**Predicting Game Sales on Steam**

Tianyu Qin (A53280978) and Xinyu Huang (A53279031)

***Abstract* With the help of available game data on the Steam platform, we apply Machine Learning Models to predict game sales on the Steam yet to be released.**

1. INTRODUCTION

For game developers, Steam is a video game distribution service developed and maintained by Valve. As on September 2019, Steam has 14.15 million active users[[1]](#footnote-1) and over 30,000 games. It is estimated that by 2017, Steam represents about 18% of global PC game sales.[[2]](#footnote-2) As we can imagine, distributing games on Steam comes with a price. Developers typically face $100 submission fee, 30% cut of their sales value by Valve, and the remaining 70% is subject to tax, publishers cut. Moreover, for a released game to be profitable, developers need to deduct production costs from the last piece of cake.

This is why predicting game sales on Steam is useful. The predicted sales number on Steam can provide a reference of the profitability to developers when they consider the game design before distributing. Given certain features, if their game is unlikely to become popular on Steam, the develop team can adjust their prospect of profit. Furthermore, ML models can tell which feature contributes more to sales and developers can adjust their strategies accordingly.

1. DATASET

In this section, we will conduct a brief overview of dataset. Then we will do the exploratory data analysis. Since our objective is to predicting sales numbers, we will try to understand the relative importance of features and decide which features to include instead of others. Data preprocessing part will describe how we handle the imbalance of our dataset and how to parse game tags and genres into vectors ready-to-use for the ML models.

1. *Dataset Description*

The dataset is gathered by Davis N. from the “Steam Store and SteamSpy APIs”[[3]](#footnote-3) around May 2019. It provides most of aspects of games available on the store, including estimated number of owners, genres, etc. By Games, this dataset means all Games on Steam, not including other software and hardware.

The dataset has games released on Steam between June 1997 to May 2019, each with the following features: appid; name; release\_date; english; developer; publisher; platforms; required\_age; categories (“Multiplayer”, ”Singleplayer”); genres (“Action”, “Adventure”, “Indie”); steamspy\_tags (“FPS”, “Classic”, “Casual”); achievements; positive\_ratings (positive comments given by players on Steam); negative\_ratings (negative comments given by players on Steam); average\_playtime; median\_playtime; owners; price.

1. *Exploratory Data Analysis*

In this part, we will first explore the distribution of the dependent variable “owners” and then discuss the extent in which the rest of variables are correlated with “owners”, i.e. number of sales.

“Owners” is our dependent variable. It is an categorical sale number estimated by Steamspy (“0-20k”, “20k-50k”, etc.). As we can see from the Figure 1, it is highly imbalanced: more than 10,000 games sitting on the shelf of Steam quietly (<20k sales) while only 27 games have sold over 1 million copies. We will discuss this problem later in the preprocessing section.

For categorical variables “english”, “platforms”, and “required\_age”, we drew heat maps (Figure 2-4) illustrating Steam users welcome more games that are cross-platform, English-versioned, and without any age requirement since top-selling games are all with these features. Given too many unique values of the feature “developers”, we apply the binary encoding directly. As for “genres”, and “steamspy tags”, we turn them into vectors and then apply one-hot encoding, the algorithm of which is described in the next section.

For other continuous variables, we drew box plots with their quantiles (Figure 8-12). However, it’s hard to see whether they are correlated with number of owners or which feature is more important. Moreover, since we are predicting the feature owners, which represents how many times a game is purchased on Steam, the features “average\_playtime”, “median\_playtime” will be excluded for further analysis because it is part of information unavailable before the game release. We will also ignore the features “positive\_ratings”, “negative\_ratings” because their number reflects the popularity of a game, which is the result of the number of sales instead of the reason. Therefore, we generate a new feature named “positive\_ratio”

, as shown in Figure 12, reflecting a rating score for a game, which is similar to scores on game critics websites (IGN, Metacritic) easy to get even before the release of that game.

1. *Dataset Preprocessing*

(天宇看这里 你看需不需要这样做)

In this section, I will describe the way in which we balance the data and how we turn tags features into vectors before applying any ML algorithm.

Considering the highly right-skewed distribution of owners, we decide to combine top-selling games and sample games poorly sold. To do this, we assign a new categorical feature named “new\_owners” to each game, with three categories representing the number of sales 1. Less than 500k, 2. Between 500k and 1 million, 3. Over 1 million. From each new sales group, we sample 513 games, which together become our new balanced dataset.

To tackle with tags feature, including “genres”, “steamspy”, we first split and encode these features into individual columns. Then we add these columns to construct a new vector for each game representing their tags. For instance, an entry with genres “Action;FPS;Multiplayer” are divided into vectors with values , , for corresponding columns. Then we add these vectors so that the new genre feature for that game is .

1. PREDICTIVE MODELS
2. *Training Set and Test Set*

We split our dataset by the release date of a game. Our training set consists of games released before July 2018, which is 90.38% of our balanced data. The test set is games after June 30 2018, which is 9.6% of total observations.

1. *Algorithm:*
2. Logistic Regression

Content

1. Support Vector Machine

Content

1. Random Forest

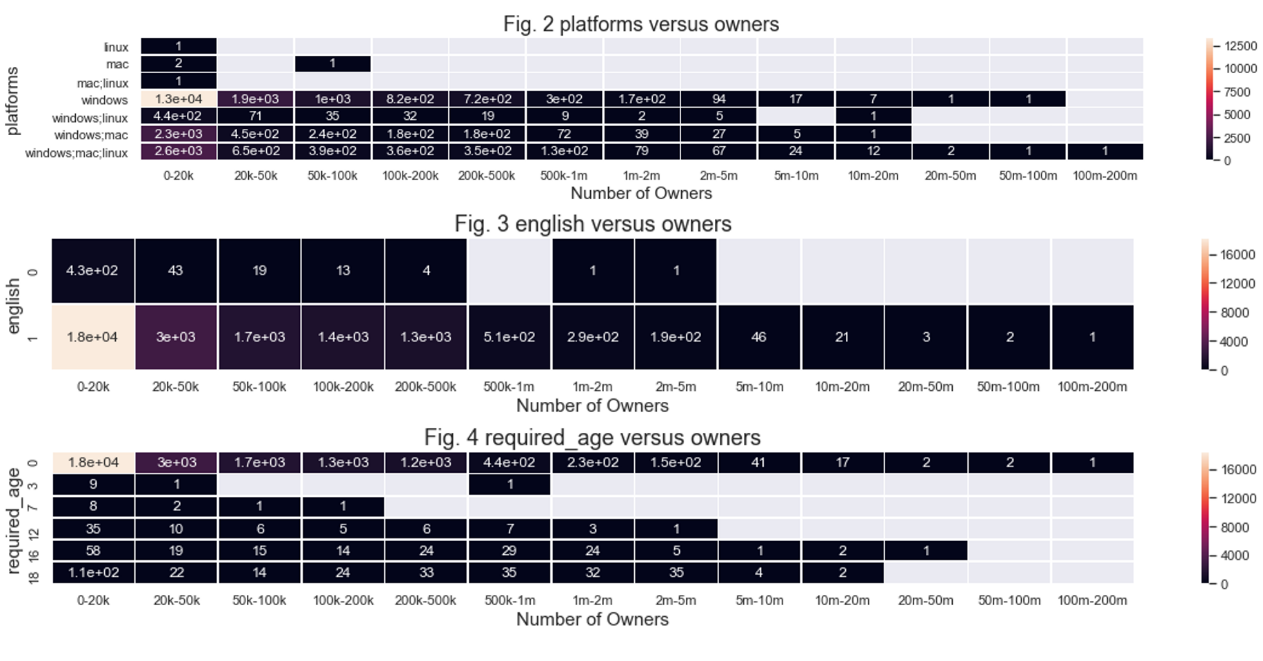
Content

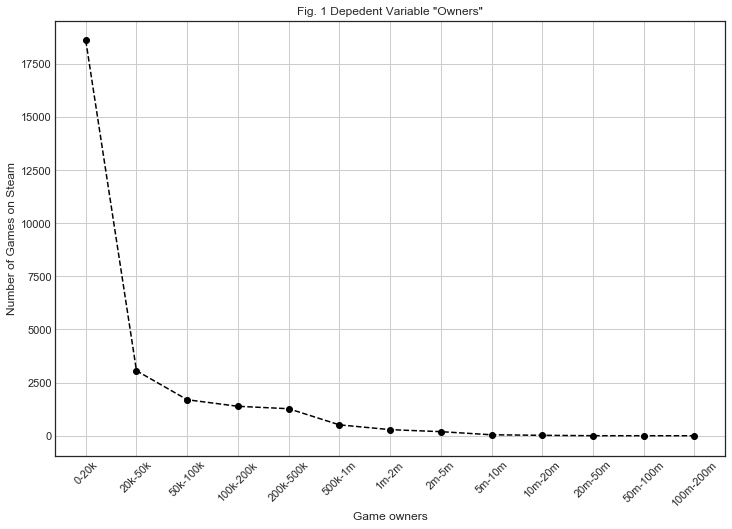
1. Result and Prediction

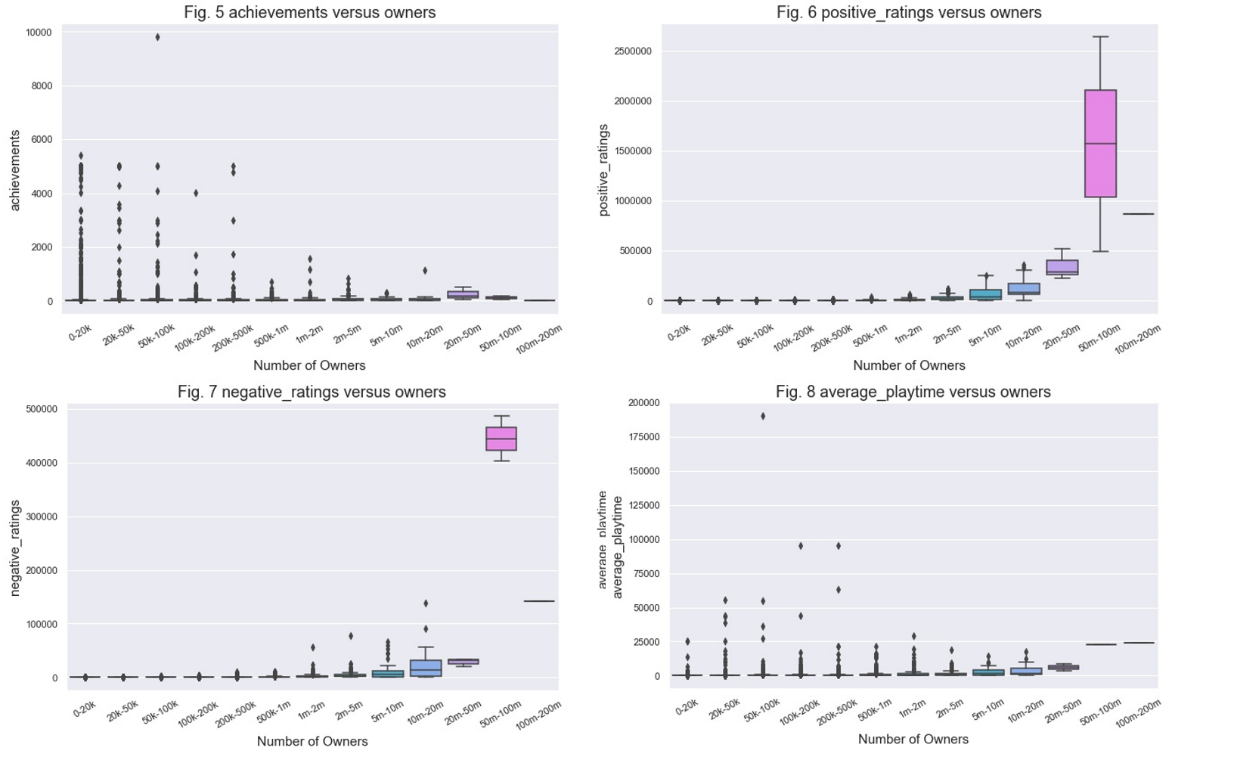
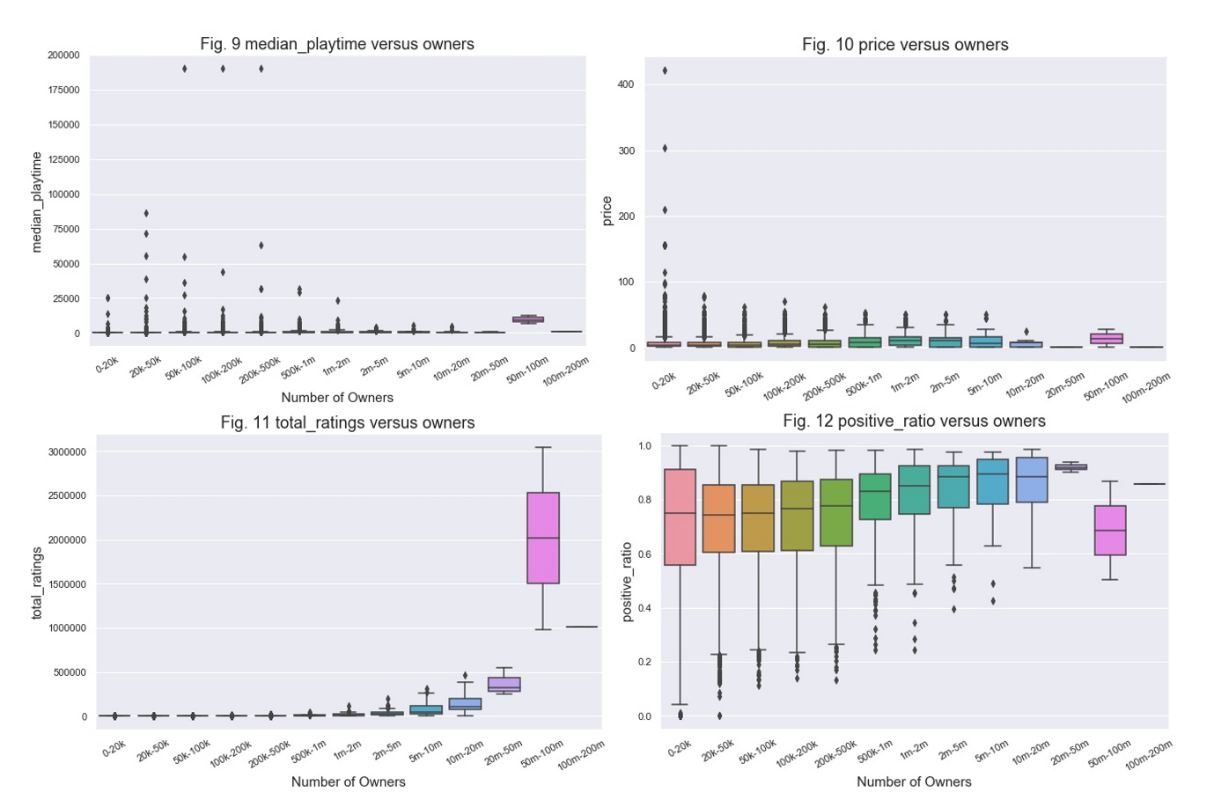
Content

References

APPENDIX (这里最后还要根据需要处理一下，你决定要不要放这么多图，是放在文章中还是放出来)







1. <https://www.statista.com/statistics/308330/number-stream-users/> <https://www.pcgamer.com/steamnow-has-30000-games/> https://en.wikipedia.org/wiki/Steam\_(service) [↑](#footnote-ref-1)
2. <https://partner.steamgames.com/steamdirect> <https://www.quora.com/Valve-company-What-percentage-does-Steam-keep-from-sales> [↑](#footnote-ref-2)
3. https://www.kaggle.com/nikdavis/steam-store-games [↑](#footnote-ref-3)