reveal whether the game is popular or not. For instance, the “early access” in Table 4.4 represents the game has not fully developed, which is a common publishing mode for indie games on Steam. It allows
games to be public in the early stage, giving the game developers and publishers opportunities to make a profit soon and adjust game development based on the reaction from the market. However, it also brings many low-quality games to market and therefore “early access” becomes the second indicative feature for the positive class (<20k).

Some game genres might appeal to only a small number of players, such as “VR”, “text based”, “dark comedy”, “grid based movement”. VR games are expensive to play because they require an extra head-mounted device to display the game graphics. Text based games are more like virtual novels so that they have fewer interactions than traditional video games. Similarly, dark comedy games might consist of more video clips rather than interesting gameplay. Grid based movement games were popular in the past but lost much attention in the last decade due to the development of game technology.

Some features have a strong correlation and they can be discussed together. For example, “free to play” and “loot” in Table 4.5 represent a common business model in the game industry, which allows players to download the game free and start playing, with revenue being generated by microtransactions (loot or loot box is one type of microtransactions). Free games will attract more players in nature so that it ranks the first in the most indicative features for negative class (>20k).

Games that allow multiple players to interact with each other, such as “multiplayer”, “MMORPG” and “local co-op”, have the social attribute, contributing to an easier spread across the players who would like to keep their offline social events in the game world.
“Rogue like” and “open world” are two popular game genres in the past few years, so it is no surprise to see them in Table 4.5. On the other hand, some game genres including “RTS”, “platformer”, “hack and slash”, with a long history though, still stand out to become popular genres, which reflects their exceptional playabilities.

Another finding from the two tables is that stop words may fairly useful for the prediction. Since this paper removes only a small number of stop words (only words appeared in more than 95% of the samples are removed), we can see some typical stop words both in Table 4.4 and Table 4.5, such as “to”, “you”, “if”, “of”.

4.5.2 Random Forest

In the random forest model, the parameter importance is measured by the average decrease of Gini impurity when a variable is selected to split a node in decision trees. It should be noted that the parameter importance of random forest does not reflect the preference of classes. Thus, for categorical features, this paper will use the larger sample count between two classes to infer the model preference. However, we cannot infer the tendency of continuous variables since the split threshold is hidden. Table 4.6 shows 60 features with the highest importance in the random forest model. Similarly, features with the label “T” represent context features. In the class column, “?” represents “undefined” because the corresponding feature is continuous.
Compared with the logistic regression, it is clear that more text features appear in the table. In addition, the context feature “positive rate” and “price” also rank higher. This might result from the fact that continuous variables are preferred in variable selection.

Many researchers observed such preference. For example, Strobl et al. (2007) found that variables with more potential cut points are more likely to produce a good criterion value
by chance, as in a multiple testing situation. Nevertheless, we can find there is an overlap between the two machine learning algorithms. “Free to play” and “multiplayer” have high predictive power associated with the negative class. “Great soundtrack” and “sandbox” also appear in both models.

It is interesting to note that stop words are also considered as indicative features in the random forest model. If removing any stop word, the performance of either model reduces slightly, which indicates stop words filtering is not always a good decision for machine learning tasks.

4.6 Error Analysis

Digging into the mislabeled samples could help to find out how bad the model performs for some corner cases. Take Skater XL as an example, it was predicted erroneously by the logistic regression model with a 0.9434 probability to be positive class (<20k). The game introduction used to extract text features is quoted below:

“From the people who brought you Skater, which reached #1 sports game world-wide for mobile, Skater XL is an evolution in skateboarding games. Experience unparalleled board control and responsiveness while you skate legendary real-world skate spots. Style your tricks the way you want, from the way you flip the board to the way you move your feet. Feel connected to your board like never before. Style and Control: With new ground breaking and intuitive controls, skateboarding in a video game has never felt more authentic, fluid, and responsive. Style tricks any way you want and make every trick and line your own. Legendary Spots: Shred real world legendary skate spots, like the LA Courthouse which is featured in early access. IMPORTANT: This game REQUIRES an Xbox 360, One, PS3 or PS4 compatible controller to play. There are NO keyboard controls available. It also requires a DirectX 11 or above graphics card.”
Here is the list of user-generated tags for this game:

<table>
<thead>
<tr>
<th>feature</th>
<th>coefficient</th>
</tr>
</thead>
<tbody>
<tr>
<td>early access (T)</td>
<td>0.7392</td>
</tr>
<tr>
<td>action (T)</td>
<td>-0.0551</td>
</tr>
<tr>
<td>simulation (T)</td>
<td>-0.0466</td>
</tr>
<tr>
<td>skateboarding (T)</td>
<td>-0.0026</td>
</tr>
<tr>
<td>skating (T)</td>
<td>-0.1572</td>
</tr>
<tr>
<td>sports (T)</td>
<td>0.2950</td>
</tr>
</tbody>
</table>

Table 4.7 The tag features of Skater XL

Although this game has several negative-prone tags such as “skating”, “simulation” and “action”, the strength of these variables is too small to outweigh the dominating positive features “early access” and “sports”. Sports game is a relatively unpopular game genre, not to mention skating is a minor category of the sports game. Similarly, the text features contain some negative-prone words, like “evolution”, “unparalleled”, “legendary”, “intuitive”, “authentic”. But the word “move” has a much larger coefficient, as previously shown in Table 4.4.
5. DISCUSSION

5.1 Summary

This paper presented an exploratory experiment for using logistic regression and random forest to predict the popularity of indie games. Both two models outperformed the majority guess, which indicates using machine learning algorithms to predict the game popularity is feasible. The relatively poorer performance for random forest might result from its preference towards continuous variables, which account for the majority among all features. From the confusion matrix, we could observe that random forest predicted more samples as positive, which might be the representation of overfitting on the training set.

This paper also tried to explain the models by the importance of specific features. For logistic regression, the user-generated tags showed high predictive power. In addition, traditional stop words usually ignored by many NLP tasks had positive impacts on prediction. This finding shows for similar predictive analytics the stop words may be always taken into consideration and test the model performance with or without stop words.
5.2 Limitation

This paper did not leverage all possible sets of features to build the model. For instance, there is a large number of game reviews that could be used to extract more text features. The real playing experience from customers might be more useful than the plain positive rating ratio. Also, this paper inferred that user-generated tags could represent the art style and game graphics shown by images from the game gallery. However, game reviews might contain more specific and subtle perspectives of players with regard to various aspects of the game, not only the graphics but also music, stories, characters, etc.

We should never neglect the strength of game communities and the media for the decision of buying a game. Costumers may hear of a game by the advertisement and then make a purchase. Luckily, indie games suffer less from the commercial-related variables since indie developers usually have a very limited budget to propagate their games. The Steam platform itself has a strong community that pushes the most popular games to players, or makes a guess about the customer preferences and then recommends the most appealing game, no matter what type of niche it is. However, these kinds of features are fairly difficult to represent.

5.3 Implication

Indie game developers and publishers could utilize the feature importance as a feasibility report when developing new games. From the example for error analysis, it might be wiser to choose a game genre that does not have many competitors. On the other hand,
we must notice the game industry is keeping developing, and player tastes are keeping changing. Therefore, an unpopular game genre does not necessarily mean a bad business decision, which may attract more attention in the future.
6. CONCLUSION

This paper employs context features and text features extracted from the game introduction page on Steam to perform a binary classification task, where the popularity of indie games is predicted by both logistic regression and random forest algorithms. The experiment results show there is a correlation between these features and the game popularity. Logistic regression outperforms random forest in terms of accuracy and F-score, with more user-generated tags as the most important variables. The feature importance could be used for game development reference to choose practical game genres. Future studies could consider assigning weights to context features and text features, combing more possible features, and tuning the hyperparameters related to bag-of-words to improve the model performance.
7. REFERENCES

About -. (2021). SteamSpy - All the Data about Steam Games.

https://steamspy.com/about


