Doi: 01.0022/j.Interdisciplinary.2023.06.001

More Games, More Playtime

Haochen Zhou

Abstract:

The global electronic gaming market has grown significantly, with 3.198 billion players generating a revenue of $196.8 billion in 2022. Gaming companies must understand their users and keep them engaged to succeed in this market. This article examines whether Steam, one of the top gaming enterprises, truly understands its users. Using one year's worth of data from 12393 players who made 200,000 purchases and played games on Steam, we conducted a k-means clustering analysis to identify four distinct user categories. Based on our findings, we suggest personalized services that Steam can offer to improve user stickiness.

Keywords: Data Cleaning, Video Game, Playtime, Game sale

1 Introduction

Electronic games are gradually becoming an indispensable form of entertainment for people and have even been called the "ninth art," forming a huge industry. The "2022 Global Gaming Market Report" compiled by the well-known gaming market research and data analysis firm Newzoo shows that there were 3.198 billion electronic game players in the world in 2022, creating an of $196.8 billion. In order for many gaming companies to establish themselves in this huge market, they need to keep users firmly attached to their platform, which is based on a deep understanding of users. Steam is one of the top gaming enterprises and operates one of the world's largest PC gaming trading platforms and player communities. However, does it really understand its users well enough? This article analyzes one year's worth of data from 12393 players who made 200,000 purchases and played games on Steam. We further divided these players into four categories with distinct characteristics through k-means clustering from these two dimensions. We also provided suggestions for Steam to provide personalized services and improve user stickiness based on their different categories.

2 Data Cleaning

The data set used in this study was relatively messy, as it recorded users' purchases and gameplay behavior of different games in the same column. One player ID corresponds to multiple records of gameplay and purchase of games. Additionally, the data set includes a column of useless data. Therefore, data cleaning is necessary before further data analysis can be conducted.

```
In [ ]: import pandas as pd
        import numpy as np
        playbehavior=pd.read_csv('steam-200k.csv',header=None)
        playbehavior.columns=['playerID','name','behavior','time','']
        playbehavior
```

Out[ ]:

	playerID	name	behavior	time	
0	151603712	The Elder Scrolls V Skyrim	purchase	1.0	0
1	151603712	The Elder Scrolls V Skyrim	play	273.0	0
2	151603712	Fallout 4	purchase	1.0	0
3	151603712	Fallout 4	play	87.0	0
4	151603712	Spore	purchase	1.0	0
...	...	...	...	...	...
199995	128470551	Titan Souls	play	1.5	0
199996	128470551	Grand Theft Auto Vice City	purchase	1.0	0
199997	128470551	Grand Theft Auto Vice City	play	1.5	0
199998	128470551	RUSH	purchase	1.0	0
199999	128470551	RUSH	play	1.4	0

200000 rows × 5 columns

Data cleaning for this study will be conducted from two perspectives:

(1) Using player lD as the primary key, we will separate each player's game purchase and gameplay behavior from the entire data set and create a separate data set.

(2) Using game names as the primary key, we will separate the sales and total gameplay time of each game within the 12393 players and create a separate data set.

Let's start with the first perspective. We first identified the unique player lD for the 12393 players, which is the core variable for our entire data set. We then recorded their game purchase behavior and gameplay behavior separately. Based on the raw data set, we calculated the total number of games purchased and stored in the Steam

library (inventory), the total gameplay time for all games (time), and the names of all games purchased (name) for each player. For the last item, we concatenated each game name as a string and separated them with commas for further analysis. Finally, we obtained a clean data set, player ID.

In [ ]:
```
## How many players are there
num_player=pd.DataFrame(np.unique(playbehavior['playerID']).tolist())
## build a dataset only includes purchase behavior
onlybuy=playbehavior[playbehavior['behavior']=='purchase']
## Find out how many games does a particular player has
num_game=[]
i=0
for i in range(0,12393):
    count = (onlybuy['playerID']==num_player.iloc[i,0]).sum()
    num_game.append(count)
num_game=pd.DataFrame(num_game)
```

```
num_game
num_Inventory = pd.concat([num_player, num_game],axis=1)
num_Inventory.columns=['playerID','inventory']
#find player's play time
onlyplay=playbehavior[playbehavior['behavior']=='play']
playtime = onlyplay.groupby(['playerID',]).agg({'time': sum,'name':','.join}).re
##build a dataset used playerID as main key
dfplayer= playtime.merge(num_Inventory, on='playerID')
dfplayer=dfplayer.reindex(columns=['playerID','inventory','time','name'])
dfplayer
```

Out[ ]:

	playerID	inventory	time	name
0	5250	21	225.5	Cities Skylines,Deus Ex Human Revolution,Porta...
1	76767	36	1227.0	Counter-Strike,Call of Duty World at War,Total...
2	86540	82	255.0	The Elder Scrolls V Skyrim,Audiosurf,XCOM Enem...
3	144736	8	0.1	Counter-Strike
4	181212	12	2.2	Counter-Strike,Half-Life 2 Lost Coast
...	...	...	...	...
11345	309434439	1	0.8	Dota 2
11346	309554670	1	5.9	Mitos.is The Game
11347	309626088	1	6.7	Age of Empires II HD Edition
11348	309824202	1	0.7	Dota 2
11349	309903146	1	0.2	Dota 2

11350 rows × 4 columns

Next, we will conduct data cleaning from the second perspective. Similar to the previous work, we first counted the names of 3600 different PC games in the raw data set and used it as the primary key (name) forthe new data set. Then, we gathered information on the game sales and total gameplay time of each game
within the 12393 players, saved as variables of sales and time, respectively. Finally, we obtained a new data set, dfgames.

```
In [ ]: ##build a dataset used video games as main key
games=pd.DataFrame(np.unique(playbehavior['name']).tolist())
games.columns=['name']
gametime = onlyplay.groupby(['name']).agg({'time': sum}).reset_index()
gamebuy = onlybuy.groupby(['name']).agg({'time': sum}).reset_index()
gamebuy.columns=['name','sales']
dfgames = gamebuy.merge(gametime, on='name')
dfgames
```

Out[ ]:

	name	sales	time
0	007 Legends	1.0	0.7
1	0RBITALIS	3.0	1.2
2	1... 2... 3... KICK IT! (Drop That Beat Like a...	7.0	20.0
3	10 Second Ninja	6.0	5.9
4	10,000,000	1.0	3.6
...	...	...	...
3595	rymdkapsel	1.0	1.1
3596	sZone-Online	102.0	56.7
3597	the static speaks my name	13.0	2.0
3598	theHunter	248.0	309.2
3599	theHunter Primal	4.0	85.9

3600 rows × 3 columns

More games, more playtime?- Analysis of player behavior data

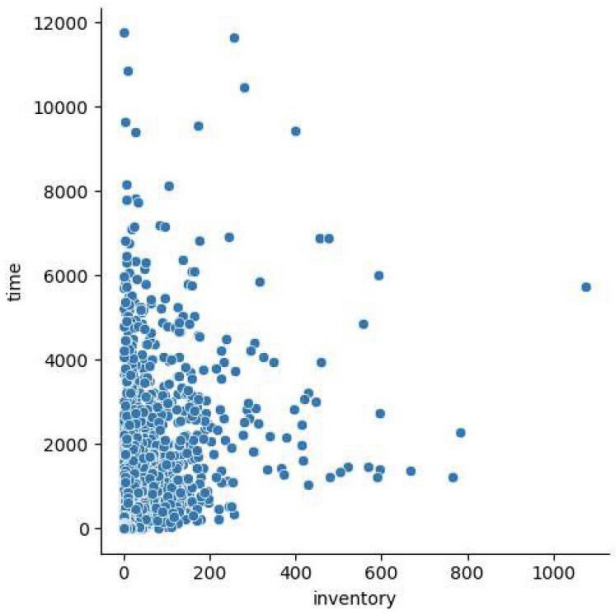
After a long process of data cleaning, we are finally ready to begin data analysis.

Firstly, we conducted an analysis of the player data. Our clean data set, dfplayer, includes two dimensions of data for each player, the number of games purchased and gameplay time. Intuitively, players who purchase more games may spend more time playing them, but is this actually the case?

Before drawing a conclusion, we first plotted a scatter plot of inventory and time.

```
In [ ]: ##data analysis
import seaborn as sns
import matplotlib.pyplot as plt
import matplotlib.style as style

p1 = sns.relplot(
    data = dfplayer,
    x = 'inventory',
    y = 'time'
    )
p1
```

Out[ ]: <seaborn.axisgrid.FacetGrid at 0x2ebfc272e60>



The scatter plot appears to be quite confusing. The number of games purchased by players and their gameplay time does not seem to exhibit a clear positive correlation. Instead, there seems to be a certain group difference in performance among different players in these two aspects. For example, some players purchase very few games but spend a lot of time playing them, while others purchase a large number of games but do not play them much. Inspired by this, we will explore ways to segment players and discover

the characteristics of different types of players, which can serve as references for the personalized services that the Steam platform may offer in the future.

3 Segmenting players using means

We first considered using the mean as the segmentation criterion as it is the most intuitive. After calculations, the average number of games purchased per player is 11.18, and the average gameplay time is303 hours per year. Based on this, we can segment all players into four categories:

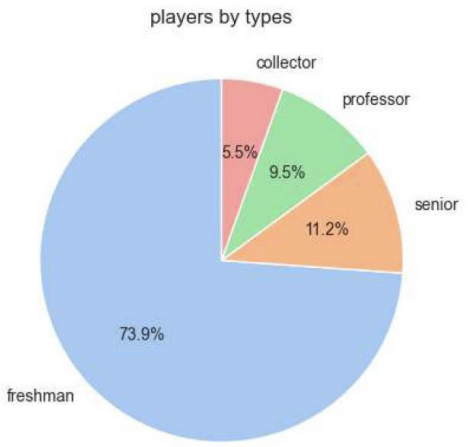
- “freshman”: users who purchase a few games and have low gameplay time, and may not yet be proficient in using the Steam platform.

- “professor”: users who purchase a few games but spend a lot of time playing one or a few games.
- “collector”: users who purchase more games than the average but have low gameplay time, and they mayjust purchase games for collection purposes and not spend too much time playing.
- “senior'”: users who purchase more games than the average and spend more time playing them. They are the top players.

(Note: While writing this, I suddenly realized that segmenting the market based solely on the mean may not be the most scientific way, so I did not continue the analysis using this method. Is this method effective? Please leave me a message.)

```
In [ ]: ##build segments of players by average
time_mean = dfplayer['time'].mean()
inventory_mean = dfplayer['inventory'].mean()
print(time_mean)
print(inventory_mean)
dfplayer.loc[(dfplayer['time'] <= 303) & (dfplayer['inventory'] <= 11), 'segment
dfplayer.loc[(dfplayer['time'] > 303) & (dfplayer['inventory'] <= 11), 'segment'
dfplayer.loc[(dfplayer['time'] <= 303) & (dfplayer['inventory'] > 11), 'segment'
dfplayer.loc[(dfplayer['time'] > 303) & (dfplayer['inventory'] > 11), 'segment']
dfplayer

types = dfplayer['segment'].value_counts()
plt.figure(figsize=(5,5))
sns.set_style("whitegrid")
sns.set_palette("pastel")
plt.pie(types.values, labels=types.index, autopct='%1.1f%%', startangle=90)
plt.title('players by types')
plt.show()
```

```
303.5564581497797
11.180528634361233
```



3. Segmentation with K-means Method

Next, we considered using the K-means method to cluster players and achieve market segmentation. This approach is more accurate and reasonable than simply using averages.

To avoid the influence of dimensional units on the clustering results, we first standardized the data using Z-scores. Then, we classified all players into the same four categories as in the previous section, labeled with different colors in the scatter plot. We also marked the center of each category with a red five-pointed star.

It is worth noting that the number of players in each category varies dramatically.

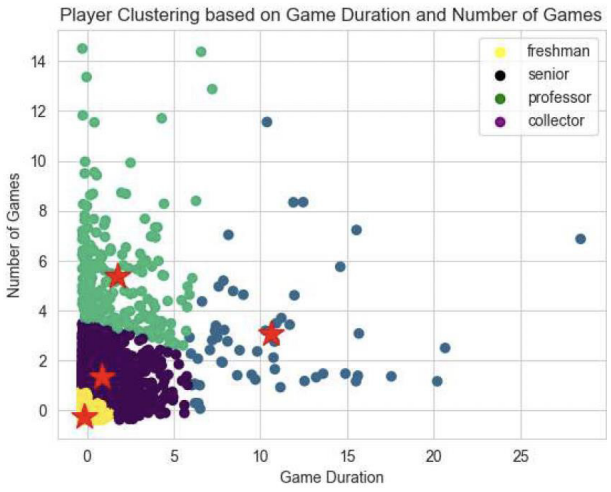
Freshman players constitute the vast majority of all users, accounting for over 88% of the total, followed by senior players with a proportion of nearly 10%. Professor players only make up 1.7%, while collectors have the least representation, accounting for only 0.4%.

This is relatively reasonable, as not everyone can resist playing so many games they purchased.

```
In [ ]: ##build segments of players by k-means method
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
randomseed=666
standard=dfplayer
kmeans = KMeans(n_clusters=4,random_state=randomseed)
X = standard.iloc[:, [1, 2]]
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
standard.iloc[:, [1, 2]] = X_scaled


kmeans.fit(X_scaled)

colors = ['yellow','black', 'green', 'purple']
label=['freshman','senior','professor','collector']

plt.scatter(X.iloc[:, 0], X.iloc[:, 1], c=kmeans.labels_, cmap='viridis')
plt.xlabel('Game Duration')
plt.ylabel('Number of Games')
plt.title('Player Clustering based on Game Duration and Number of Games')
# 添加图例
for col, label in enumerate(label):
    plt.scatter([], [], c=colors[col], label=label)

plt.legend()

plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:, 1], marker
plt.show()

labels = pd.DataFrame(kmeans.labels_)
standard['segment']=labels
types = standard['segment'].value_counts()

dfplayer= playtime.merge(num_Inventory, on='playerID')
dfplayer=dfplayer.reindex(columns=['playerID','inventory','time','name'])
dfplayer['segment']=labels

counts = dfplayer['segment'].value_counts()
proportions = dfplayer['segment'].value_counts(normalize=True)
```

```
structure = pd.DataFrame({'Counts': counts, 'Proportions': proportions})
print(structure)
```

```
c:\Users\天语辰\AppData\Local\Programs\Python\Python310\lib\site-packages\sklearn
\cluster\_kmeans.py:870: FutureWarning: The default value of `n_init` will change
from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the wa
rning
  warnings.warn(
```




```
   Counts  Proportions
3   10010     0.881938
0    1087     0.095771
2     199     0.017533
1      54     0.004758
```

Next, we analyzed the performance of each player type in terms of time and inventory by calculating the mean values, which further demonstrates the characteristics of each category. Freshman players purchased an average of only 4.24 games and played for 89 hours, which is much lower than the overall average we previously calculated. This indicates a significant difference between the roughly 88% of ordinary players and 12% of senior players on Steam. The user stickiness of Steam seems to follow the Pareto principle,with 12% of players making the majority of contributions. As for senior players, their average playtime saw a significant increase to 1378.9 hours, and their inventory was more than ten times that of freshmen, with an average of 43.37 games. It is worth noting that senior players' performance in these two indicators was inferior to professors and collectors, which deviated slightly from our earlier assumptions.

For professors, they spend a staggering 4528.2 hours playing games per year, which means they spend almost half of their time on games. Their inventory was around seven- times the overall average at 77.18. For collectors, they purchased an average of 406 games per year, which is about 40 times the overall average. They also spent an average of 2719.67 hours playing games, which is almost four months of time.

In [ ]:
```
seg0=dfplayer[dfplayer['segment']==0]
seg1=dfplayer[dfplayer['segment']==1]
seg2=dfplayer[dfplayer['segment']==2]
seg3=dfplayer[dfplayer['segment']==3]
seg0.iloc[:,1:3]

mean_value0 = pd.DataFrame(seg0.iloc[:, 1:3].mean()).T
mean_value1 = pd.DataFrame(seg1.iloc[:, 1:3].mean()).T
mean_value2 = pd.DataFrame(seg2.iloc[:, 1:3].mean()).T
mean_value3 = pd.DataFrame(seg3.iloc[:, 1:3].mean()).T
data_list = [mean_value3,mean_value0, mean_value2,mean_value1]
mean_value = pd.concat(data_list)
mean_value['cluster']=['freshman','senior','professor','collector']
mean_value = mean_value.iloc[:, ::-1]
mean_value
```

Out[ ]:

	cluster	time	inventory
0	freshman	89.759381	4.240659
0	senior	1378.932567	43.373505
0	professor	4528.203015	77.180905
0	collector	2719.672222	406.370370

After summarizing the differences between different categories of players in terms of the number of games purchased and the length of time played, we also wanted to know if there were systematic differences among different categories of players in terms of the games they played. Therefore, we calculated the top 10 most frequently purchased games for each player category.

From the table, we can see that Dota 2 was the most common game in the Steam library for freshmen, seniors, and professors. Games such as Team Fortress 2, Unturned, the CS series, L4D2, Gmod, The Elder Scrolls V, and Sid Meier's Civilization V were all popular among freshmen, seniors, and professors players,only with slightly different orders.

However, for the latter two types of players, Terraria and Portal 2 seem to be the games that differentiate them from freshmen, as they pay more attention to these two games in particular.

Collectors are the most unique among all players, as Warhammer 40 is their favorite game. Compared to other players, Borderlands 2 seems to be more in line with their taste.

In [ ]:
```
player_type=0
top_games0 = dfplayer[dfplayer['segment'] == player_type]['name'].str.split(',',
top_games0['rank'] = range(1, len(top_games0) + 1)
top_games0 = top_games0[['rank', 'name', 'frequency']]

player_type=1
top_games1 = dfplayer[dfplayer['segment'] == player_type]['name'].str.split(',',
top_games1['rank'] = range(1, len(top_games1) + 1)
top_games1 = top_games1[['rank', 'name', 'frequency']]

player_type=2
top_games2 = dfplayer[dfplayer['segment'] == player_type]['name'].str.split(',',
top_games2['rank'] = range(1, len(top_games2) + 1)
top_games2 = top_games2[['rank', 'name', 'frequency']]

player_type=3
top_games3 = dfplayer[dfplayer['segment'] == player_type]['name'].str.split(',',
top_games3['rank'] = range(1, len(top_games3) + 1)
top_games3 = top_games3[['rank', 'name', 'frequency']]

topgames=pd.concat([top_games3['name'],top_games0['name'],top_games2['name'],top
topgames.columns=['freshman','senior','professor','collector']
topgames
```

Out[ ]:

	freshman	senior	professor	collector
0	Dota 2	Dota 2	Dota 2	Warhammer 40
1	Team Fortress 2	Counter-Strike Global Offensive	Counter-Strike Global Offensive	Left 4 Dead 2
2	Unturned	Team Fortress 2	Team Fortress 2	Team Fortress 2
3	Counter-Strike Global Offensive	Left 4 Dead 2	Left 4 Dead 2	The Elder Scrolls V Skyrim
4	Counter-Strike Source	The Elder Scrolls V Skyrim	Warhammer 40	Borderlands 2
5	Counter-Strike	Garry's Mod	The Elder Scrolls V Skyrim	Portal 2
6	Left 4 Dead 2	Unturned	Counter-Strike Source	Dota 2
7	Garry's Mod	Terraria	Portal 2	Terraria
8	The Elder Scrolls V Skyrim	Counter-Strike Source	Sid Meier's Civilization V	Counter-Strike Global Offensive
9	Sid Meier's Civilization V	Portal 2	Garry's Mod	Magicka

4. Attempt at Analyzing the Relationship between Game Salesand Playtime

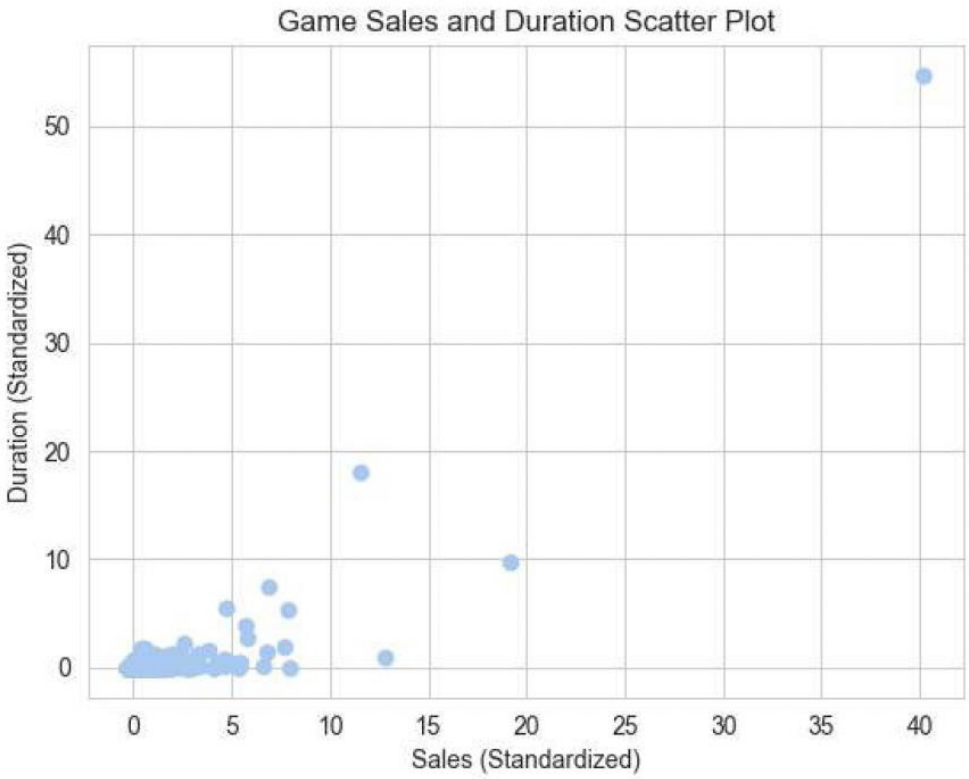
On the other hand, we attempted to explore whether there is a special relationship between game sales and playtime among these sample players using the cleaned dfgames dataset. To do so, we also plotted a scatterplot of standardized data.

However, the plot shows that most games demonstrate consistent performance across these two dimensions,and there is not much value in further analysis.

(Note: I am not sure if it is possible to conduct further

analysis. If so, please leave your ideas in the comments.)

```
In [ ]: scaler = StandardScaler()
df_scaled = pd.DataFrame(scaler.fit_transform(dfgames[['sales','time']]), columr


# 绘制散点图
plt.scatter(df_scaled['sales_scaled'], df_scaled['duration_scaled'])
plt.xlabel('Sales (Standardized)')
plt.ylabel('Duration (Standardized)')
plt.title('Game Sales and Duration Scatter Plot')
plt.show()
```



5. Conclusion

In summary, based on the results of the data analysis, Steam needs to recognize the following facts andtake a series of improvement measures:

1. Steam users consist of approximately 88% shallow users and 12% deep users. There is a significant difference between these two types of users in terms of the number of games purchased and playtime. Deep users are obviously loyal supporters of Steam's business, so it is necessary to establish a more comprehensive membership system. Steam needs to offer more substantial benefits and privileges to these senior players, not just display their Steam level and some childish stickers on their homepage. They should receive more game purchase discounts, free trial hours, and even some precious metal commemorative items to further increase their loyalty.

2. For freshmen users, Steam needs to quickly increase their time spent on the platform and their desire to purchase games. On the one hand, Steam needs to create targeted player communities for the most popular games among them, such as Dota 2, Team Fortress 2, and Unturned, to attract them to interact with other players and extend playtime. On the other hand, Steam can directly recommend a collection of "Top 10 Must-Play Games for Beginners," similar to our summary, for sale at a discounted price. This may be moreeffective than complex personalized recommendation algorithms in stimulating freshmen's purchasing desires and transforming them into senior players.

3. For senior players are the backbone of the Steam platform. Steam needs to fully utilize its rich game

experience and encourage them to write game reviews, guides, or other articles on Steam through physical rewards, thus helping more freshmen to become senior players. Another approach is to establish a “Game Mentor” mechanism that assigns one senior player to provide knowledge-based content, such as game selection and basic game knowledge, to ten freshmen, attracting them to play more games. Mentors who successfully develop freshmen into seniors should be awarded monetary rewards.

4. For professors, their game time seems a bit excessive. My suggestion is to establish an anti-addiction mechanism that limits their maximum game time per day, week, and month to prevent them from being too addicted to games and drifting away from real life. This is also an excellent way for Steam to fulfill its corporate social responsibility.

5. For collectors, they are truly rare species. Steam should establish direct contact with these unique players and allow them to participate in some gray game testing or early access play. They have the most extensiveexperience playing many games and are experts in the gaming field, providing game developers with the most effective suggestions and even optimization plans.

Reference

[1] data source:https://www.kaggle.com/code/fprime00/starter-steam-video-games-cb109076-f%5C

[2] https://newzoo.com/cn/trend-reports/newzoo-global-games-market-report-2022-free-version-cn%5C

[3] Collins,L.(1977). A name to compare with a discussion of the naming of new brands. European Journal of Marketing, 11(5),337- 363.

[4] Klink, R. R.(2003). Creating meaningful brands: The relationship between brand name and brand mark. Marketing Letters, 14, 143-157.