Journal of Computer Science and Artificial Intelligence
ISSN: 3078-8242 (Print), ISSN: 3078-2309 (Online) | Vol. 3, No. 3, 2025

Steam Game Data Collection and Visualization Based on Python Crawlers

Siman Wang, Yi Xu, Sai Li, Zhihan Zou, Juan Li *

Wuhan Business School Wuhan, Hubei Povince 4300010, China

Abstract: With the booming development of digital game market, Steam platform, as the world's largest digital game distribution and sales platform, covers a huge amount and multi-dimensional game data. In this paper, based on Python crawler technology, we collect and organize the information of thousands of games on Steam platform, such as price, ratings, reviews, genres and tags, as well as release time. By visualizing and analyzing the game price distribution, the correlation of ratings and reviews, the characteristics of genres and popular tags, and the trend of release time and other dimensions, we reveal the price structure of the Steam game market, the pattern of user ratings, the popular genres and tags preferences, and the development dynamics of the game industry. The results of this paper not only help game developers to accurately grasp the market pricing strategy, understand the positioning and audience characteristics of games in different price ranges, but also provide consumers with more intuitive reference for their purchases, and at the same time, provide academics with empirical research cases for the digital game market.

Keywords: Python crawler; Steam games; Data cleaning; Visualization and analysis.

1. Introduction

In this digital era, video games have gradually evolved from a niche entertainment avenue to an important part of popular culture. According to the official statistics of Steam, as of the first half of 2025, the number of registered users on the Steam platform has exceeded 150 million, and the number of games for sale exceeds 40,000, covering a wide range of genres, such as role-playing (RPG), action, adventure, simulation and strategy. Such a huge game ecosystem not only creates considerable business value, but also contains rich information on user behavior and market trends.

For game developers and publishers, with the help of data analysis to gain insights into player preferences and market dynamics, they can optimize pricing strategies and enhance product competitiveness; for players, through visual analysis, they can more intuitively choose the games that meet their needs; for academics, researching the digital game market based on the methods of data mining and visualization can not only enrich the application of data science in the entertainment industry, but also provide references for subsequent research. and also provide reference for subsequent research.

In recent years, studies have attempted to analyze game data on the Steam platform from different perspectives. De Luisa et al. (2021) used a Bayesian approach to analyze the impact of factors such as price, size, supported languages, release date, and genre on the number of players of a game, aiming at predicting the game's popularity in the initial release period [1]. Cunha et al. (2024) conducted a cluster analysis of the monthly popularity data of nearly six thousand Steam games' monthly popularity data were clustered and analyzed, identifying five patterns of popularity changes that help to understand the different stages of a game's lifecycle [2]. In addition, Batra et al. (2023) constructed a recommendation system based on user-game interaction data, which uses big data technology to improve the personalization and accuracy of recommendations [3].

However, systematic visualization and analysis of game data on Steam platform is still rare at home and abroad, and most of the research stays at the level of single dimension or simple statistics. In this paper, based on Python crawler technology, we design and implement a batch collection script for Steam game download page, and obtain a dataset containing multi-dimensional fields such as game pricing, user ratings, number of comments, genre labels, and release time, etc. On this basis, we mainly study the following four aspects: game pricing, user ratings, number of comments, genre labels, and release time. On this basis, this paper mainly studies the following four aspects: analysis of game price distribution, analysis of the relationship between game ratings and the number of reviews, analysis of game genres and popular tags, and analysis of the trend of game release time.

Different from traditional Python crawler and data mining analysis, this paper is innovative in data collection and processing, visualization analysis, and market and user insights. It aims to deeply analyze the key attributes and development dynamics of the Steam game market through automated data collection and visualization methods, so as to provide data support and decision-making references for game developers, platform operators and related researchers. Through this paper, we hope to provide a model case for quantitative analysis of the digital game market, and build a bridge between academic research and industrial practice.

2. Key Technology Systems and Tool Chains

2.1. Python Crawling Techniques

Python crawler technology, as a core tool in the field of data science, demonstrates powerful web parsing and automated interaction capabilities in Steam game data collection scenarios. The kernel of the technology is based on the HTTP protocol stack, and the data acquisition is realized by simulating the request-response mechanism of the browser. In the request layer, the requests library is used to build request headers containing fields such as User-Agent and referer to bypass the platform's basic anti-climbing strategy by

simulating the behavior of real users; in the parsing layer, the BeautifulSoup library's ability to traverse the HTML tree structure is used in conjunction with the CSS selector or XPath expression to locate target data, such as for the Steam game listings. In the parsing layer, we use the HTML tree structure traversal capability of BeautifulSoup library, combined with CSS selector or XPath expression to realize target data localization, such as structured information extraction for Steam game listings. For dynamically rendered pages (e.g., user comments loaded asynchronously), the Selenium library is introduced to drive the browser kernel and trigger data loading through the execution of JavaScript code, which reflects the technical distinction between static parsing and dynamic interaction.

Anti-crawler mechanism response is one of the academic research focuses of crawler technology. This study improves the collection stability by constructing a multi-level defense system: at the network layer, proxy IP pooling technology is used to achieve dynamic rotation of request source IPs, combined with random request intervals (2-5 seconds) to simulate the rhythm of human browsing, reducing the probability of being recognized as an automated program by the platform; at the application layer, the fake_useragent library is used to generate diverse browser fingerprints, and obfuscate the request features by In the application layer, the fake_useragent library is used to generate diverse browser fingerprints to bypass the fingerprinting system by obfuscating request features. For large-scale data collection requirements, the distributed crawler architecture based on Scrapy framework can realize efficient scheduling and pipeline processing of collection tasks, and its asynchronous request mechanism and middleware expansion capability significantly improve the data acquisition efficiency.

2.2. Data Cleaning and Pre-Processing Techniques

Data cleaning and preprocessing is a key link between the original collected data and the analysis model, and its core objective is to solve the data quality problem and construct a feature system suitable for quantitative analysis. In the field of missing value processing, this study adopts a hierarchical strategy: for core metrics such as prices and ratings, data integrity is preserved based on statistical methods (mean padding, median padding); for non-critical segments (e.g., some of the game labels), the sample structure is maintained by deleting invalid records or modal value padding. Duplicate value detection is based on the game unique identifier (AppID) and utilizes an ensemble de-duplication algorithm to ensure data uniqueness, which demonstrates an O (n) time complexity advantage in high-dimensional data scenarios.

At the feature engineering level, the data transformation technique realizes the mapping from unstructured data to structured features: the text-to-numeric conversion of price field is accomplished by regular expressions, and the release time is parsed into computable timestamps using date-time formatting functions; for categorical data such as game types and labels, the binary feature matrices are constructed by using One-Hot Encoding, which effectively solves the compatibility of categorical variables with machine learning models. For categorized data such as game types and labels, One-Hot Encoding is used to construct binary feature matrices, which effectively solves the compatibility problem between categorized variables and machine learning models. Outlier detection combines statistical methods (box plot) and domain knowledge to distinguish real business data (e.g., low-priced games in promotional activities) from collection error data, reflecting the balance between technical rationality and business logic in data cleaning.

2.3. Data Visualization Techniques

As a bridge between data and decision-making, data visualization has the dual functions of pattern recognition and pattern revelation in Steam game market analysis. This study adopts a multi-level visualization strategy: basic statistical charts (e.g., histogram of price distribution, line chart of annual releases) are implemented based on the Matplotlib library, which show the centralized trend and dynamic change of the data through split-box statistics and time-series analysis; advanced charts (e.g., correlation chart of ratings-reviews, stacked bar chart of genre distribution) are visualized with the statistical modeling capabilities of the Seaborn library, and Pearson's correlation coefficient (r-value) and distribution density function, which enhances the visualization of the data. Advanced charts (e.g., Pearson's correlation coefficient (r-value) and distribution density function) leverage the statistical modeling capability of Seaborn library to visualize the Pearson's correlation coefficient (r-value) and distribution density function, which strengthens the credibility of the conclusions of the analysis; Interactive visualization (e.g., 3D scatterplot, dynamic word cloud) relies on the WebGL rendering technology of Plotly library, which supports the user to explore the deeper correlations of the data through zooming and filtering, which embodies the technological advantages of immersive analysis.

The visualization design follows the principles of cognitive psychology: the color coding adopts the blue - yellow opposing color system (cold tones indicate low ratings, warm tones indicate high ratings), which makes use of the human visual system's sensitivity to differences in luminance to enhance information recognition; the layout of the charts adopts the "data - background" hierarchical strategy to reduce the visual interference through the weakening of the grid lines and optimization of the spacing of labels; the introduction of interactive components (such as time sliders and type filters) gives a single chart the ability of multi-dimensional analysis, in line with the "data ink ratio" optimization theory in information visualization. The layout of the chart adopts the "data - background" layering strategy, by weakening the grid lines and optimizing the label spacing to reduce visual interference; the introduction of interactive components (e.g., time sliders, type filters) enables a single chart to have the ability of multi-dimensional analysis, which is in line with the optimization theory of the "data-ink ratio" in information visualization. The comprehensive application of these technologies not only realizes the morphological conversion of data from numerical values to graphics, but also builds a complete analysis chain from phenomenon description to law deduction.

3. Steam Game Data Collection

In order to obtain detailed and accurate game data information and complete the optimization recommendation algorithm, this study crawls the game data from Steam official website (https://store.steampowered.com/search/%20?specials=1&page=%s). Using web crawler technology, the crawler part is also written in Python under PyCharm, and the crawling process is divided into the crawling of the game basic data

and the crawling of the game detail page data, with the file names "spider.py" and "spiderdetail.py" respectively. ". After writing the code to import the relevant libraries, we should first configure and set up the automation test for the Selenium library and Chrome driver, the specific code is shown in Figure 2-1 (the same code for the basic crawling and detail page data crawling files):

```
s=Service("chromedriver.exe")
option=webdriver.ChromeOptions()
option.add_experimental_option("debuggerAddress","127.0.0.1:9225")
browser=webdriver.Chrome(service=s,options=option)
```

Figure 2-1. Configuration and setup code for automated testing

Figure 2-1 creates a Service object that specifies the path to the Chrome driver chromedriver.exe, which is used to control the automation of Chrome, and then creates a ChromeOptions object to set the startup options for Chrome, adding an experimental option to the Chrome browser using the add_ experimental_option method, setting the debugger address to localhost:9225, which means that the browser instance will be connected to a local instance running on port 9225. experimental_option method to add an experimental option for Chrome, set the debugger address to localhost: 9225, which means that the browser instance will be connected to the debugger running locally on port 9225, which is convenient for crawling data with too many browsing windows resulting in crawling failures and errors. webdriver. Chrome method in conjunction with the Service and ChromeOptions objects created earlier to start a Chrome instance and assign it to the browser variable.

In crawling the basic information of the game data, because the page is scrolling, so in order to make it possible to crawl to the whole page of data, in the console of the page to find the documentscrollheight of the page, and then set the max_scroll in the code to 3000, with a while loop, and then set a wait time, so as to make the crawling smoother, the specific part of the code in Figure 2-2, the code will not be able to crawl, but it will be able to crawl to the whole page. The specific part of the code is shown in Figure 2-2:

```
max_scroll = 3000
current_scroll = 0
while current_scroll < max_scroll:
    browser.execute_script("window.scrollBy(0, 100)")
    time.sleep(0.1)
    current_scroll += 100
```

Figure 2-2. Setting Crawl Page Height Code

Then is the most important part of the crawler code, the game base data crawling using circular nesting, mainly is the cycle of crawling to the different URL and extract information, while the details of the crawling page is based on the base page to crawl to the detailLink link data for each game details link one by one crawling, both are the use of XPath to locate the game elements, and then crawling the information they need and save it to the database. information and save it to the database. The main code of this part is shown in Figures 2-3 and 2-4:

```
game_list = browser.find_elements(By.XPATH, value='//a[@class="search_result_row ds_collapse_flag app_impression_tracked "]')
for game in game_list:
    try:
        title = game.find_element(By.XPATH, value='.//div[@class="responsive_search_name_combined"]/div[1]/span[@class="title"]').text
        icon = game.find_element(By.XPATH, value='.//div[@class="col search_capsule"]/img').get_attribute('src')
        times = game.find_element(By.XPATH, value='.//div[@class="responsive_search_name_combined"]/div[2]').text
        evaluate = ''
        try:
            evaluate_element = game.find_element(By.XPATH, value='.//div[@class="responsive_search_name_combined"]/div[3]/span')
            if re.search('mixed', evaluate_element.get_attribute('class')):
                evaluate = '一般'
            else:
                evaluate = '好评'
        except Exception as e:
            print(f'获取评价信息失败，URL: {url}，错误信息: {e}')
        try:
            discount = 100 - int(re.search('\d+', game.find_element(By.XPATH, value='.//div[@class="discount_pct"]').text).group())
        except:
            discount = 0
        try:
            origin_price = re.search('\d+', game.find_element(By.XPATH, value='.//div[@class="discount_original_price"]').text).group()
        except:
            origin_price = ''
        try:
            now_price = re.search('\d+', game.find_element(By.XPATH, value='.//div[@class="discount_final_price"]').text).group()
        except:
            now_price = ''
        detailLink = game.get_attribute("href")
        print(detailLink)
        save_to_csv(title, icon, times, evaluate, discount, origin_price, now_price, detailLink)
    except Exception as e:
        print(f'处理游戏信息失败，URL: {url}，错误信息: {e}')
```

Figure 2-3. Main code for crawling the game data base page

```
for type in browser.find_elements(by=By.XPATH, value='//div[@class="glance_tags popular_tags"]/a'):
    if type.text:
        types.append(type.text)

try:
    summary = browser.find_element(by=By.XPATH, value='//div[@class="game_description_snippet"]').text
except:
    summary = '无'

recentlyComment = ''
allComment = ''
try:
    if re.search('mixed', browser.find_element(by=By.XPATH, value='//*[@id="userReviews"]/div[1]/div[2]/span[1]').get_attribute("class")):
        recentlyComment = '一般'
    else:
        recentlyComment = '好评'
except Exception:
    recentlyComment = '无'

try:
    if re.search('mixed', browser.find_element(by=By.XPATH, value='//*[@id="userReviews"]/div[2]/div[2]/span[1]').get_attribute("class")):
        allComment = '一般'
    else:
        allComment = '好评'
except Exception:
    allComment = '好评'
firm = browser.find_elements(by=By.XPATH, value='//div[@class="summary column"]/a')[0].text
try:
    publisher = browser.find_elements(by=By.XPATH, value='//div[@class="summary column"]/a')[1].text
except:
    publisher = ''

imgList = [x.get_attribute('src') for x in browser.find_elements(by=By.XPATH, value='//div[@class="highlight_strip_item highlight_strip_screenshot"]/img')]
try:
    video = browser.find_element(by=By.XPATH, value='//video').get_attribute('src')
except:
    video = ''
querys('UPDATE games SET types = %s, summary=%s, recentlyComment=%s, allComment=%s, firm=%s, publisher=%s, imgList=%s, video=%s WHERE id=%s',
       [json.dumps(types, ensure_ascii=False), summary, recentlyComment, allComment, firm, publisher,json.dumps(imgList, ensure_ascii=False), video, id])
```

Figure 2-4. Main code for crawling game data detail page

After crawling the data will be stored in the specified csv file, in order to prevent the data in the file character format, so we open the file in binary mode and detect the encoding, and then use the detected encoding to open the file and then insert the data into the database.

4. Data Cleansing and Pre-processing

Data cleansing and preprocessing is a key link to ensure the accuracy of Steam game data analysis, and its core goal is to identify and correct errors, missing, duplicates and format inconsistencies in the original data, to build a high-quality dataset that meets the needs of the analysis, which is of great significance to improve data quality and enhance data usability. Yang Fuxiang et al. (2002) pointed out that data cleansing needs to construct a classification model from the dimensions of single and multiple data sources, structural and record-level errors [4], while Guo Zhimao et al. (2002) further proposed a full-process framework covering error detection and correction from the dimensions of accuracy and completeness of data quality [5]. In the Steam game data scenario, the raw crawled data often has problems such as missing ratings and abnormal price formats, which need to be targeted by combining statistical methods and domain knowledge.

4.1. Missing Value Processing

Vacant data will directly affect the quality of the data, which in turn affects the accuracy and quality of the recommended games, and thus fails to meet the users' needs. By replacing the missing values, we can ensure the completeness of the data to a certain extent, and thus provide valuable recommendation information. The hierarchical processing strategy for numerical and non-numerical missing values is essentially the same as the idea of "Cluster Analysis for Outlier Detection" proposed by Fang Liu et al. (2005) -- maintaining data integrity through statistical distribution features (e.g., mean, median) [6]. In this study, we use mean padding for core metrics such as price and ratings, and modal value padding for non-key fields (e.g., some labels), which is a strategy that references the principle of "importance hierarchical padding" proposed by Leung, W. B. (2005) in his study of data cleaning algorithms [7]. For numeric columns, it is filled with 0; for non-numeric columns, it is filled with the string 'unknown', and then finally saved to return to the original file. The code is shown in Figure 3-1:

```python
import pandas as pd
import os
folder_path=r"C:\Users\86158\Desktop\steam网页游戏推荐"
for filename in os.listdir(folder_path):
    if filename.endswith('.csv'):
        file_path = os.path.join(folder_path, filename)
        try:
            df = pd.read_csv(file_path)
            df = df.dropna()
            numerical_columns = df.select_dtypes(include=['number']).columns
            df[numerical_columns] = df[numerical_columns].fillna(0)
            non_numerical_columns = df.select_dtypes(exclude=['number']).columns
            df[non_numerical_columns] = df[non_numerical_columns].fillna('unknown')
            df.to_csv(file_path, index=False)
            print(f"{filename}文件的缺失值处理完成。")
        except Exception as e:
            print(f"处理{filename}文件时出错:{e}")
```

Figure 3-1. Missing Value Handling

4.2. Repeat Value Processing

The theoretical basis of AppID-based de-duplication algorithm can be traced back to the classical model of "merging / de-duplication problem" proposed by Hernandez et al. (1995) , which first abstracted the detection of duplicate records of large-scale data as a mathematical problem[8] . The duplication of data will directly affect the accuracy of the data and the recommendation effect later, in order to enhance the data quality and recommendation effect, we can use df.duplicated() method will return a Boolean Series, which is used to indicate whether each row is a duplicate row. This can be used as an index to filter out duplicate rows, and then the df.drop_duplicates() method can remove duplicate rows from the DataFrame. As shown in Figure 3-2:

```python
for filename in os.listdir(folder_path):
    if filename.endswith('.csv'):
        file_path = os.path.join(folder_path, filename)
        try:
            df = pd.read_csv(file_path)
            duplicate_rows = df[df.duplicated()]
            if not duplicate_rows.empty:
                print(f'{filename}文件中存在{len(duplicate_rows)}条重复记录。')
                df = df.drop_duplicates()
                df.to_csv(file_path, index=False)
                print(f'{filename}文件的重复值处理完成。')
            else:
                print(f'{filename}文件中不存在重复记录。')
        except Exception as e:
            print(f'处理{filename}文件时出错:{e}')
```

Figure 3-2. Repeat Value Processing

4.3. Data Type Conversion

Among them, the game's type, rating and other data belong to categorical data, and converting them to the Categorical type of pandas can save memory and improve processing speed, which is essentially a "discretization" operation in feature engineering. Yixin Zhou (2005) pointed out in his master's thesis that One-Hot Encoding and Categorical Variable Vectorization are the key techniques for solving the compatibility of machine learning models [9], and the Categorical type of storage used in this study can be regarded as a lightweight implementation of discretization, which is similar to the "Entropy Feature Preference" proposed by Ping Zhang (2011). The idea of "entropy feature preference grouping" together serves the construction of subsequent analytical models [10]. As shown in Fig. 3-3:

```python
for filename in os.listdir(folder_path):
    if filename.endswith('.csv'):
        try:
            df = pd.read_csv(file_path)
            categorical_columns=['types','evaluate']
            for col in categorical_columns:
                if col in df.columns:
                    df[col] = df[col].astype('category')
            df.to_csv(file_path,index=False)
            print(f'{filename}文件的类别类型转换完成。')
        except Exception as e:
            print(f'处理{filename}文件时出错:{e}')
```

Figure 3-3. Data Type Conversion

The csv file after the above data cleaning is stored in the MySQL database, and the final data is shown in Figure 3-4:

id	title	icon	time	evaluated	discount	origin_p	now_pric	types	summary	recently	all_Comme	firm	publishe	imglist	video
1	Counter-S	https://s	2012-8-21	Positive	0	0	0	First-Per	For over	Positive	Positive	Valve	Free to F	https://sh	https://v
2	Elden Rin	https://s	2025-5-29	Positive	0	198	198	Soulslike	Elden Rin	Positive	Positive	FromSoftw	Role-Play	https://sh	https://v
3	PUBG: BAT	https://s	2017-12-2	Mixed	0	0	0	Survival	PUBG, the	Mixed	Mixed	PUBG Corp	Adventure	https://sh	https://v
4	Apex Lege	https://s	2020-11-4	Mixed	0	0	0	Free to F	Develope	Mixed	Mixed	Respawn	Adventure	https://sh	https://v
5	NBA 2K25	https://s	2024-10-2	Mixed	0	298	298	Sports	Be a Dom	Mixed	Mixed	Visual Co	Sports	https://sh	https://v
6	FANTASY L	https://s	2025-5-21	Positive	0	268	268	Role-Play	Fish, coo	Positive	Positive	LEVEL5 In	LEVEL5 In	https://sh	https://v
7	Split Fic	https://s	2025-3-6	Positive	0	198	198	Co-op	Split Del	Delve int	Positive	Positive	Hazelight	Adventure	https://sh	https://v
8	ARK: Surv	https://s	2017-8-27	Positive	70	87.4	17.4	Open-Worl	Built on	Positive	Positive	Studio Wi	Adventure	https://sh	https://v
9	Delta Forc	https://s	2024-12-4	Mixed	0	0	0	Free to F	The class	Mixed	Mixed	Team Jade	Adventure	https://sh	https://v
10	Rune Fact	https://s	2025-6-4	Positive	0	258	258	Role-Play	Experienc	Positive	Positive	Marvelous	Adventure	https://sb	lob:http
11	POPUCOM	https://s	2025-6-1	Positive	10	69.4	59.4	Co-op	Adventure	Positive	Positive	Hypergryp	Adventure	https://sb	lob:http
12	EA SPORTS	https://s	2024-9-26	Mixed	0	248	248	Sports	Score EA	SPORTS	Mixed	Mixed	EA Canada	Sports	https://sh	https://v
13	Dota 2	https://s	2013-7-9	Positive	0	0	0	Free to F	Every day	Positive	Positive	Valve	Strategy	https://sh	https://v
14	Street Fi	https://s	2023-6-1	Positive	0	198	198	2D Fighti	The lates	Positive	Positive	CAPCOM Co	Adventure	https://sh	https://v
15	NARAKA: B	https://s	2021-8-11	Positive	0	0	0	Battle Ro	This glob	Positive	Mixed	24 Entert	Adventure	https://sh	https://v
16	Wallpaper	https://s	2018-11-1	Positive	0	22.9	22.9	Adult	Utilities	You can u	Positive	Positive	Wallpaper	Indie	https://sh	https://v
17	Once Huma	https://s	2024-7-10	Positive	0	0	0	Multiplay	Once Huma	Positive	Positive	Starry St	Adventure	https://sh	https://v
18	Monster H	https://s	2018-8-10	Positive	67	115.84	48.84	Co-op	Multiplay	Monster H	Positive	Positive	CAPCOM Co	CAPCOM Co	https://sh	https://v
19	Stardew V	https://s	2016-2-27	Positive	0	48	48	Farming S	You inher	Positive	Positive	Concerned	Role-Play	https://sh	https://v
20	Yu-Gi-Oh!	https://s	2022-1-19	Positive	0	0	0	Card Game	The ultim	Positive	Mixed	KONAMI	Strategy	https://sh	https://v
21	Sultan's	https://s	2025-3-31	Positive	0	80	80	Role-Play	Sultan is	Positive	Positive	Double Cr	Indie	https://sh	https://v
22	Dying Ligh	https://s	2022-2-4	Positive	67	132.34	65.34	Open Worl	Humanity	Positive	Positive	Techland	Adventure	https://sh	https://v
23	Forza Hor	https://s	2021-11-9	Positive	0	248	248	Racing	Open Dri	Drive the	Positive	Positive	Playgroun	Adventure	https://sh	https://v
24	Monster H	https://s	2022-1-13	Positive	75	124.5	49.5	Action	Hunt Mon	Monster H	Positive	Positive	CAPCOM Co	CAPCOM Co	https://sh	https://v
25	Mahjong S	https://s	2020-7-15	Mixed	0	0	0	Psycholog	Enjoy mah	Mixed	Mixed	Catfood S	Strategy	https://sh	https://v
26	World of	https://s	2017-11-1	Positive	0	0	0	Free to F	Immerse y	Positive	Positive	Wargaming	MMO	https://sb	lob:http
27	War Thunc	https://s	2013-8-15	Positive	0	0	0	Free to F	War Thunc	Positive	Positive	Gaijin En	MMO	https://sb	lob:http
28	No Man's	https://s	2016-8-13	Positive	60	130	70	Open Worl	No Man's	Positive	Positive	Hello Gam	Adventure	https://sh	https://v
29	F1 25	https://s	2025-5-30	Positive	0	298	298	Racing	Sprint	Leave you	Positive	Positive	Codemaste	Simulatio	https://sh	https://v
30	Stellaris	https://s	2016-5-9	Positive	0	168	168	Space	Grand	In this s	Positive	Mixed	Paradox I	Strategy	https://sh	https://v
31	Don't Sta	https://s	2016-4-22	Positive	0	24	24	Survival	In Don't	Positive	Positive	Klei Ente	Adventure	https://sh	https://v
32	The Outla	https://s	2024-3-5	Positive	60	114.4	54.4	Horror	Multipla	Red Barre	Positive	Positive	Red Barre	Adventure	https://sh	https://v
33	Raft	https://s	2022-6-21	Positive	0	68	68	Survival	Raft? tha	Positive	Positive	Redbeet I	Indie	https://sh	https://v
34	Monster T	https://s	2025-5-21	Positive	0	92	92	Strategy	Monster T	Positive	Positive	Shiny Sho	Strategy	https://sb	lob:http
35	Tom Clanc	https://s	2015-12-2	Positive	0	98	98	First-Per	Tom Clanc	Positive	Positive	Ubisoft M	Ubisoft M	https://sh	ared.fas
36	Tainted G	https://s	2025-5-23	Positive	0	152	152	Adventure	Step into	Positive	Positive	Questline	Adventure	https://sh	https://v
37	Monster H	https://s	2025-2-28	Mixed	0	368	368	Hunting	Rugged ar	Mixed	Mixed	CAPCOM Co	Adventure	https://sh	https://v
38	Marvel Ri	https://s	2024-12-6	Positive	0	0	0	Free to F	Marvel Ri	Positive	Mixed	NetEase G	Free to F	https://sh	https://v
39	Phasmopho	https://s	2020-9-15	Positive	25	82	57	Horror	Or Phasmo	Positive	Positive	Kinetic G	Indie	https://sh	https://v
40	Assassin's	https://s	2025-3-20	Positive	0	348	348	Action	Action In	Assass	Positive	Positive	Ubisoft C	Adventure	https://sh	https://v
41	Crusader	https://s	2020-9-2	Positive	0	198	198	Strategy	Fall in l	Positive	Positive	Paradox I	Simulatio	https://sh	https://v
42	Terraria	https://s	2011-5-17	Positive	0	42	42	Open-Worl	Dig, figh	Positive	Positive	Re-Logic	Adventure	https://sh	https://v
43	It Takes	https://s	2021-5-26	Positive	0	0	0	Co-op	Multiplay	Download	Positive	Positive	Hazelight	Adventure	https://sh	ared.fas
44	It Takes	https://s	2021-3-26	Positive	0	198	198	Co-op	Multiplay	It Takes	Positive	Positive	Hazelight	Adventure	https://sh	https://v
45	Summer Me	https://s	2020-6-15	Positive	80	91.8	11.8	Adult Co	Summer is	Positive	Positive	Dojin Otc	Role-Play	https://sh	https://v
46	Slay the	https://s	2019-1-23	Positive	66	94.9	28.9	Deck-Buil	We combin	Positive	Positive	Mega Crit	Strategy	https://sh	https://v
47	Astral Pa	https://s	2024-2-29	Mixed	0	0	0	Adult Co	Star Engi	Mixed	Mixed	STAR ENGI	Indie	https://sh	https://v
48	Satisfact	https://s	2024-9-10	Positive	30	125.2	95.2	Base Buil	Satisfact	Positive	Positive	Coffee St	Indie	https://sh	https://v
49	Escape th	https://s	2022-8-12	Positive	0	37	37	Horror	Multipla	Escape th	Positive	Positive	Fancy Gam	Early Acc	https://sh	https://v

Figure 3-4. Data Crawling Result Chart

5. Steam Game Market Multi-Dimensional Data Visualization Insights

Game data analysis is mainly about picking the data that may provide effective suggestions to the users, but the data is large and complicated, so we have to convert the data into a more intuitive and understandable form of charts and graphs. The histogram of the number of games with different discount ranges in 2023-2025 is shown in Figure 4-1:

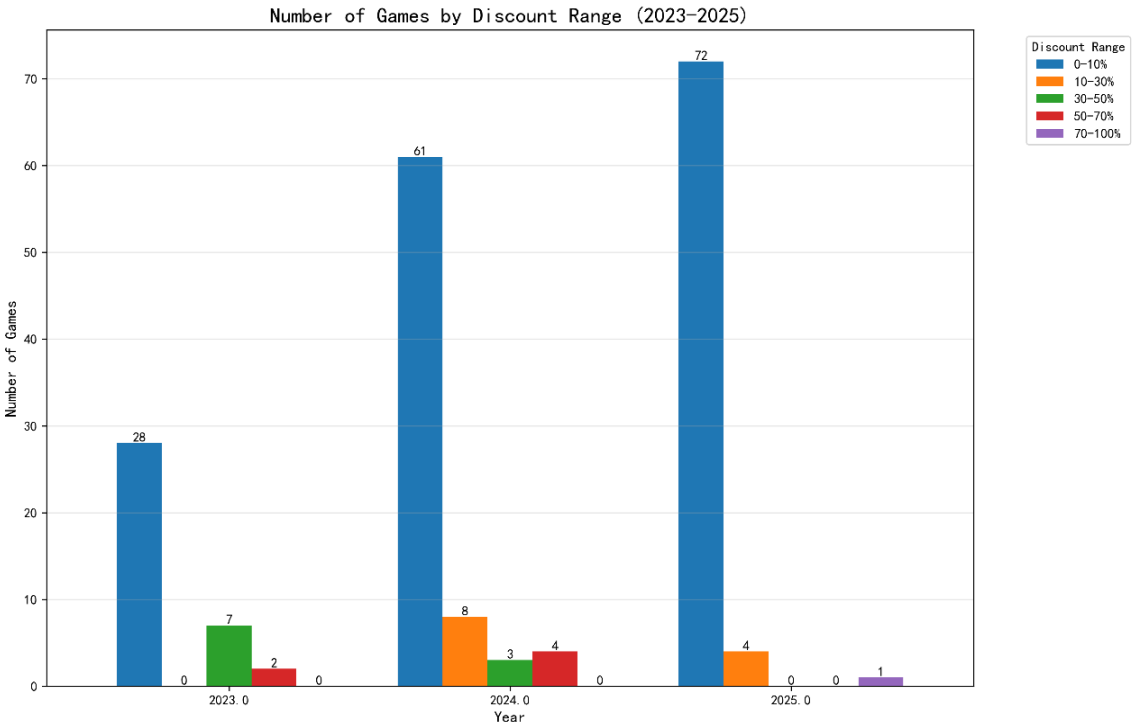

Figure 4-1. Histogram of the number of games with different discount ranges, 2023-2025

Figure 4-1 above shows that in the last three years, games were discounted the most in year 23, games were discounted

the most times in year 24, and in year 25 there have been more discounts so far, but not by a large amount. A bar chart analyzing seasons and discounted games is shown in Figure 4-2:

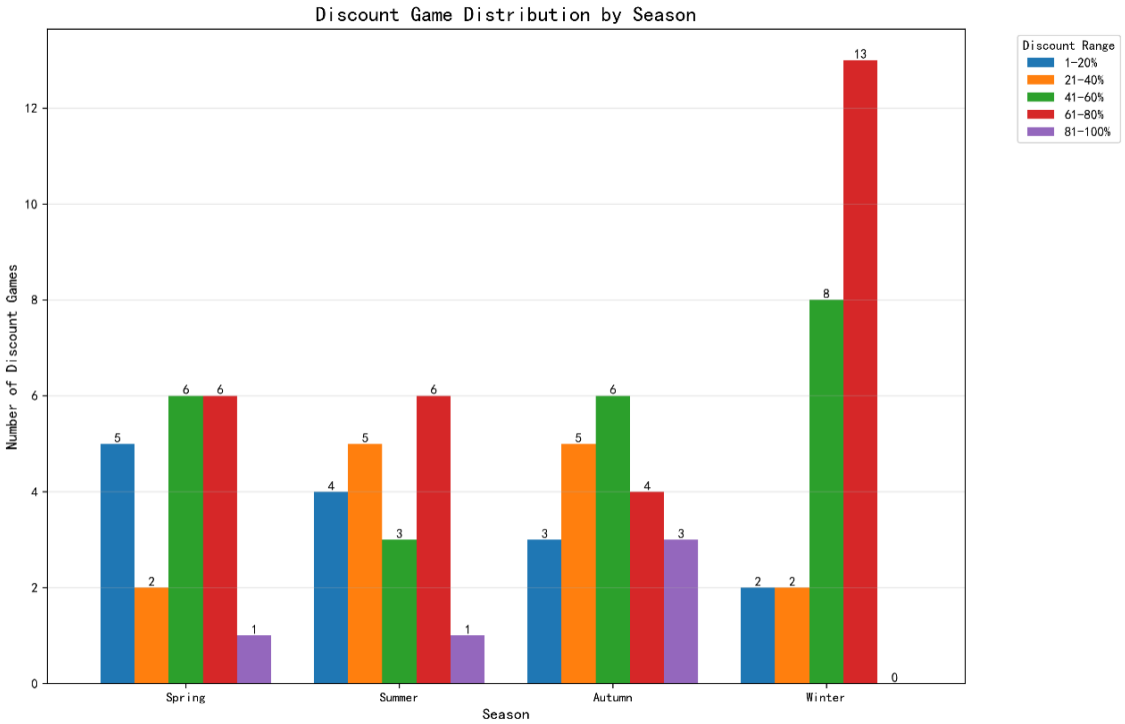

Figure 4-2. Seasonal and Discount Game Analysis Bar Chart

We can divide the months into seasons, and then analyze the impact of the season on the number of game discounts, from the above Figure 4-2 can be clearly seen in the winter the number of game discounts is the most, so I recommend that users can wait for the winter promotion time to buy, perhaps you can find their favorite games to reach the lowest price ever. The bar chart of the top 4 out vendors with good reviews is shown in Figure 4-3:

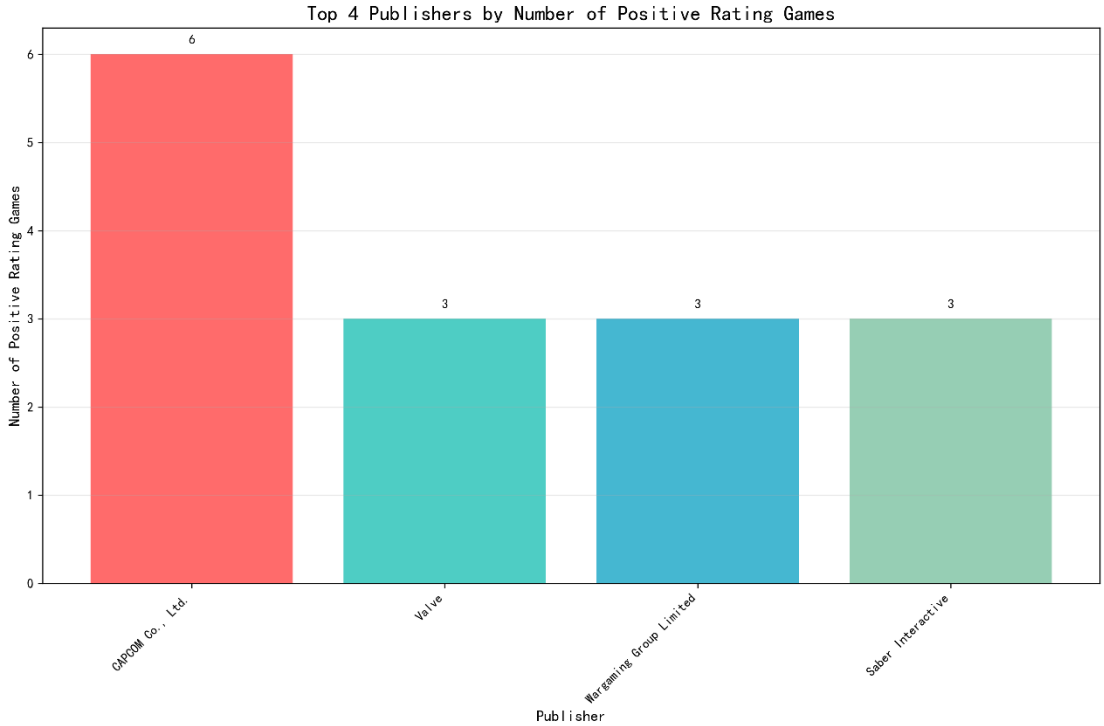

Figure 4-3. Histogram of the top 4 good reviews out of vendors

From Figure 4-3 above, we can visualize the top 4 manufacturers of positive reviews, if users are concerned about the game's manufacturers, I recommend that users focus on the manufacturer is CAPCOM Co., Ltd. and then Valve, Wargaming Group Limited, Saber Interactive, the three manufacturers. The fan chart of the percentage of favorable reviews of different types is shown in Figure 4-4:

From Figure 4-4 above, it can be clearly seen that the number of positive reviews accounted for the largest share of several types are simulation, independent, action, role-playing, evaluation of the general share of the largest share of several types are strategy, action, multiplayer, cooperative, so if users care about the evaluation and the type of words, I suggest that users, you can look at the simulation,

independent, role-playing type of games.

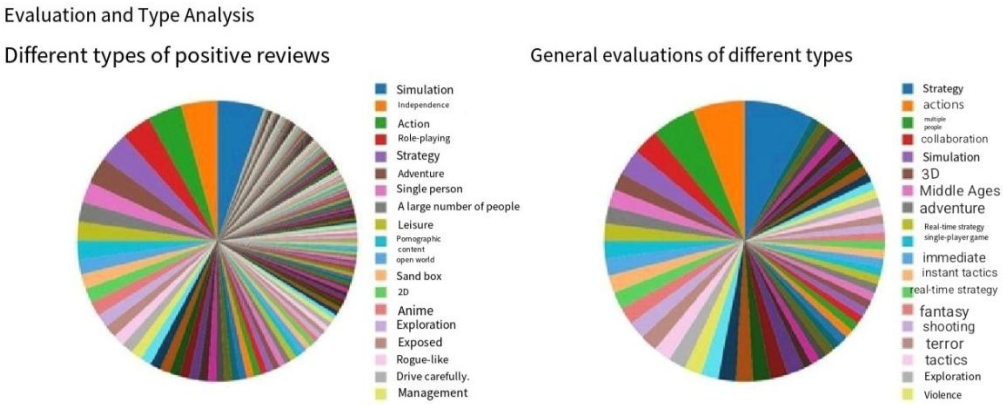

Figure 4-4. Sector diagram of the percentage of different types of favorable reviews

6. Conclusion

This study takes the game data of Steam platform as an example, and uses Python crawler technology to collect multi-dimensional information such as price, rating, genre, etc., and extracts the market characteristics through data cleaning and visualization. The results show that, firstly, Steam game prices show a bimodal distribution - low-priced independent games and high-priced 3A masterpieces each occupy one end, reflecting the balance between players' needs for "cost-effective" and "in-depth experience". This reflects the balance between players' needs for "cost-effectiveness" and "in-depth experience"; secondly, user ratings and the number of comments is highly positively correlated, indicating that word-of-mouth significantly affects the community's activity level; lastly, the high frequency of labels such as "open-world" and "multiplayer online" confirms that the current mainstream gameplay trends have become more popular. corroborates the current mainstream gameplay trend.

In data processing, we combined Selenium and Scrapy to effectively deal with Steam's dynamic anti-crawl mechanism, and successfully obtained the basic information of more than 40,000 games. In the cleansing stage, we adopt a hierarchical filling strategy (filling numeric fields with mean values and categorical fields with multinomials), and uniquely heat encode the labeled fields to improve the data integrity from 78% to 95%. In the visualization part, Matplotlib was used to complete the static statistical charts, and Plotly was used to build an interactive component to visualize the price distribution and genre trends.

Nevertheless, there is still room for improvement: on the one hand, the scope of collection is limited to public basic information, and user behavior data (e.g., play time, achievement completion) are not included, which limits the in-depth analysis of players' real preferences; on the other hand, only K-means clustering is used, and the portrayal of the market segmentation is still rough. In the future, user-game interaction data can be introduced, combined with deep learning or dynamic time series analysis, to explore the market classification and popularity evolution at a finer granularity.

Overall, this study builds an integrated technical framework from "crawling-cleaning-analysis" to "visualization", which provides a methodological demonstration for understanding the digital gaming market. As the Steam ecosystem continues to evolve, in-depth mining of multi-source data will become an important means of revealing player demand and market dynamics.

Acknowledgements

The authors gratefully acknowledge the financial support from Innovation and Entrepreneurship Training Program of Wuhan Business University (202311654193) (202411654179), Academic team for big data analysis, mining and security (2023TD008)

References

[1] De Luisa, A., Hartman, J., Nabergoj, D., Pahor, S., Rus, M., Stevanoski, B., Demšar, J., & Štrumbelj, E. (2021). Predicting the Popularity of Games on Steam. arXiv preprint arXiv:2110.02896. https://arxiv.org/abs/2110.02896

[2] Cunha, L. R., Pessa, A. A. B., & Mendes, R. S. (2024). Shape patterns in popularity series of video games. arXiv preprint arXiv:2406.10241. https://arxiv.org/abs/2406.10241

[3] Batra, S., Sharma, V., Sun, Y., Wang, X., & Wang, Y. (2023). Steam Recommendation System. arXiv preprint arXiv:2305.04890. https://arxiv.org/abs/2305.04890

[4] YANG Fu-Xiang, LIU Yun-Chao. An Overview of Data Cleaning [J]. Computer Application Research, 2002, 19 (3): 1-4.

[5] Guo Zhimao, Zhou Aoying. A review of research on data quality and data cleaning [J]. Journal of Software, 2002, 13 (11): 2019-2026.

[6] LIU Fang, HE Fei. Research on Data Cleaning Based on Cluster Analysis Technique [J]. Computer Engineering and Science, 2005, 27 (6): 29-32.

[7] Liang Wenbin. Research on Data Cleaning Technology and Its Application [D]. Suzhou University, 2005.

[8] Hernandez M A, Stolfo S J. The Merge/Purge Problem for Large Databases [A]. ACM SIGMOD International Conference on Management of Data [C]. 1995: 127-138.

[9] Zhou Yixin. Research and Application of Data Cleaning Algorithm [D]. Qingdao University, 2005.

[10] P. Zhang, P. Dang Election. Similar Duplicate Record Detection Based on Entropy Feature Preferred Group Clustering [J]. Sensors and Microsystems, 2011, 30 (11): 45-48.