The Hebrew University of Jerusalem Business School

06.08.2023

# Final Assignment

55886: Data Science

## Doron Firman

# Table of Content

## Motivation:

From a young age I have loved to play games, it was a great distraction from day-to-day life and gave me a different challenge. As I grew older and bought my first computer, I was introduced to the steam platform. I used the platform as my main gaming platform for the past 12 years or so and saw it grow, evolve, and introduce plenty of new games. With the vast amount of data in their disposal it was interesting for me to explore and gain insight to what makes a game popular and to that extant I have thought of the following business question.

## Business Problem:

What factors influence the peak concurrent players for games or software on the Steam platform?

## Steam Introduction:

Steam is a revolutionary digital distribution platform developed by Valve Corporation in 2003. It has transformed the way people access and enjoy video games, offering a vast library of titles, social features, and innovative tools like Steam Workshop and SteamVR. With millions of active users and a commitment to fostering a vibrant gaming ecosystem, Steam has become the go-to destination for gamers worldwide, providing a seamless and immersive gaming experience.

## What is Peak CCU?

Peak concurrent players on the Steam platform refers to the highest number of players actively engaged in a particular game at the same time. It serves as a metric to measure the success and popularity of a game within the Steam community. The higher the peak concurrent player count, the more successful the game is considered, as it indicates a larger player base, active engagement, and a potentially thriving multiplayer experience. Achieving high peak concurrent player numbers demonstrates a game's ability to captivate and retain players, leading to increased visibility, positive reviews, and potentially higher sales. Example of peak CCU stats can be found in figures [58]-[60].

## Data Review:

For the dataset I used I found a Kaggle dataset that contains approximately 70k rows and 50 columns. The dataset includes games and software released on the platform that were developed and published between 1997 and 2023. As Steam is a global platform some of the entries do not support English at all.

After identifying the relevant dataset, I set out to explore the data. First, I wanted to see how the data looked like as can be seen in Figure [1]. I then explored what were the types of the current columns of the dataset as well as some statistics which can be found in Figures [2] and [3] respectively. Finally, I checked the number of missing values in the table as can be found in Figure [4].

Following the initial discovery I conducted, I decided to drop irrelevant columns and rechecked the steps above.

I will now introduce important columns of the dataset:

- Name (str): The name of the game or software
- Release date (date): The release date of the game or software
- Estimated owners (categorical): A range with the number of owners of a game or platform
- Peak CCU (int): The maximal number of users who played the game at the same time
- Price (int): The original price of the game or software (in US dollars)
- Platform (bool) (Windows\Linux\Mac): Dummy variables that describe which platforms the game or software supports
- Scores (int) (Metacritic\Users): The score assigned in a scale of 0-100
- Playtime (int) (Average\Median): The amount of hours played per user per period
- Developers (str): The name of the game studio\s or people who developed the game
- Publishers (str): The name of the publisher\s of the game or software
- Categories (str): Steam assigned categories
- Genres (str): Developer assigned genres
- Tags (str): User assigned tags for the game or software

## Feature Engineering:

For this part I create several new features, I will now introduce the most important ones:

- Price range (categorical): The price range of the game
- Release by Quarter (categorical): The quarter in which the game or software was released
- Number of Publishers (int): Number of publishers
- Number of Categories (int): Number of categories
- Number of Genres (int): Number of Genres
- Number of Tags (int): Number of tags
- Review Ratio (int): The ratio of positive review out of all reviews
- Price per (x) playtime (int): The price per hour of use

As I have created new features, I have also filled the missing value with the mean average for numerical features and Null for categorical values.

I have also identified several values that appear twice as well as entries which are betas or alphas of games which might impact the accuracy of the data and as such needed to be removed. Furthermore, some columns had to be changed format wise to allow easier manipulation. Finally, to reduce loading times, and to allow aggregation Boolean values were also changed to int32.

Then a recheck was needed to make sure the manipulation was successful and determine the new statistical distribution of the data.

Upon completing this step data visualization was the logical next step.

## Frequency Analysis:

Figures [5] – [20] correspond to the frequency distribution I conducted on different features. Notable figures are [6], [7], [8], [17], [19], [20]. I will now dive into each of figures:

Figure [6]: This plot describes the frequency distribution of estimated owners; we can see that most of the game have between 1-20k owners. We can also see that having above 500k owners is rare with less than 5% of the games.

Figure [7]: This plot describes the yearly maximal amount of peak CCU, we can see a gradual increase over the years. On average excluding 2012, 2013 and 2023 we can expect the maximal peak CCU amount to be at around 200K-300K.

Figure [8]: This plot describes the frequency distribution of price range. We can see that most of the game are priced between 0.1$ to 10$. We can also see that having a price above 20$ is rare with less than 5% of the games.

Figure [17]: This plot describes the frequency distribution of the number of platforms games and software support. We can see most games support a single operating system. We can also see that the number of games supporting 2 and 3 operating systems are relatively equal.

Figure [19]: This plot describes the frequency distribution of release date by quarter of games and software. We can see that the data is evenly distributed, with Q4 having the most releases. We can also see that Q2 has the least releases (although 2023 Q2 and onwards is not in the data)

Figure [20]: This plot describes the number of releases of game and software by year and quarter. We can see by the years a gradual of increase of releases. Currently the number of releases per quarter stands at above 3000.

## Label Analysis:

For this part I wanted to how does the data of peak CCU looks like. For that I have used a histogram and for convenience I used a log scale for the number of peak CCU. We can see that most gams have a low number of peak CCU. The relevant figure for this part is figure [61]

Figure [61]: This histogram describes the frequency of peak CCU in log scale. We can also see that it is very rare to have more than 10k peak CCU.

## Dummy Variables:

For this part I wanted to see if specific categories, tags, and genres impact the peak CCU. Since there are many entries for each column, I chose to focus on the values that appeared the most. I decided at first to create a cutoff of 20 and based on the results I tweaked the number for each. Regarding the ones that didn't make the cutoff I grouped them into a value named 'other' which is visualized as well.

Following that I have created dummy variables for the top 'n' entries and the 'other' as well and added them into the data.

The relevant figures for this part are [21], [22], and [23]. I will now dive into each figure:

Figure [21]: This plot describes the most frequent values that appear in the categories section. We can see that single-player is by far the top category and Steam achievements being the second highest by a big margin. I found that there are a total of 41 unique categories.

Figure [22]: This plot describes the most frequent values that appear in the tags section. We can see that indie is the top tag with the data being very linear. I found that the total number of unique tags is 445.

Figure [23]: This plot describes the most frequent values that appear in the genres section. We can see that once again indie is the top genre but this time by a very big margin. I found that the total number of unique genres is 34.

## Outlier Detection:

For this part I wanted to go over all the numeric features in the data and check for outliers. I used IQR Visualization which means Box Plot with IQR-based Thresholds. I Assumed that the data has a skewed distribution. Also, I assumed that the interquartile range (IQR) is a robust measure of spread. Finally, I assumed that outliers can be identified based on being below Q1 - 1.5 * IQR or above Q3 + 1.5 * IQR. Figures [24] – [49] correspond to the outlier detection I conducted on different features. As we have a big number of entries, and they vary a lot the outliers in this case did not mean there was an error in the data and as such didn't need to be altered or removed. Notable figures are [41], [45] I will now dive into each of figures:

Figure [41]: This plot describes the outlier detection of number of tags; We can see that most of the values are between 3-20 tags. We can also see that the average value is about 9 tags per game or software.

Figure [45]: This plot describes the outlier detection of review ratio. We can that most of the ratios lie between 0.4-0.9 with the average ratio at about 0.7 per game or software. This tells us that in general most games have more positive reviews than negative.

## Model Training:

After completing all the preliminary data tests and cleaning I wanted to see what features might impact the value of peak CCU. For that I chose to use 4 models which are:

- Linear Regression
- Lasso
- Random Forest
- xGradiant Boosting

I wanted to compare the models to figure out what is their success ratio and find the best model to explore optimization. Furthermore, I explored for the best performing models what are their feature importances.

I measured the performance of each model based on the following criteria: R- square, MSE, R-MSE and MAE

Figures [50] – [54] correspond to the results of each model and some feature importances. I will now dive into each of figures:

Figure [50]: This table describes the results of each model in comparison to each other. We can see from the results that RF and xGB performed quite well with xGB being a bit better.

Based on the results of the figure above, I chose to take the RF model (due to time concerns) and try to improve its results using grid search. I then used the same metrics and compared both models.

Figure [51]: This table describes the results of the random forest model compared with the optimized version of random forest.  We can see the big improvement which yields the best overall results compared to all the previous models.

Figure [52]: This plot describes the feature importances of the random forest model. We can see that by far the top leading features are Mac and Linux.

Figure [53]: This plot describes the feature importances of the xGB model. We can see that very similarly to random forest results most of the features are shared between the two.

Figure [54]: This plot describes the feature importances of the optimized random forest model. We can see that very similarly to random forest results most of the features are shared between the two. The only visible change is between positive and number of supported languages.

## PCA and Clustering:

For this part I wanted to take a different approach to examine what might impact peak CCU using a different category of models.

- PCA reduction to 2-dimensions
- Elbow function and silhouette score to determine optimal number of clusters
- Clustering of all the games and platforms

I wanted to explore what information (if any) could be derived from reduction and clustering. Figures [55] – [57] correspond to the results of the different models employed.

Figure [55]: This plot describes the PCA reduction into 2 dimensions.  We can see that there is a positive correlation between PC1 and PC2. We can also see that there are two clusters of data points, one at the bottom left and one at the middle right.

Figure [56]: This plot describes the elbow function I created to find the optimal number of clusters. We can see that for the following elbow function it is hard to determine what is the optimal number of clusters. Using silhouette score I found that 28 clusters has the minimal value.

Figure [57]: This plot describes the mean value of the mean of each feature for the clusters of the data.  We can see 3 clusters with the highest peak CCU which are 10, 22 and 19. Games and software in these clusters tend to have relatively high values for almost all the chosen features, with the exception being “Achievements”. Specific features that are lower include “num_Categories” and “DLC_count” for cluster 19.

A Closer look of the three clusters with the highest value of peak CCU can be found in figure [62].

## Conclusions:

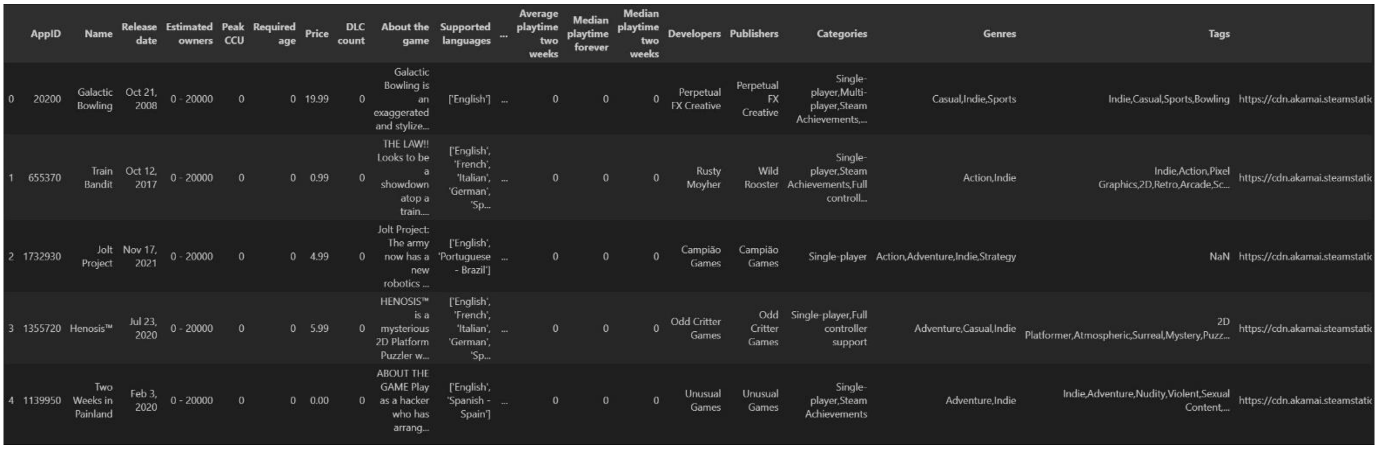
Throughout this project I have examined what impacts peak CCU on the Steam platform for a game or software and identified what developers should prioritize to achieve the highest value of peak CCU. For the first part, publishing for Mac and Linux seems to have the highest effect, as well as number of tags, user rating and reviews. For the second part increasing the number of supported languages and audio seems to be highly effective as well as supporting the game after release in the form of DLC. All of the above require increased investment impacting costs. This suggests that being a AAA game or software could encompass and cover various identified factors.

## Further Research:

Future project might include combining the current data with data from other platforms such as Ubisoft’s Uplay, EA’s Origin, CD Project Red’s GOG and Epic’s Epic games store to receive more accurate picture and see if the conclusions of this research persist. Another approach could be combing the data with console sales. Different data that could be introduced is HLTB (How long to beat) database which has information on the “real” length of a game based on players game time that might expose new information.

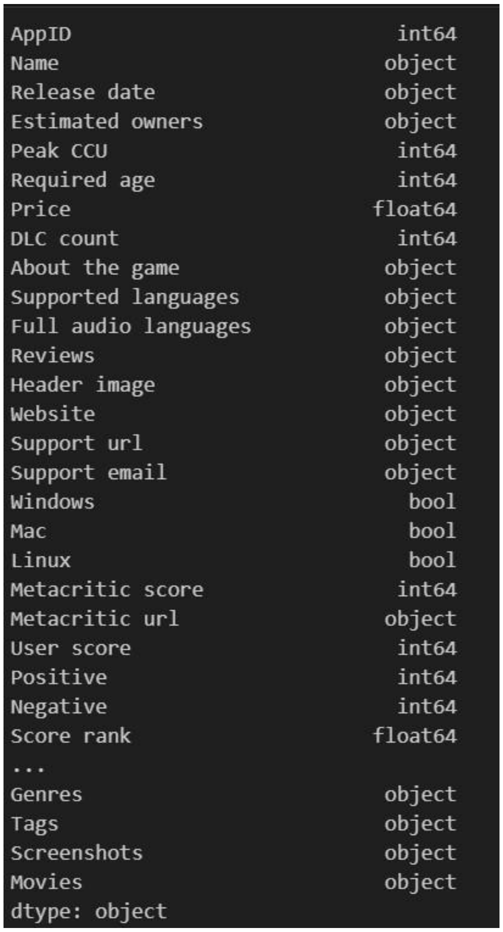
# **Figures:**

Figure [1]: Partial view of df.head()



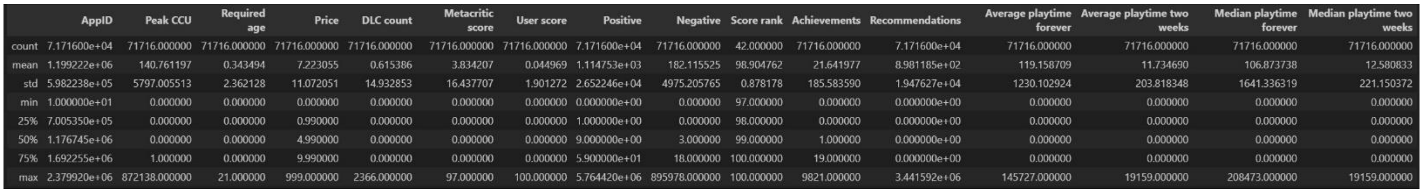
|  | AppID | Name | Release date | Estimated owners | Peak CCU | Required age | Price | DLC count | About the game | Supported languages | ... | Average playtime two weeks | Median playtime forever | Median playtime two weeks | Developers | Publishers | Categories | Genres | Tags |  |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 20200 | Galactic Bowling | Oct 21, 2008 | 0 - 20000 | 0 | 0 | 19.99 | 0 | Galactic Bowling is an exaggerated and stylize... | ['English'] | ... | 0 | 0 | 0 | Perpetual FX Creative | Perpetual FX Creative | Single-player,Multi-player,Steam Achievements,... | Casual,Indie,Sports | Indie,Casual,Sports,Bowling | https://cdn.akamai.steamstatic |
| 1 | 655370 | Train Bandit | Oct 12, 2017 | 0 - 20000 | 0 | 0 | 0.99 | 0 | THE LAW!! Looks to be a showdown atop a train.... | ['English', 'French', 'Italian', 'German', 'Sp... | ... | 0 | 0 | 0 | Rusty Moyher | Wild Rooster | Single-player,Steam Achievements,Full controll... | Action,Indie | Indie,Action,Pixel Graphics,2D,Retro,Arcade,Sc... | https://cdn.akamai.steamstatic |
| 2 | 1732930 | Jolt Project | Nov 17, 2021 | 0 - 20000 | 0 | 0 | 4.99 | 0 | Jolt Project: The army now has a new robotics ... | ['English', 'Portuguese - Brazil'] | ... | 0 | 0 | 0 | Campião Games | Campião Games | Single-player | Action,Adventure,Indie,Strategy | NaN | https://cdn.akamai.steamstatic |
| 3 | 1355720 | Henosis™ | Jul 23, 2020 | 0 - 20000 | 0 | 0 | 5.99 | 0 | HENOSIS™ is a mysterious 2D Platform Puzzler w... | ['English', 'French', 'Italian', 'German', 'Sp... | ... | 0 | 0 | 0 | Odd Critter Games | Odd Critter Games | Single-player,Full controller support | Adventure,Casual,Indie | 2D Platformer,Atmospheric,Surreal,Mystery,Puzz... | https://cdn.akamai.steamstatic |
| 4 | 1139950 | Two Weeks in Painland | Feb 3, 2020 | 0 - 20000 | 0 | 0 | 0.00 | 0 | ABOUT THE GAME Play as a hacker who has arrang... | ['English', 'Spanish - Spain'] | ... | 0 | 0 | 0 | Unusual Games | Unusual Games | Single-player,Steam Achievements | Adventure,Indie | Indie,Adventure,Nudity,Violent,Sexual Content,... | https://cdn.akamai.steamstatic |

Figure [2]: Partial view of df.dtypes()



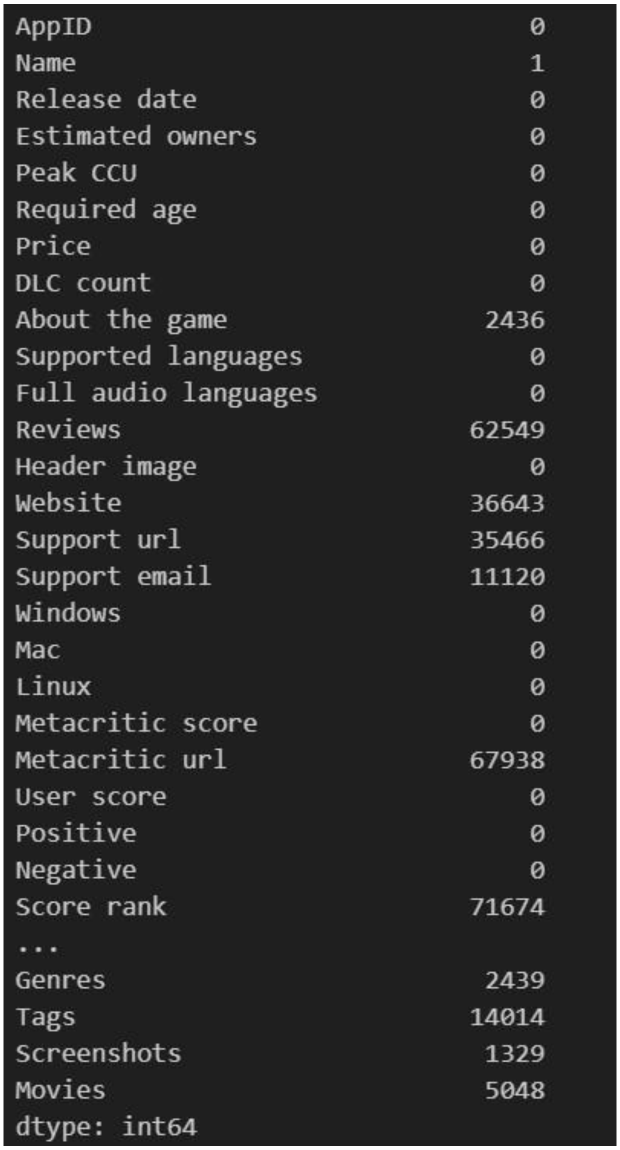
```
AppID                          int64
Name                          object
Release date                  object
Estimated owners              object
Peak CCU                       int64
Required age                   int64
Price                        float64
DLC count                      int64
About the game                object
Supported languages           object
Full audio languages          object
Reviews                       object
Header image                  object
Website                       object
Support url                   object
Support email                 object
Windows                         bool
Mac                             bool
Linux                           bool
Metacritic score               int64
Metacritic url                object
User score                     int64
Positive                       int64
Negative                       int64
Score rank                   float64
...
Genres                        object
Tags                          object
Screenshots                   object
Movies                        object
dtype: object
```

Figure [3]: Statistical description of distribution using df.describe()



| | AppID | Peak CCU | Required age | Price | DLC count | Metacritic score | User score | Positive | Negative | Score rank | Achievements | Recommendations | Average playtime forever | Average playtime two weeks | Median playtime forever | Median playtime two weeks |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| count | 7.171600e+04 | 71716.000000 | 71716.000000 | 71716.000000 | 71716.000000 | 71716.000000 | 71716.000000 | 7.171600e+04 | 71716.000000 | 42.000000 | 71716.000000 | 7.171600e+04 | 71716.000000 | 71716.000000 | 71716.000000 | 71716.000000 |
| mean | 1.199222e+06 | 140.761197 | 0.343494 | 7.223055 | 0.615386 | 3.834207 | 0.044969 | 1.114753e+03 | 182.115525 | 98.904762 | 21.641977 | 8.981185e+02 | 119.158709 | 11.734690 | 106.873738 | 12.580833 |
| std | 5.982238e+05 | 5797.005513 | 2.362128 | 11.072051 | 14.932853 | 16.437707 | 1.901272 | 2.652246e+04 | 4975.205765 | 0.878178 | 185.583590 | 1.947627e+04 | 1230.102924 | 203.818348 | 1641.336319 | 221.150372 |
| min | 1.000000e+01 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000e+00 | 0.000000 | 97.000000 | 0.000000 | 0.000000e+00 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 25% | 7.005350e+05 | 0.000000 | 0.000000 | 0.990000 | 0.000000 | 0.000000 | 0.000000 | 1.000000e+00 | 0.000000 | 98.000000 | 0.000000 | 0.000000e+00 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 50% | 1.176745e+06 | 0.000000 | 0.000000 | 4.990000 | 0.000000 | 0.000000 | 0.000000 | 9.000000e+00 | 3.000000 | 99.000000 | 1.000000 | 0.000000e+00 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 75% | 1.692255e+06 | 1.000000 | 0.000000 | 9.990000 | 0.000000 | 0.000000 | 0.000000 | 5.900000e+01 | 18.000000 | 100.000000 | 19.000000 | 0.000000e+00 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| max | 2.379920e+06 | 872138.000000 | 21.000000 | 999.000000 | 2366.000000 | 97.000000 | 100.000000 | 5.764420e+06 | 895978.000000 | 100.000000 | 9821.000000 | 3.441592e+06 | 145727.000000 | 19159.000000 | 208473.000000 | 19159.000000 |

Figure [4]: Statistical description of distribution using df.isnull().sum()



```
AppID                           0
Name                            1
Release date                    0
Estimated owners                0
Peak CCU                        0
Required age                    0
Price                           0
DLC count                       0
About the game               2436
Supported languages             0
Full audio languages            0
Reviews                     62549
Header image                    0
Website                     36643
Support url                 35466
Support email               11120
Windows                         0
Mac                             0
Linux                           0
Metacritic score                0
Metacritic url              67938
User score                      0
Positive                        0
Negative                        0
Score rank                  71674
...
Genres                       2439
Tags                        14014
Screenshots                  1329
Movies                       5048
dtype: int64
```

Figure [5]: Amount of games by operating system

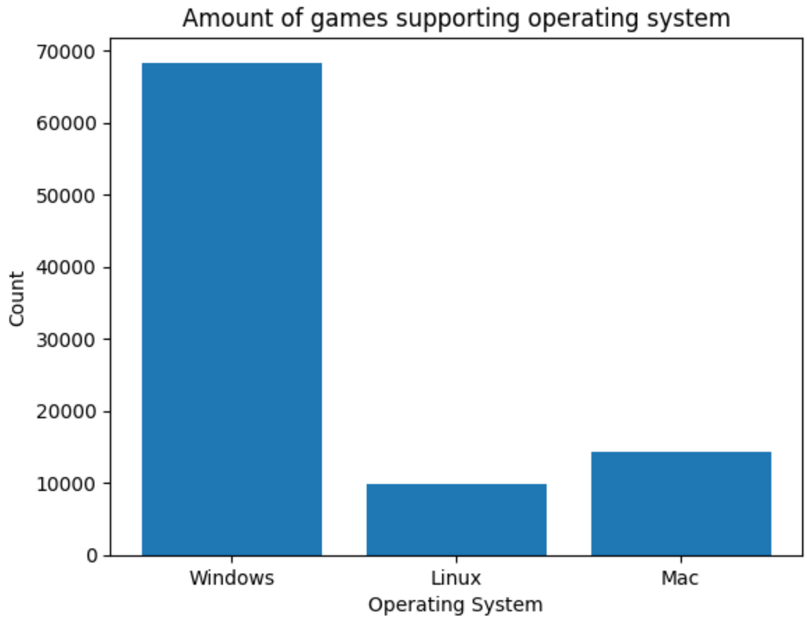


Figure [6]: Frequency distribution of estimated owners

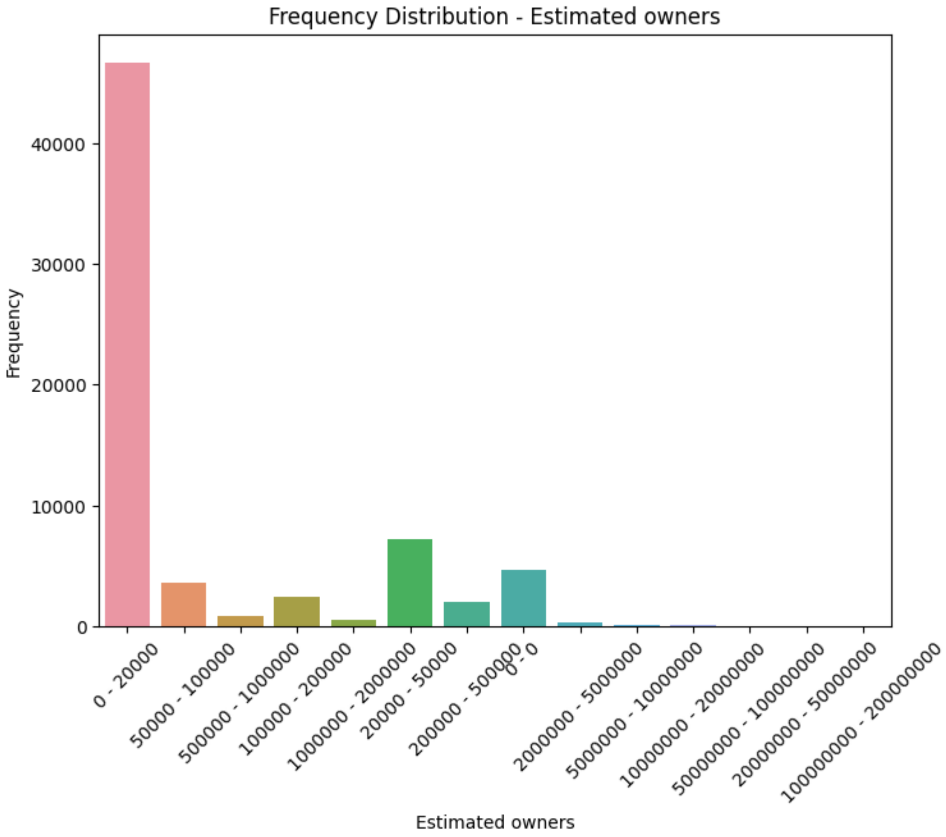


Figure [7]: Max peak CCU by year

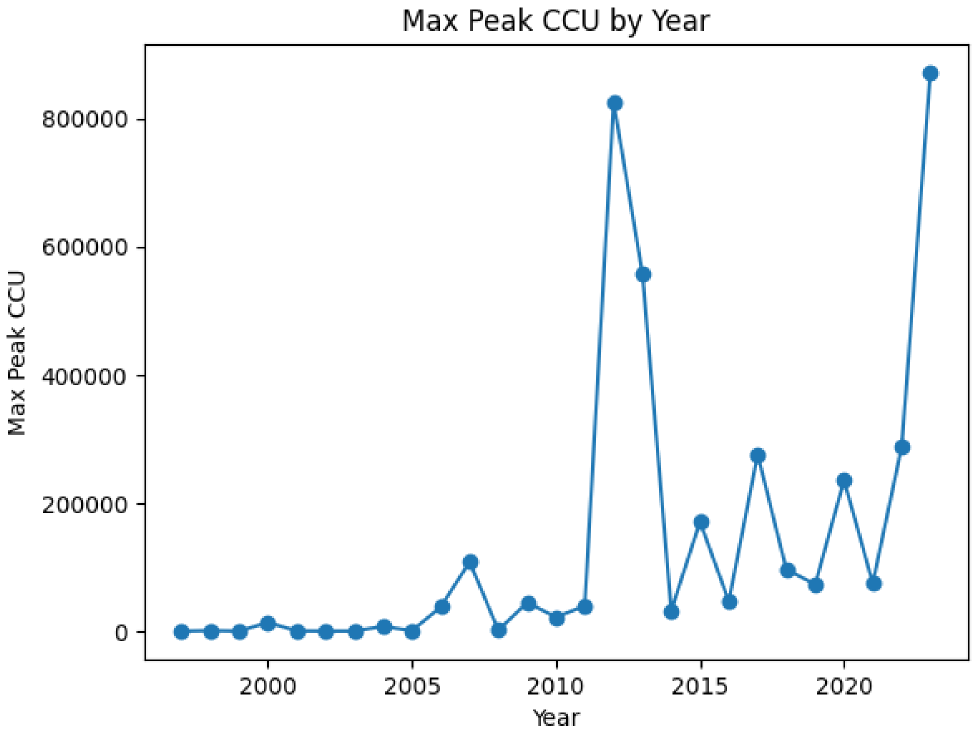


Figure [8]: Frequency distribution of price range

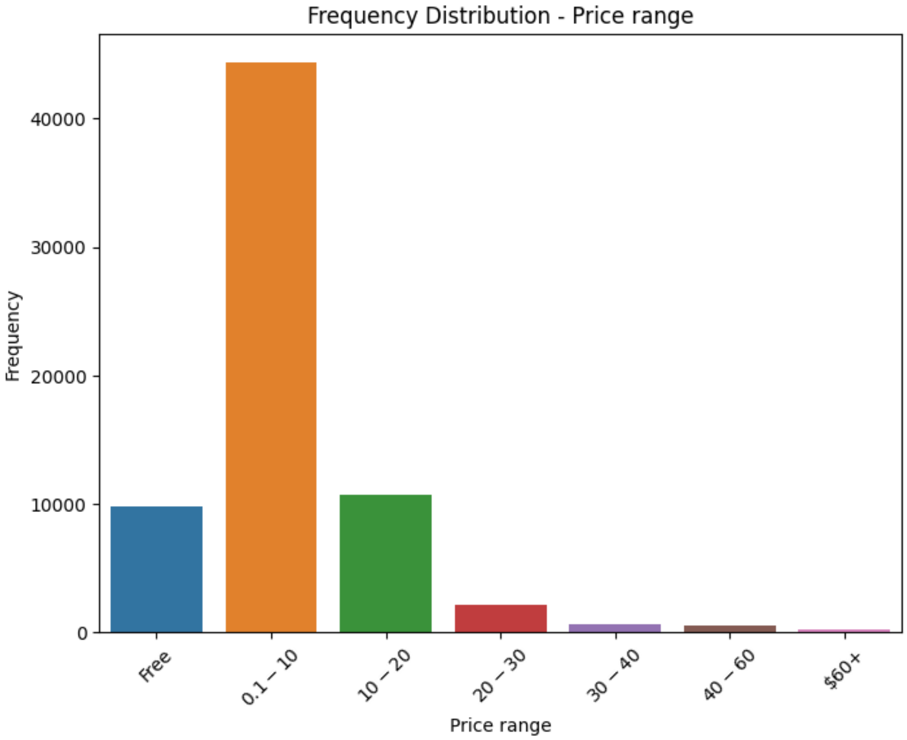


Figure [9]: Frequency distribution of required age

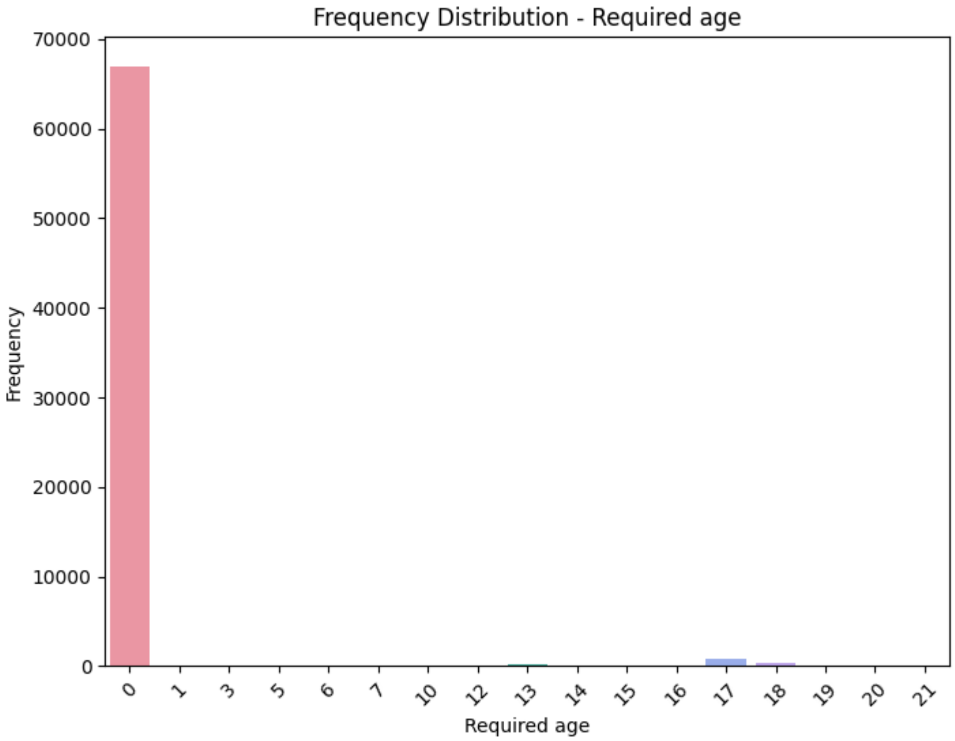


Figure [10]: Frequency distribution of number of publishers

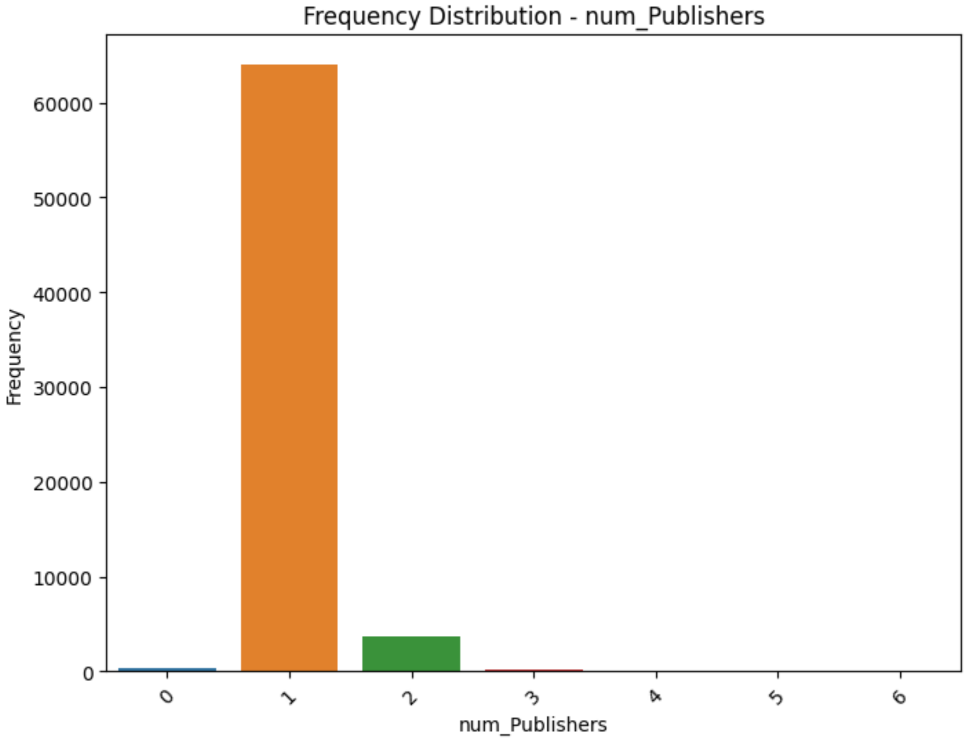


Figure [11]: Frequency distribution of number of categories

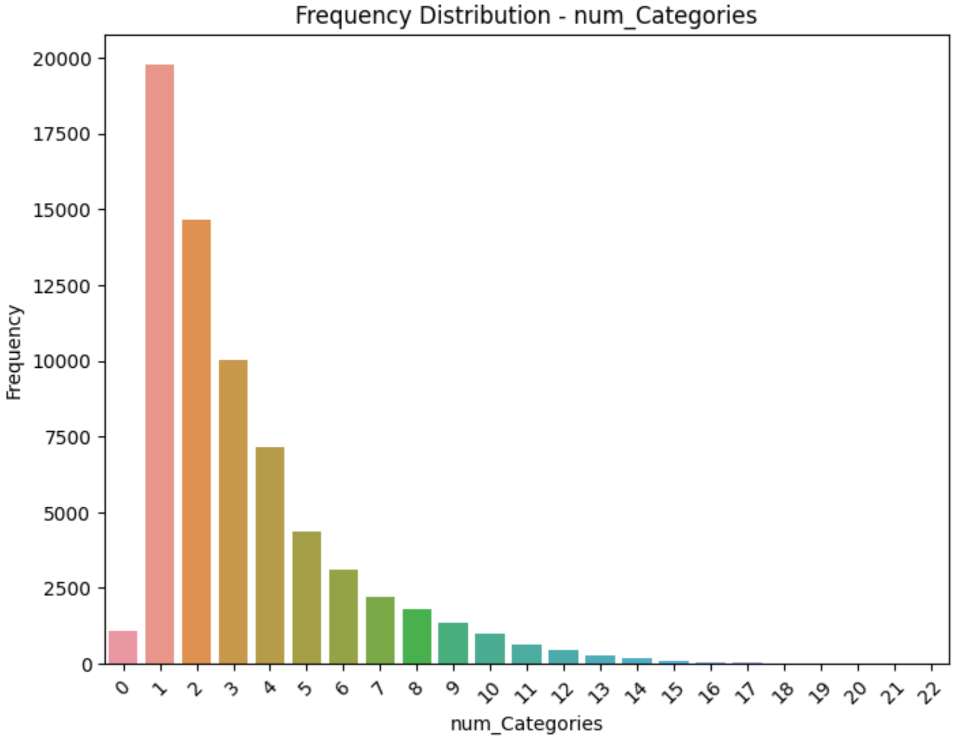


Figure [12]: Frequency distribution of games and software supporting English

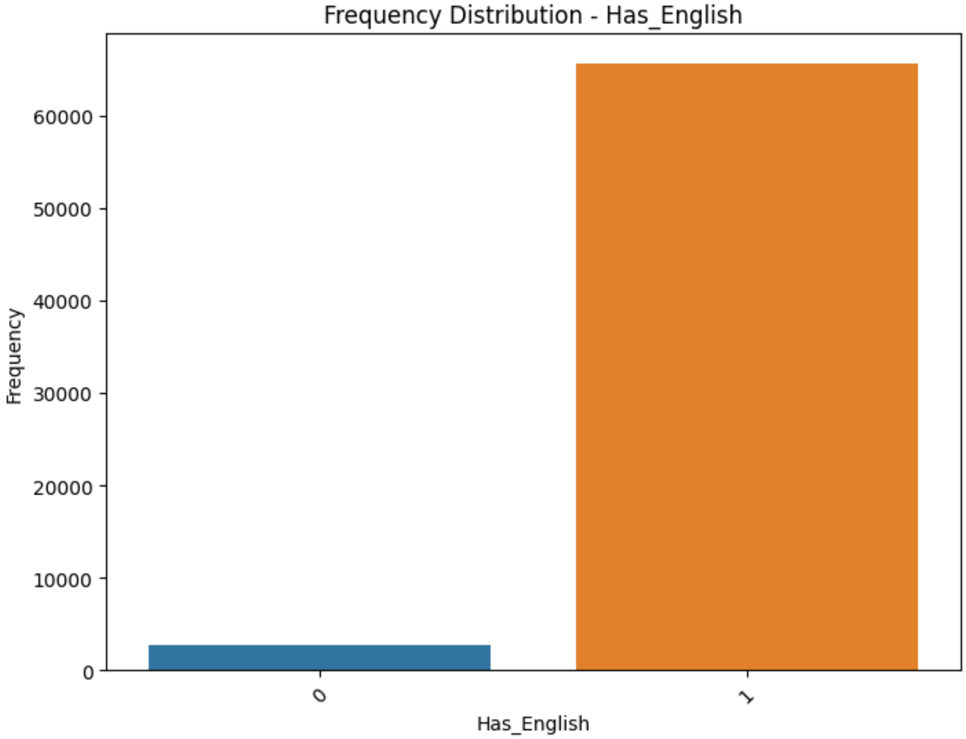


Figure [13]: Frequency distribution of number of genres

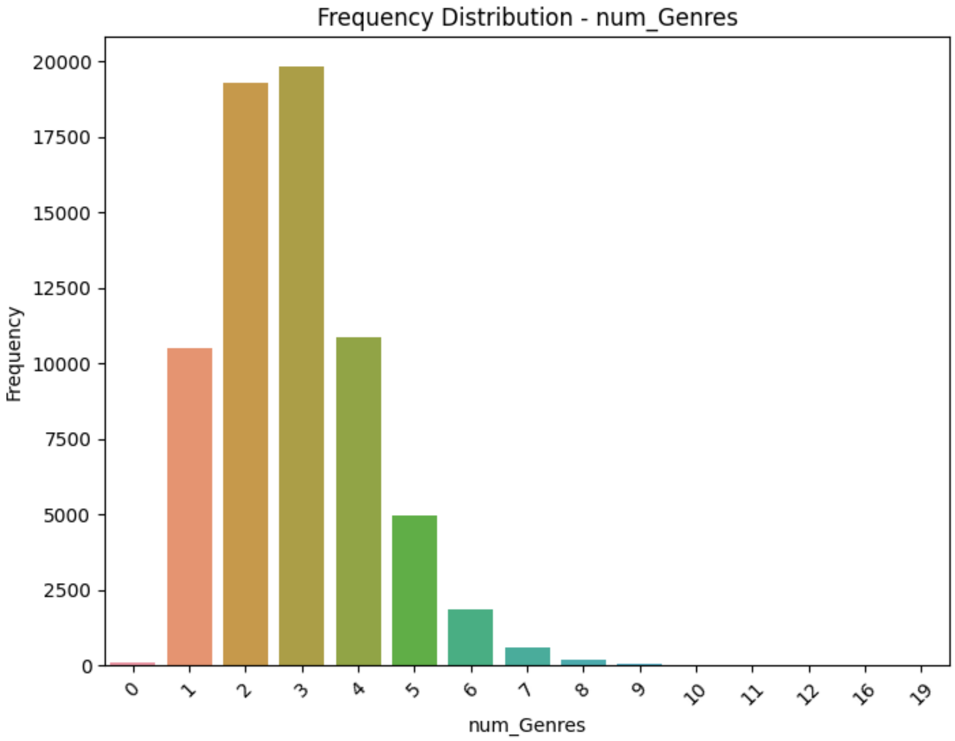


Figure [14]: Frequency distribution of number of tags

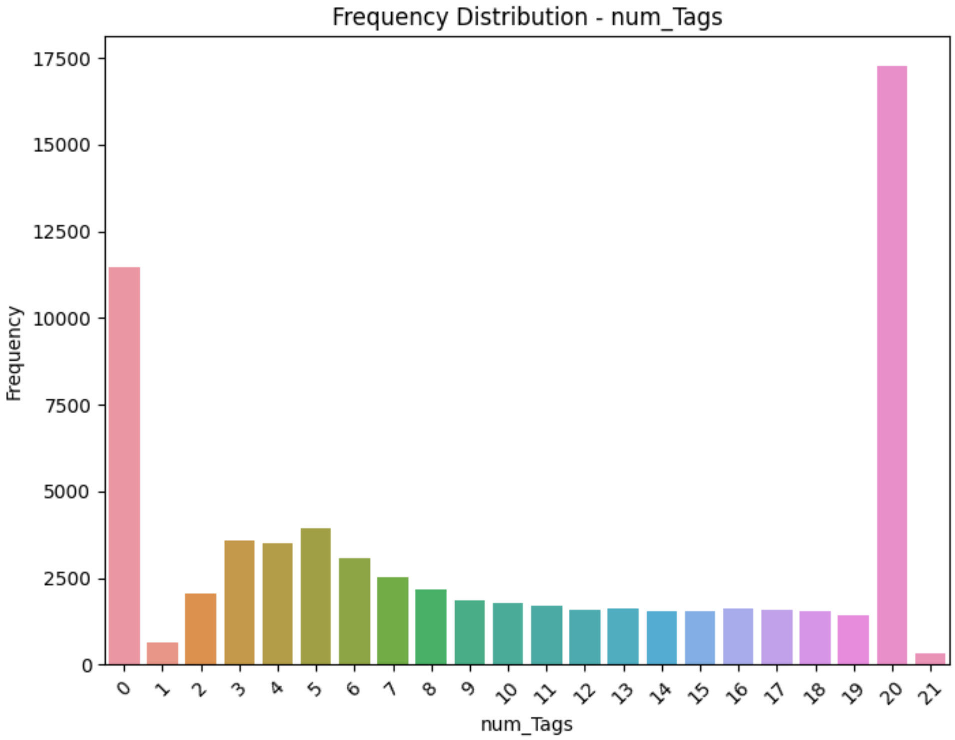


Figure [15]: Frequency distribution of number of supported languages

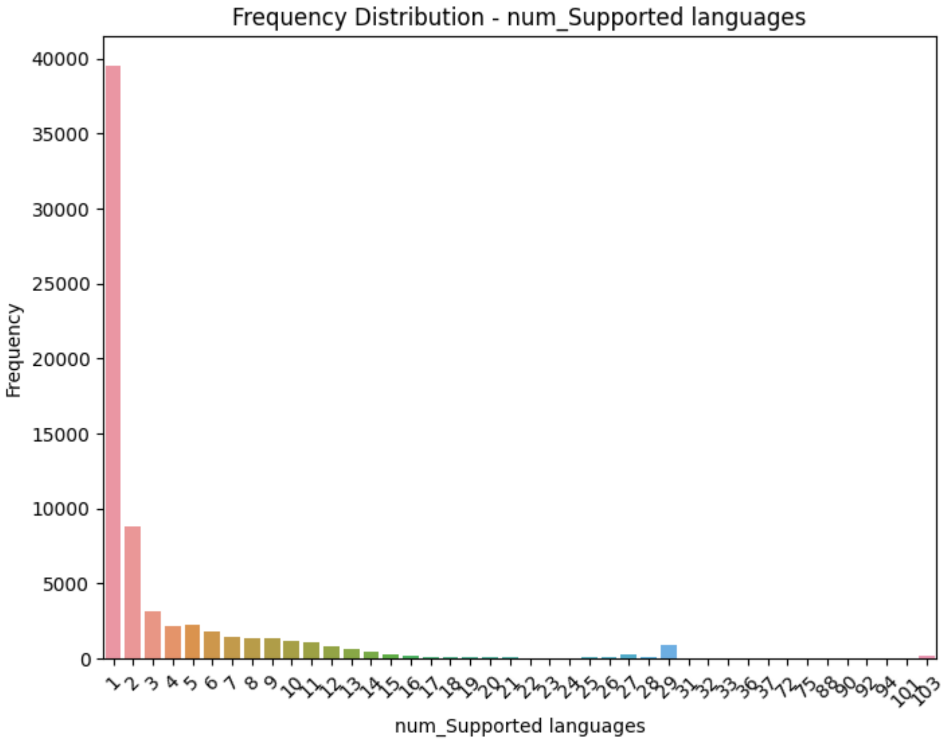


Figure [16]: Frequency distribution of number of supported audio languages

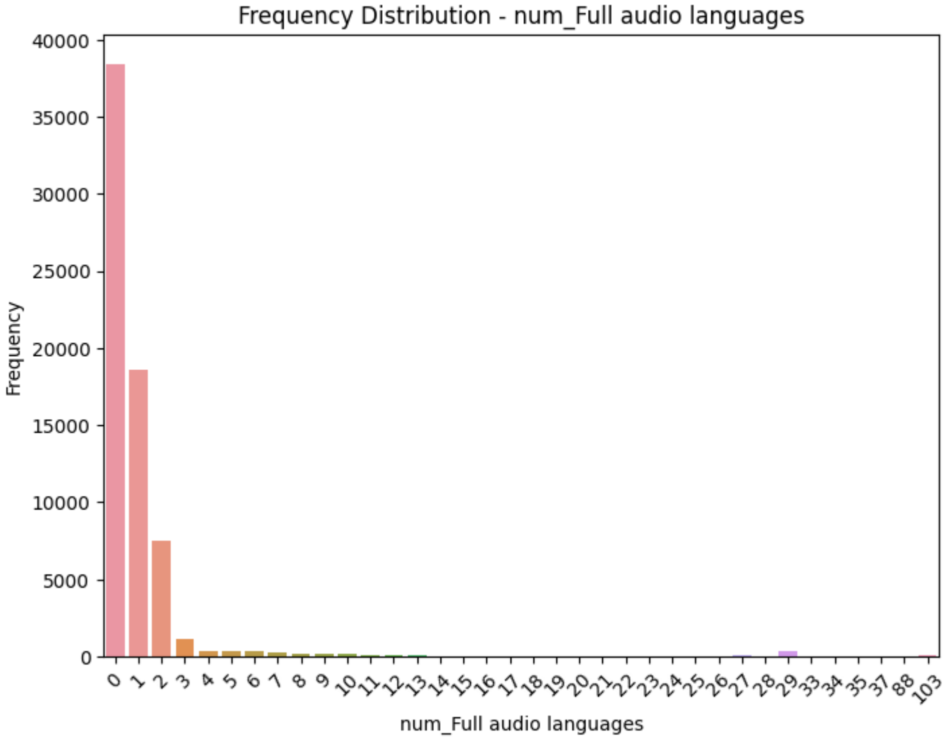


Figure [17]: Frequency distribution of number of platforms

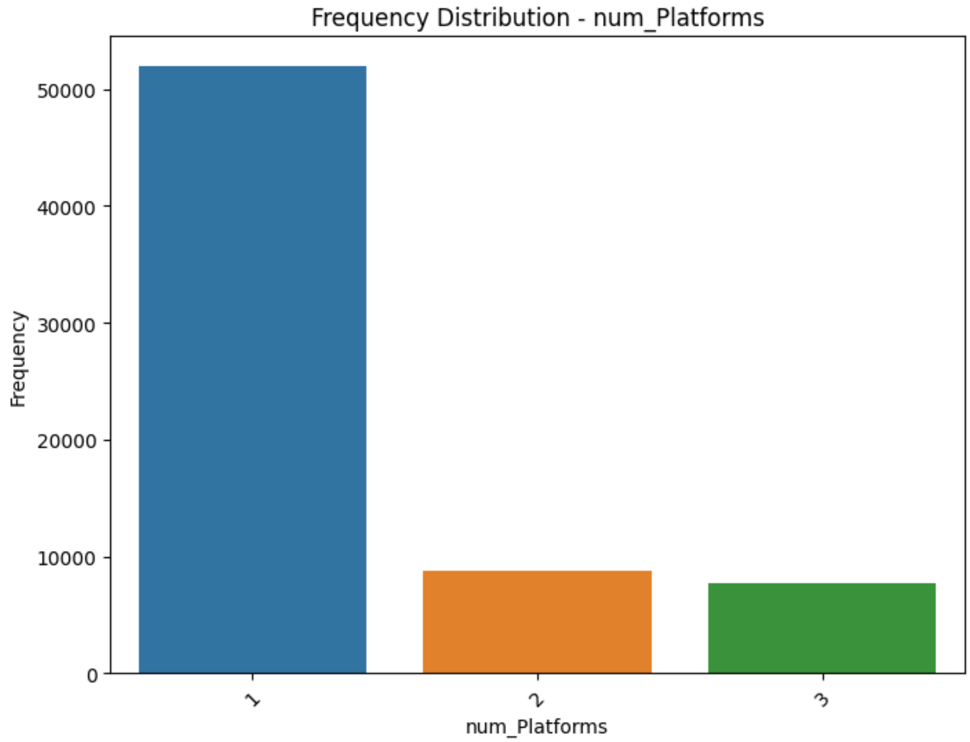


Figure [18]: Frequency distribution of games with content warning indicator

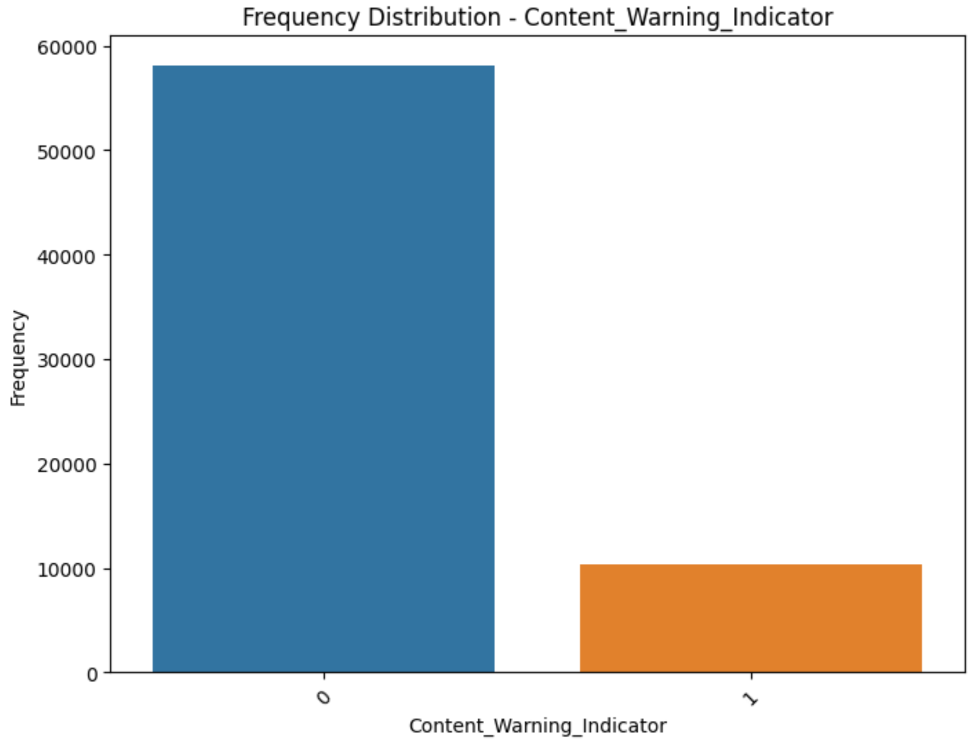


Figure [19]: Frequency distribution of release by quarter

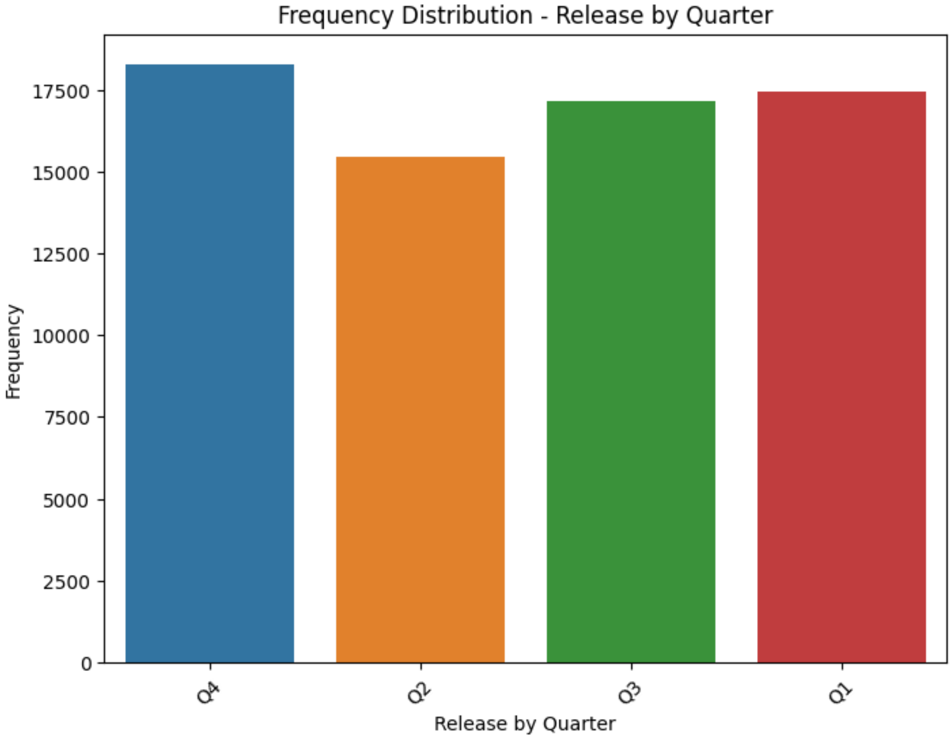


Figure [20]: Frequency distribution of release by year and quarter

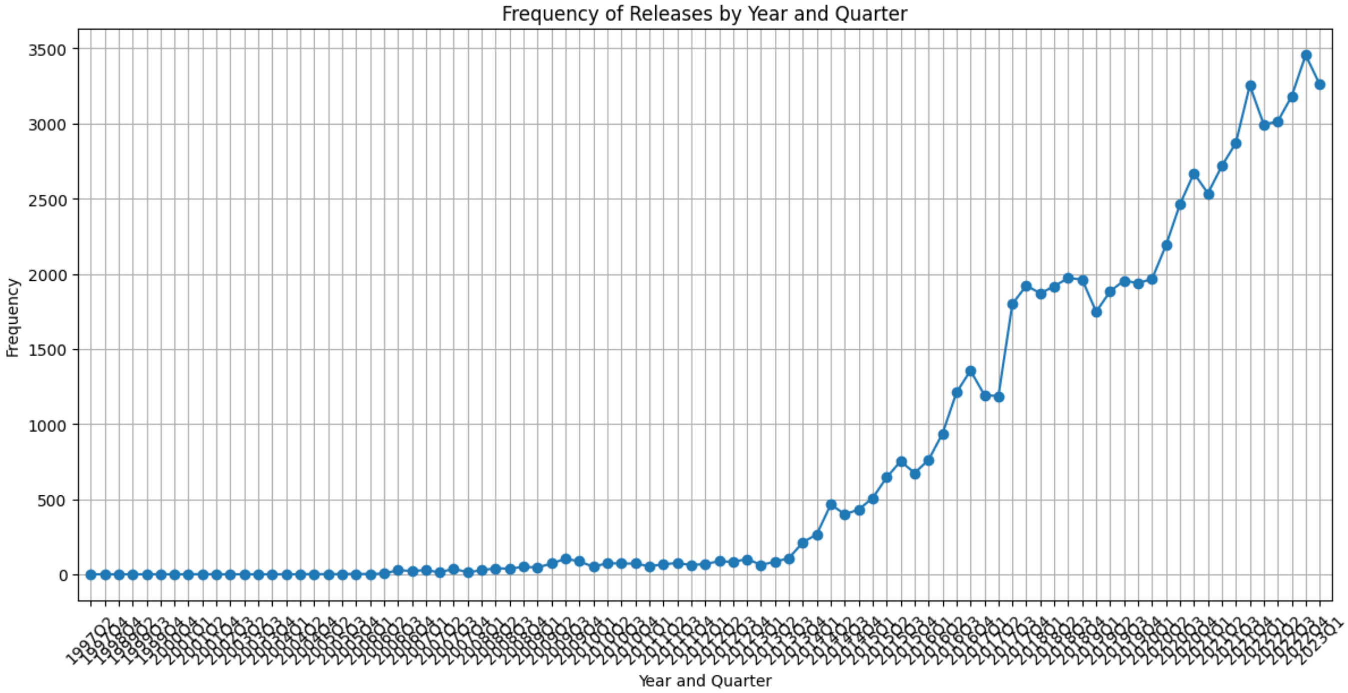


Figure [21]: Top 17 most frequent categories

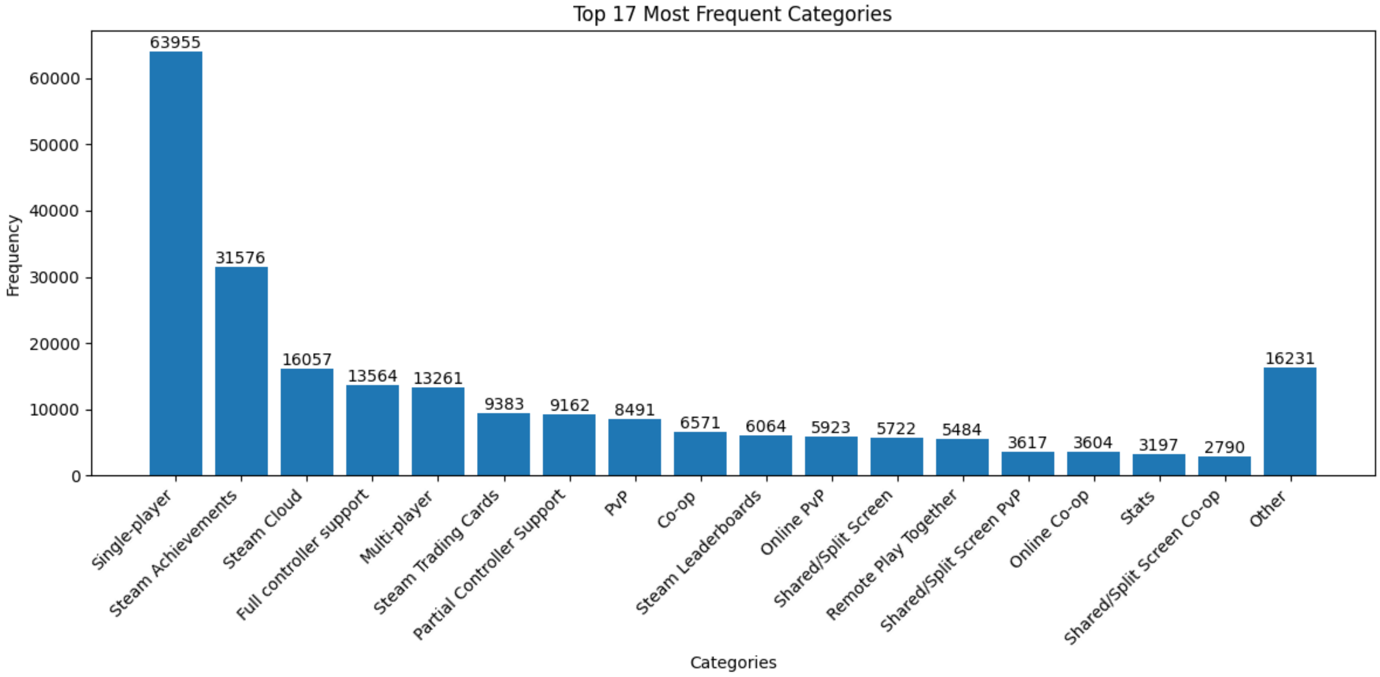


Figure [22]: Top 25 most frequent tags

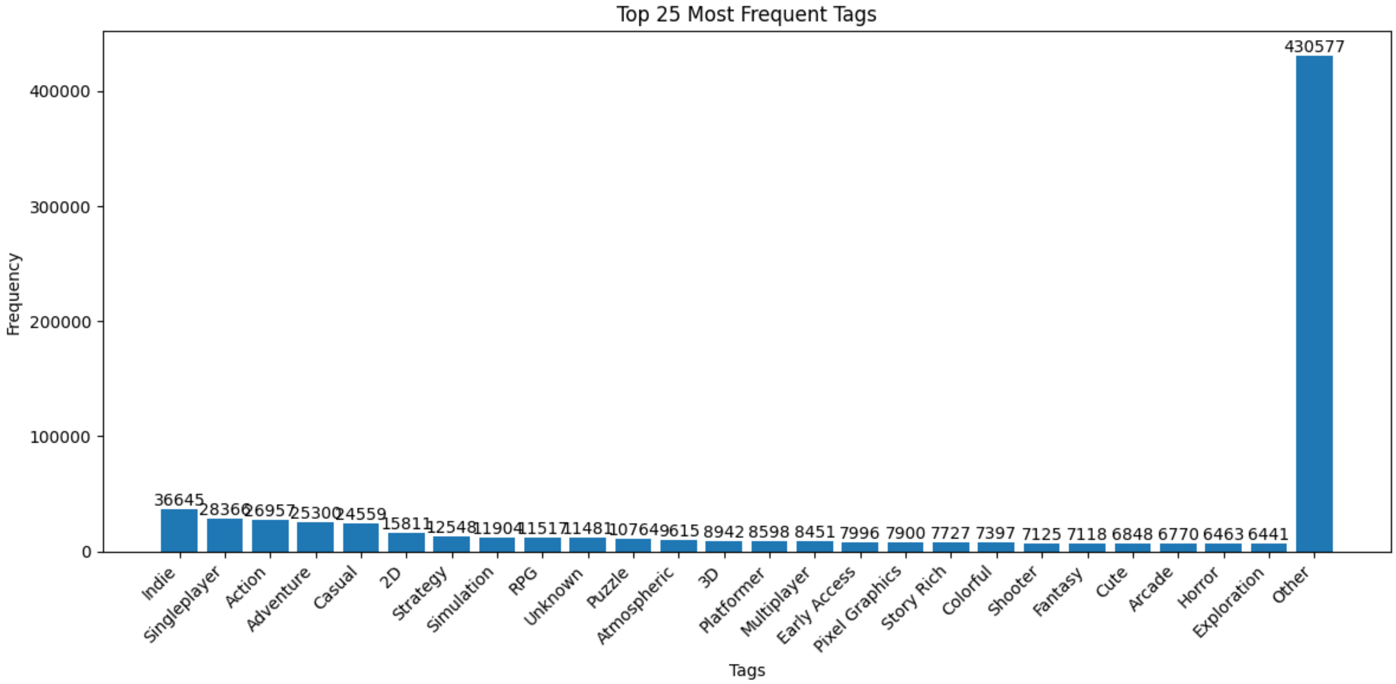


Figure [23]: Top 10 most frequent genres

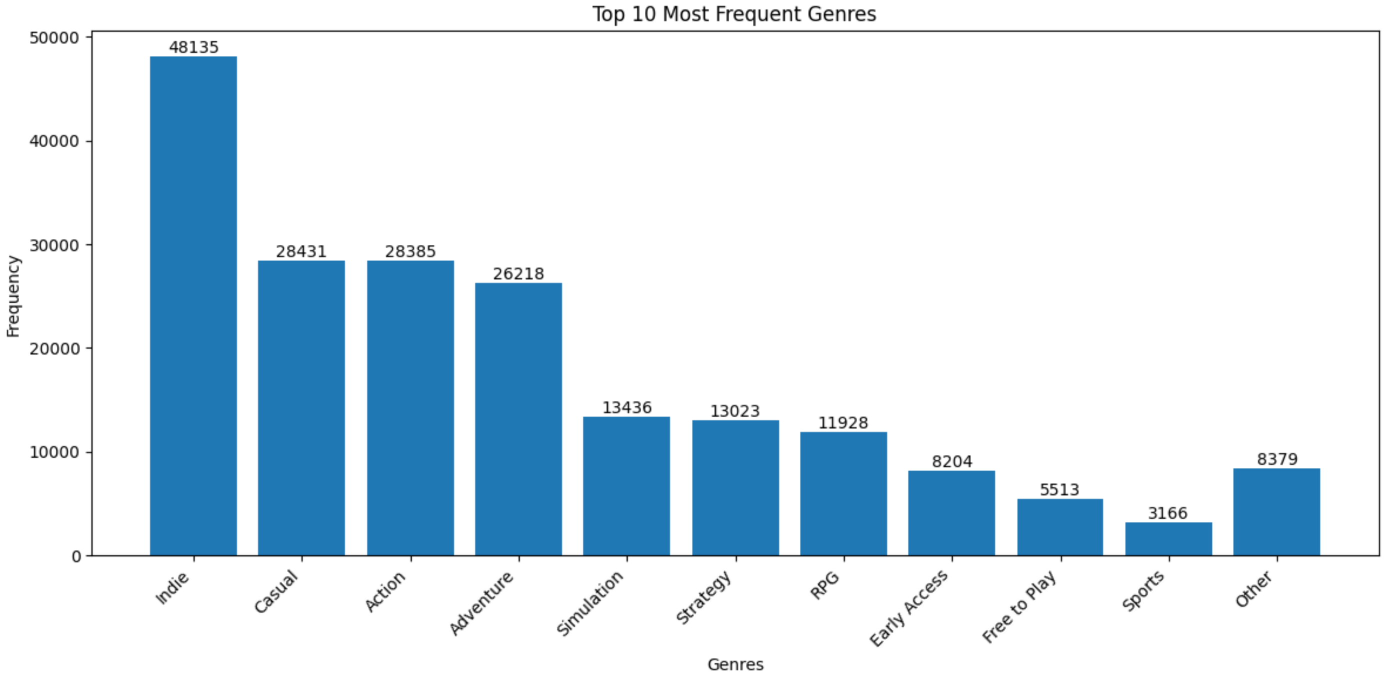


Figure [24]: Outlier detection of required age

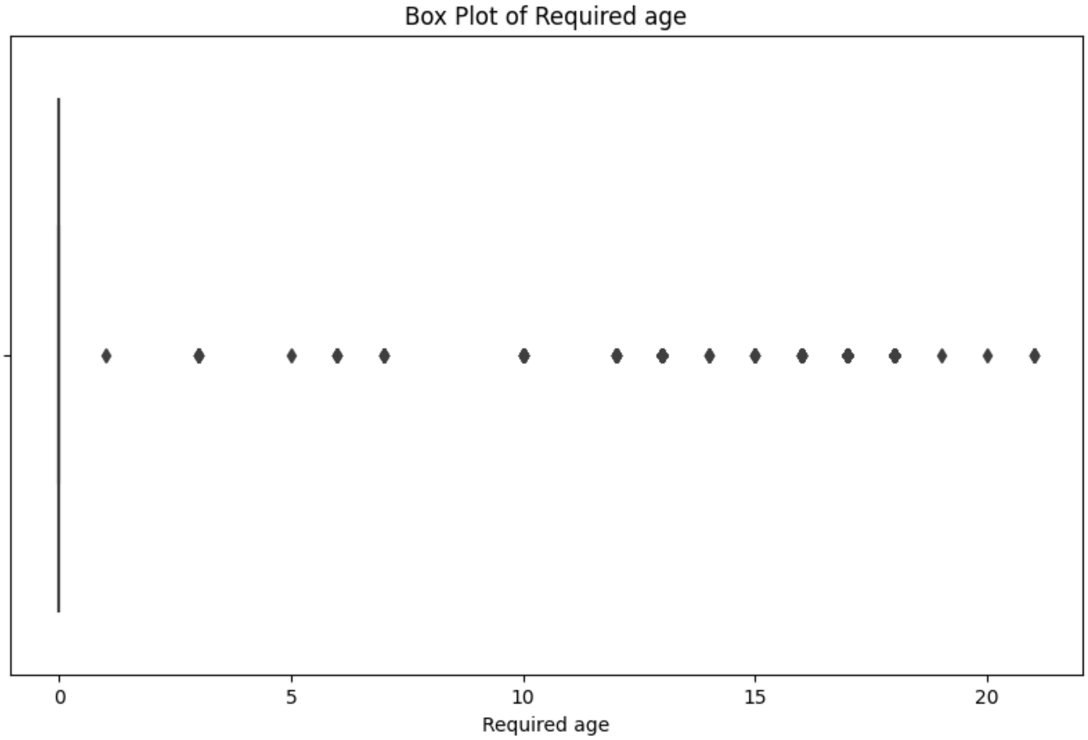


Figure [25]: Outlier detection of price

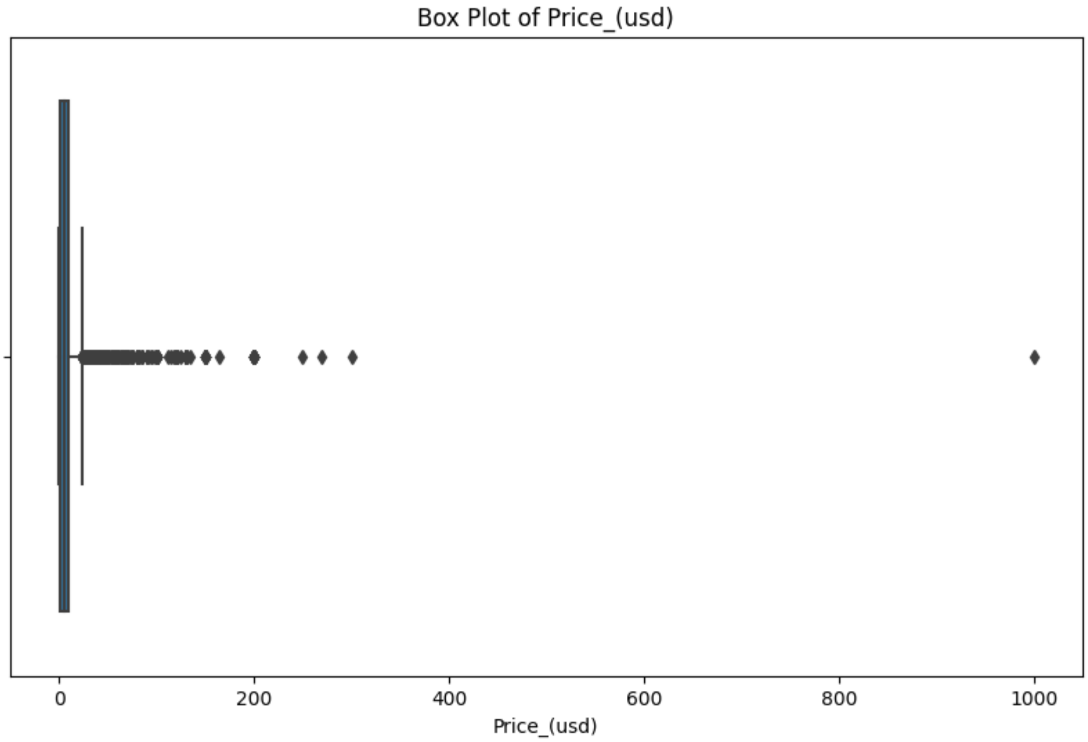


Figure [26]: Outlier detection of Metacritic score

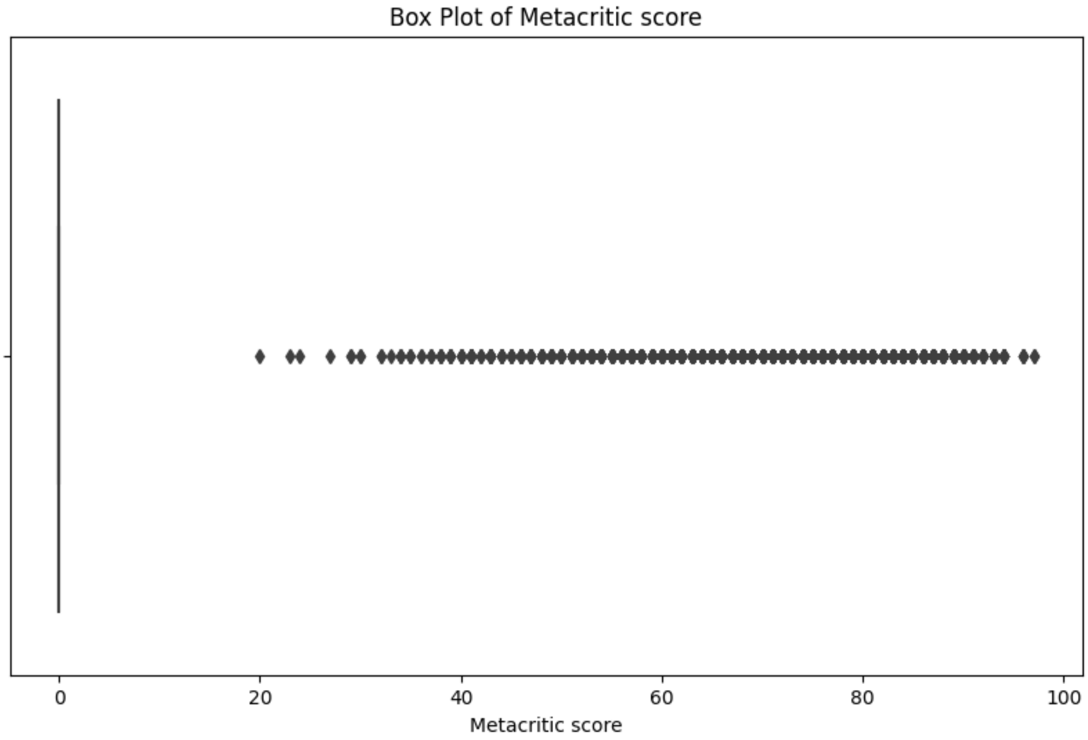


Figure [27]: Outlier detection of user score

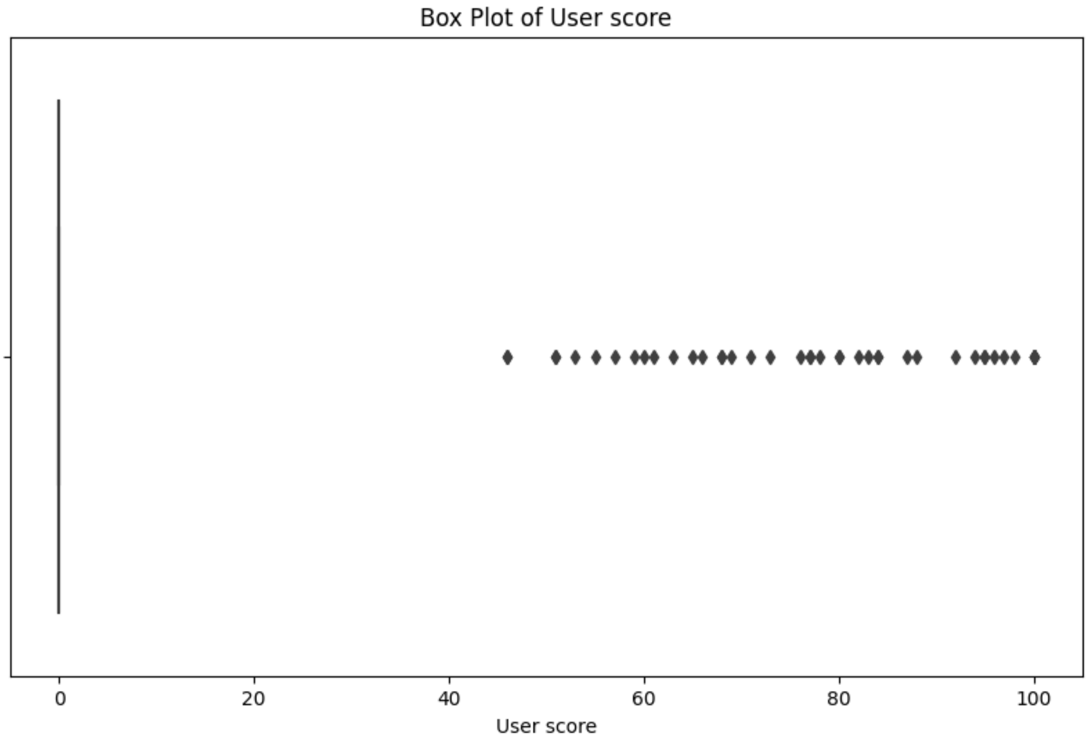


Figure [28]: Outlier detection of positive reviews

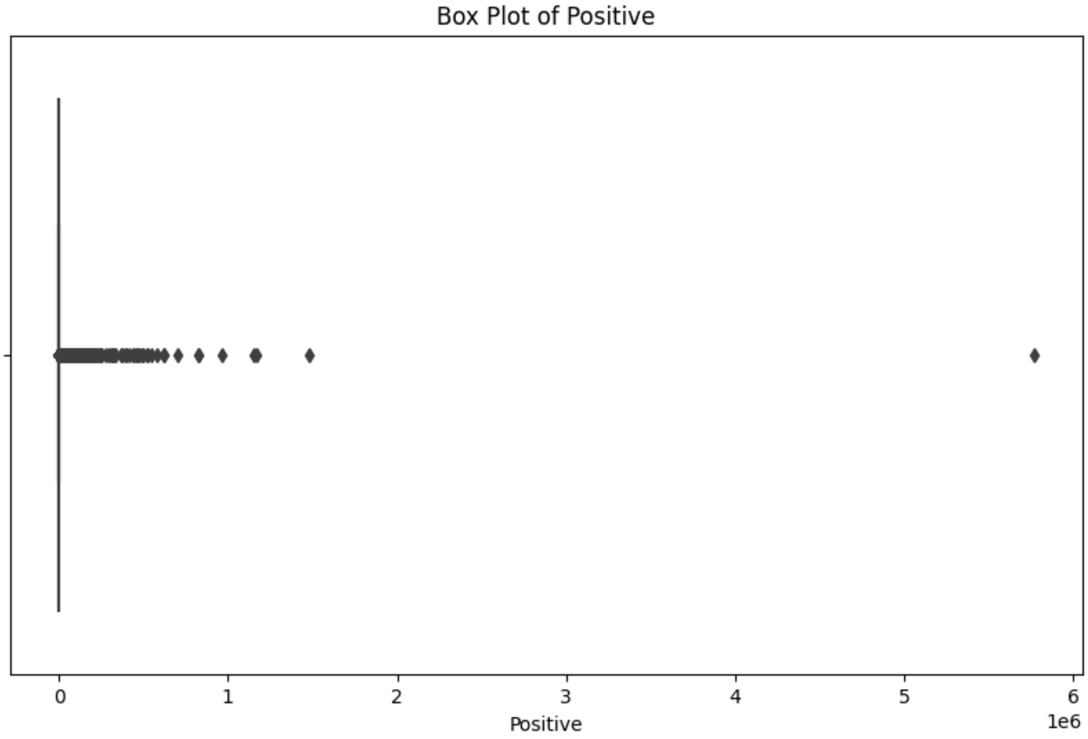


Figure [29]: Outlier detection of negative reviews

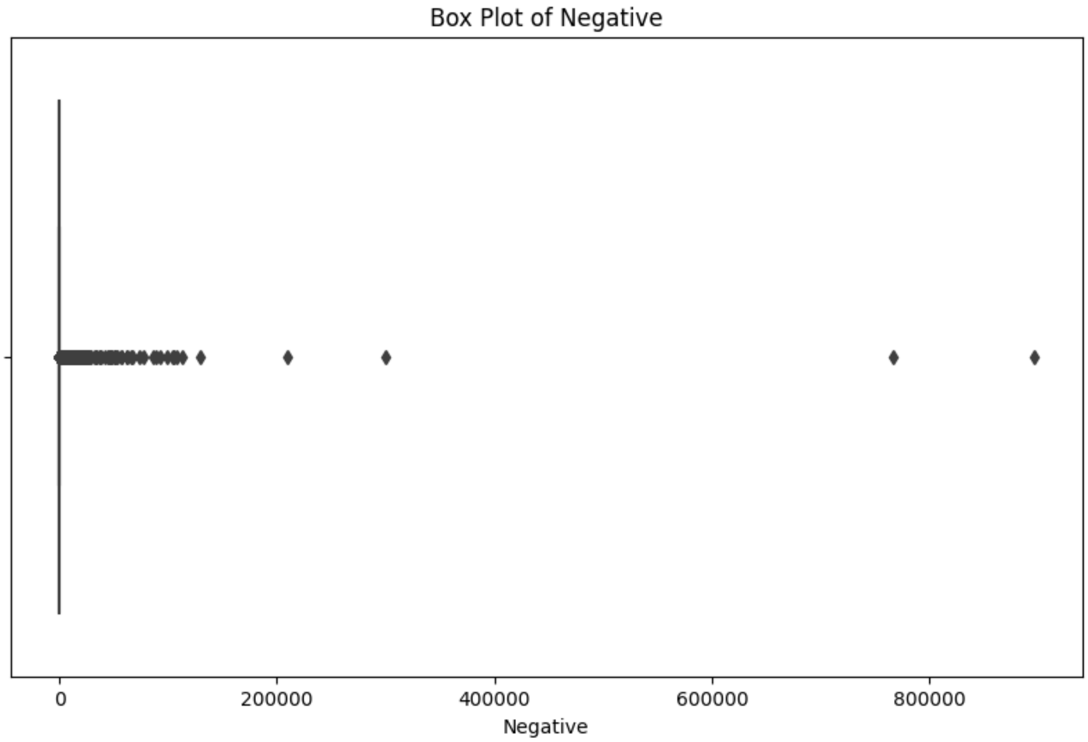


Figure [30]: Outlier detection of number of achievements

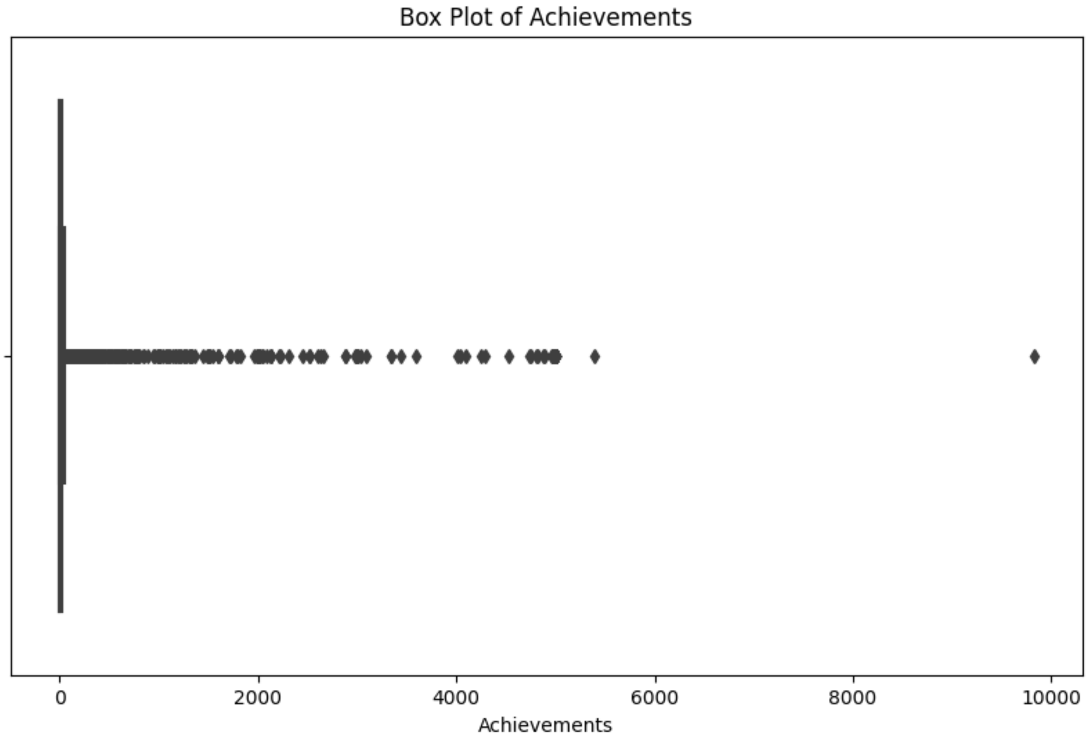


Figure [31]: Outlier detection of number of recommendations

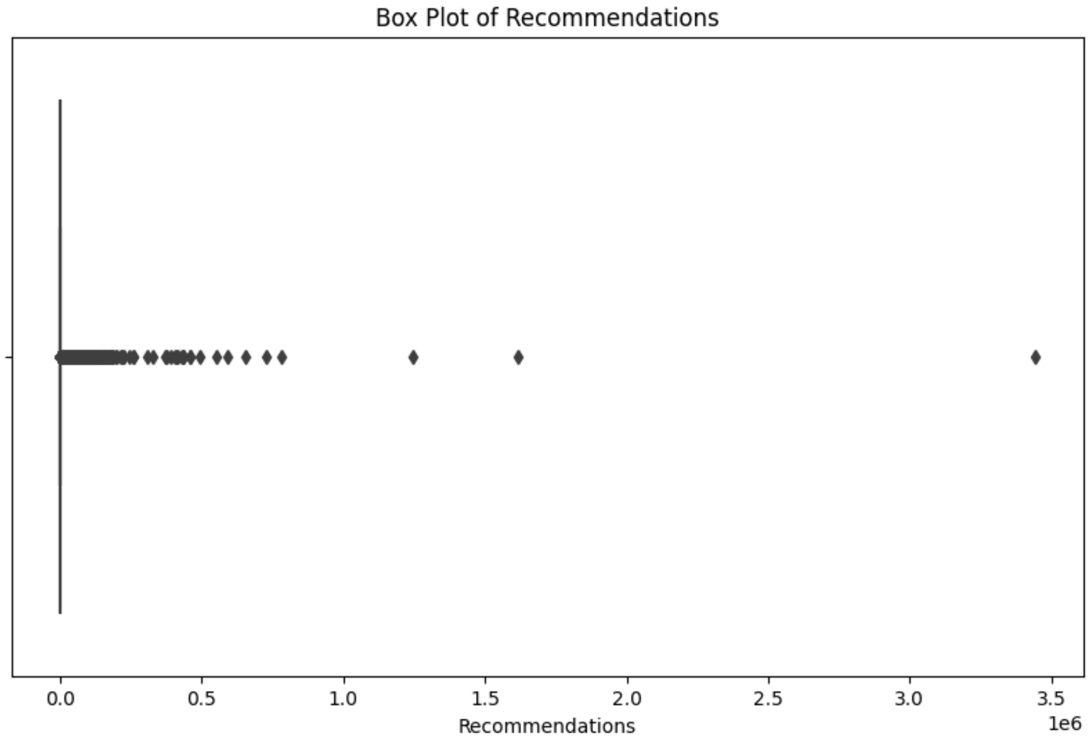


Figure [32]: Outlier detection of average playtime forever

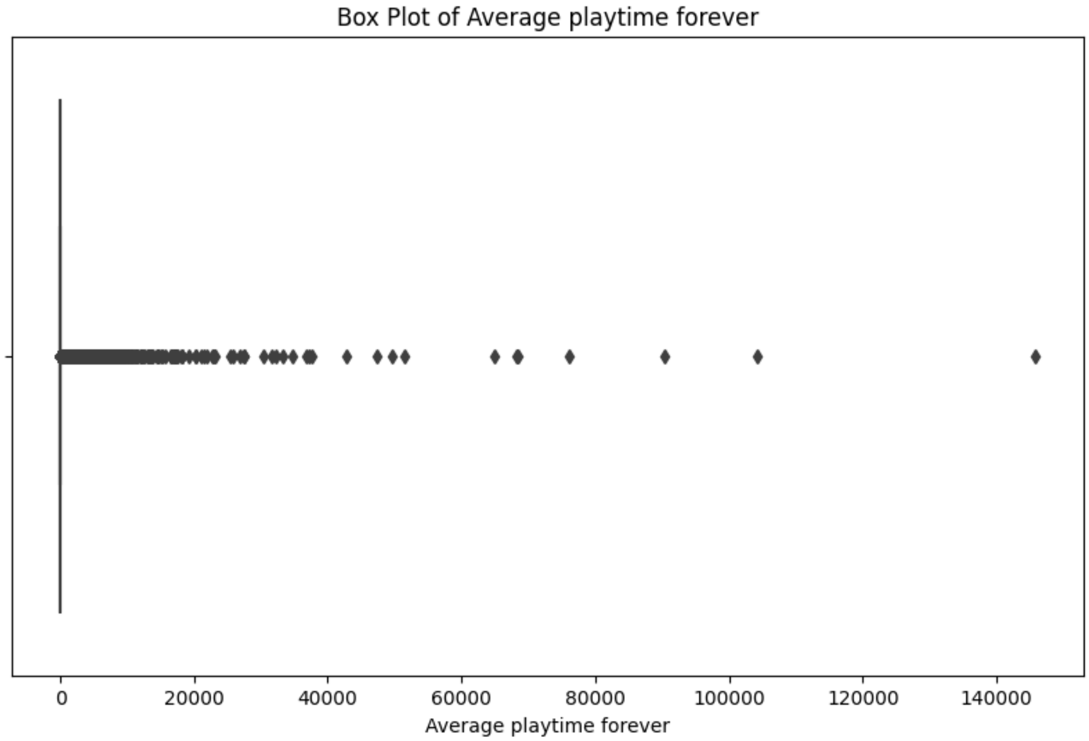


Figure [33]: Outlier detection of average playtime in two weeks

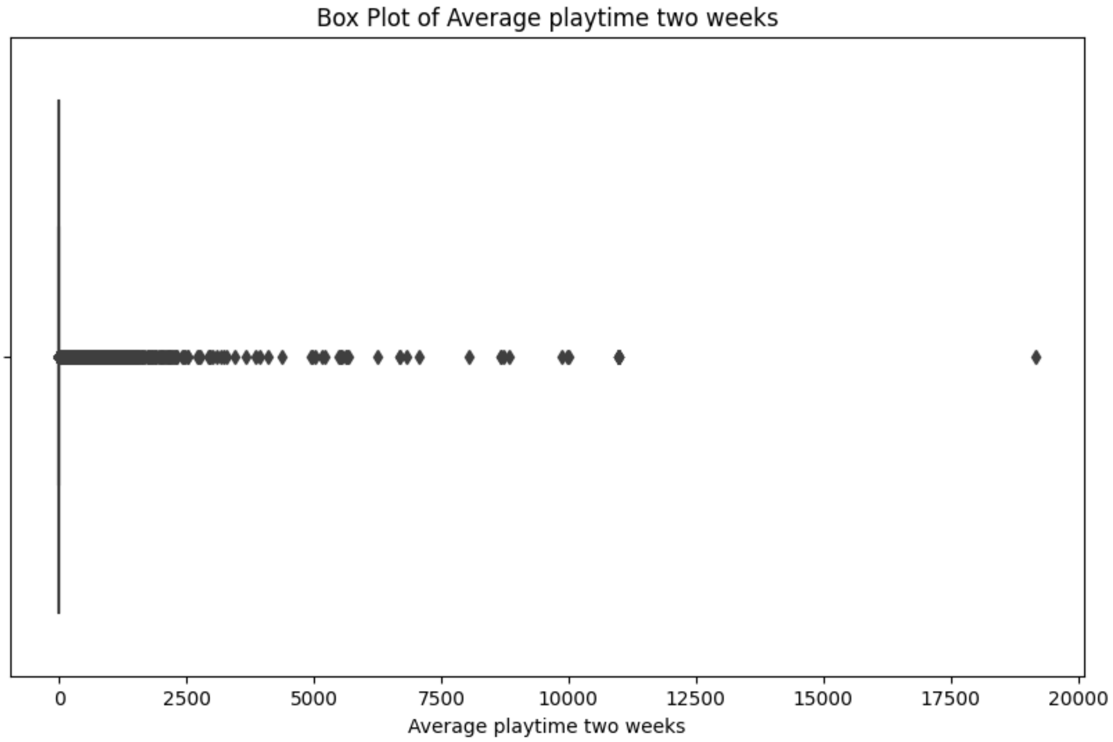


Figure [34]: Outlier detection of median playtime forever

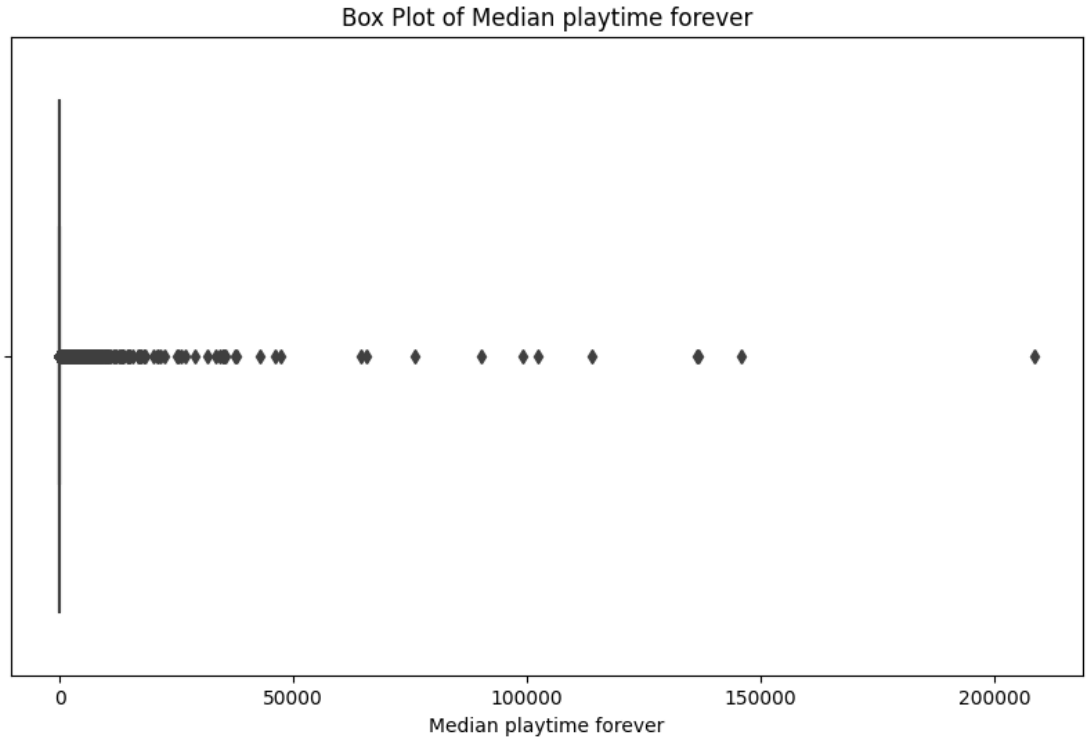


Figure [35]: Outlier detection of median playtime in two weeks

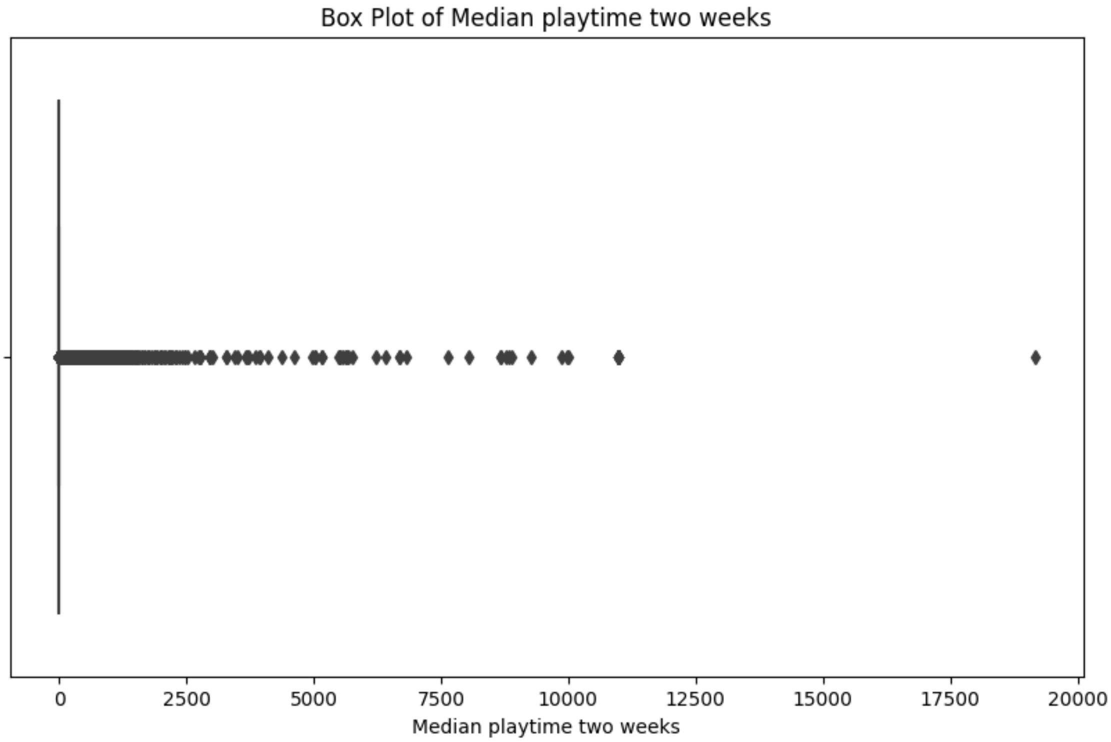


Figure [36]: Outlier detection of UNIX release date

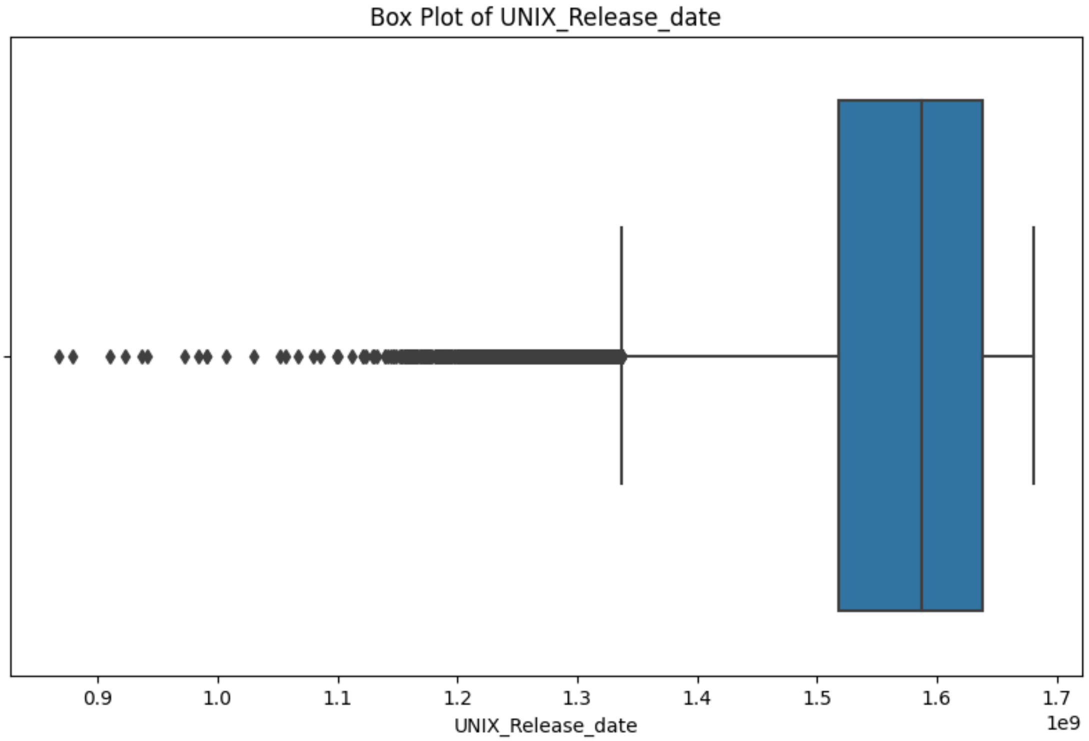


Figure [37]: Outlier detection of Max peak CCU

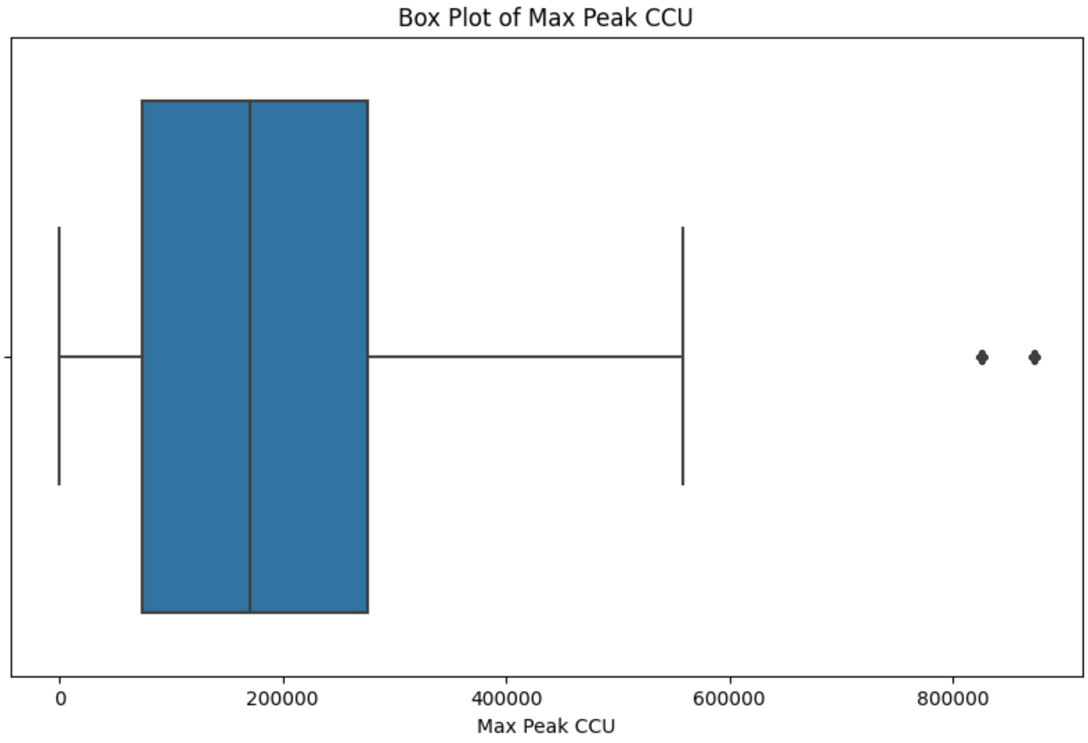


Figure [38]: Outlier detection of number of publishers

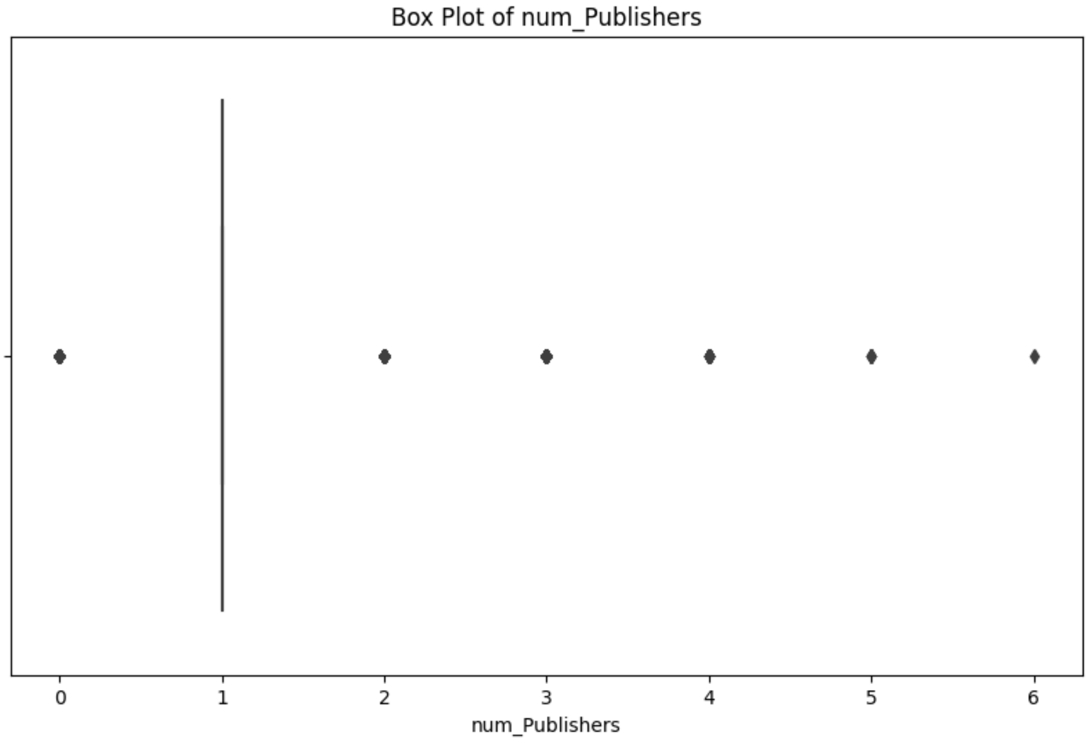


Figure [39]: Outlier detection of number of categories

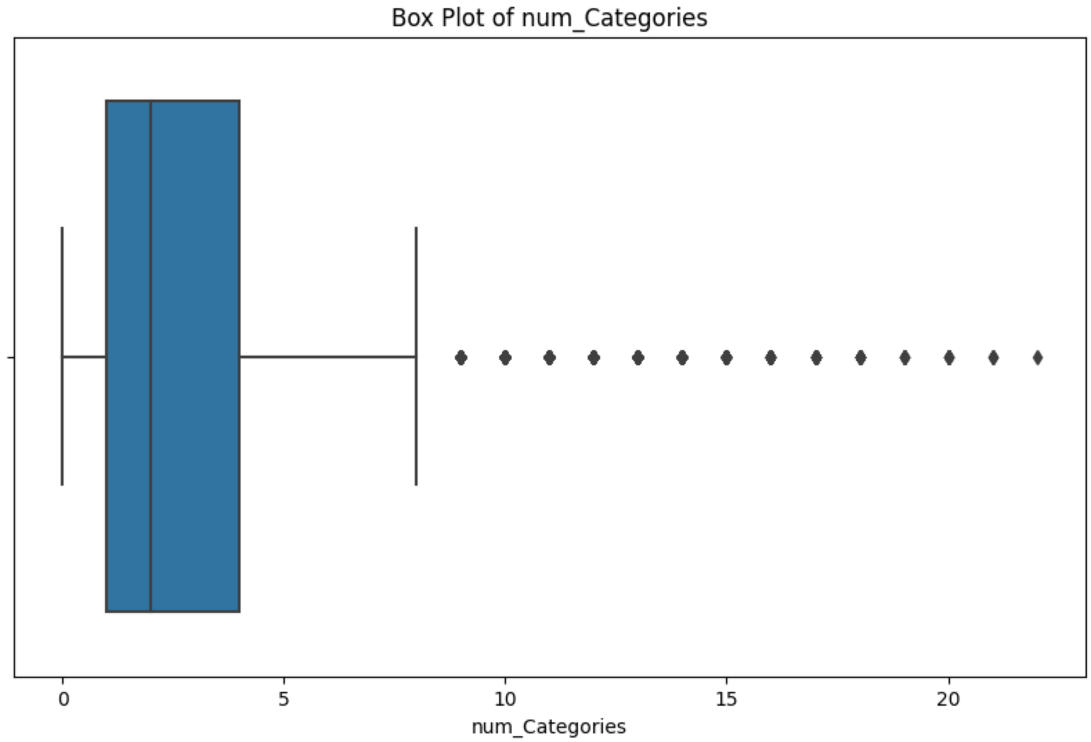


Figure [40]: Outlier detection of number of genres

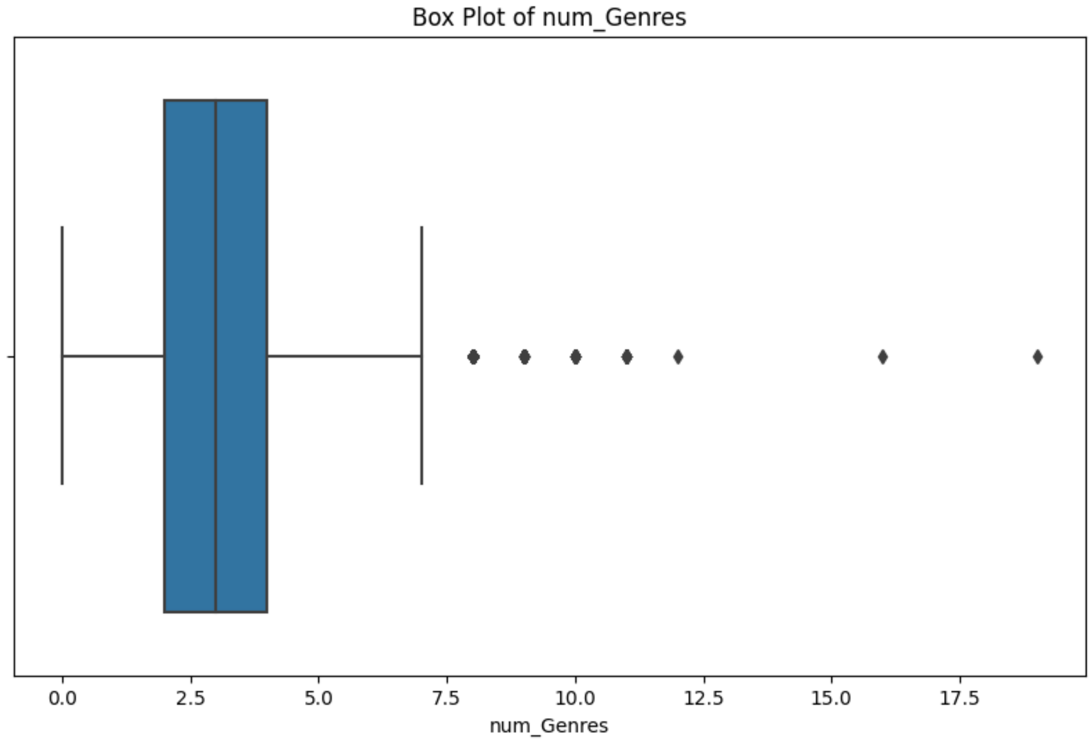


Figure [41]: Outlier detection of number of tags

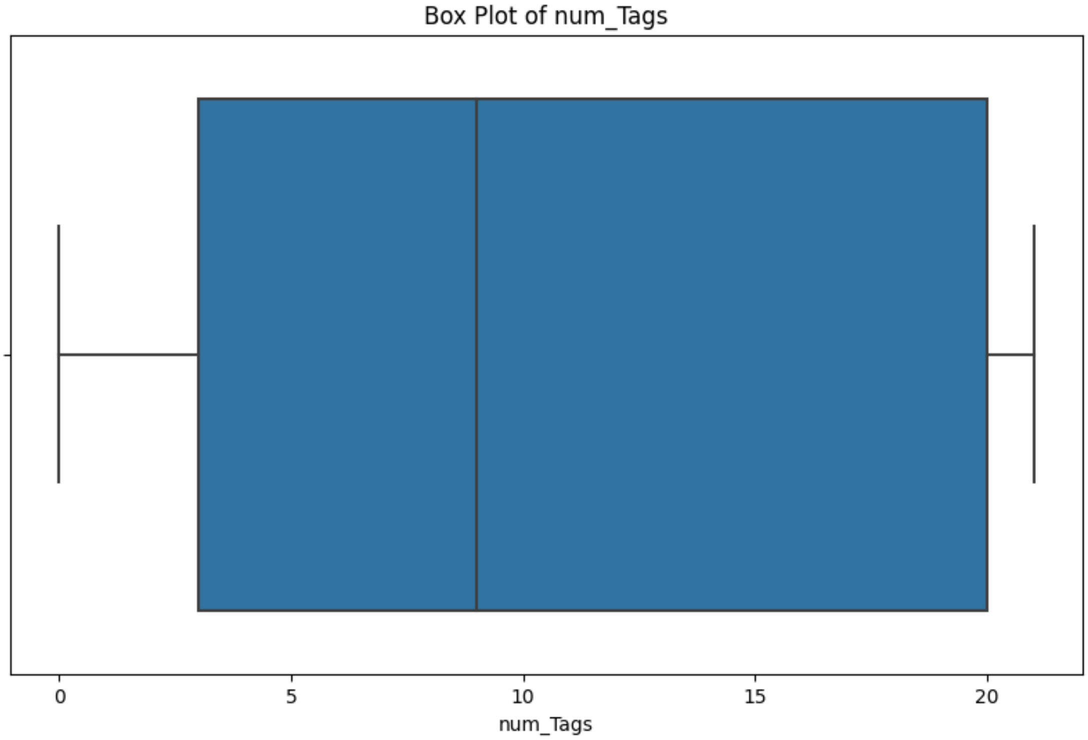


Figure [42]: Outlier detection of number of supported languages

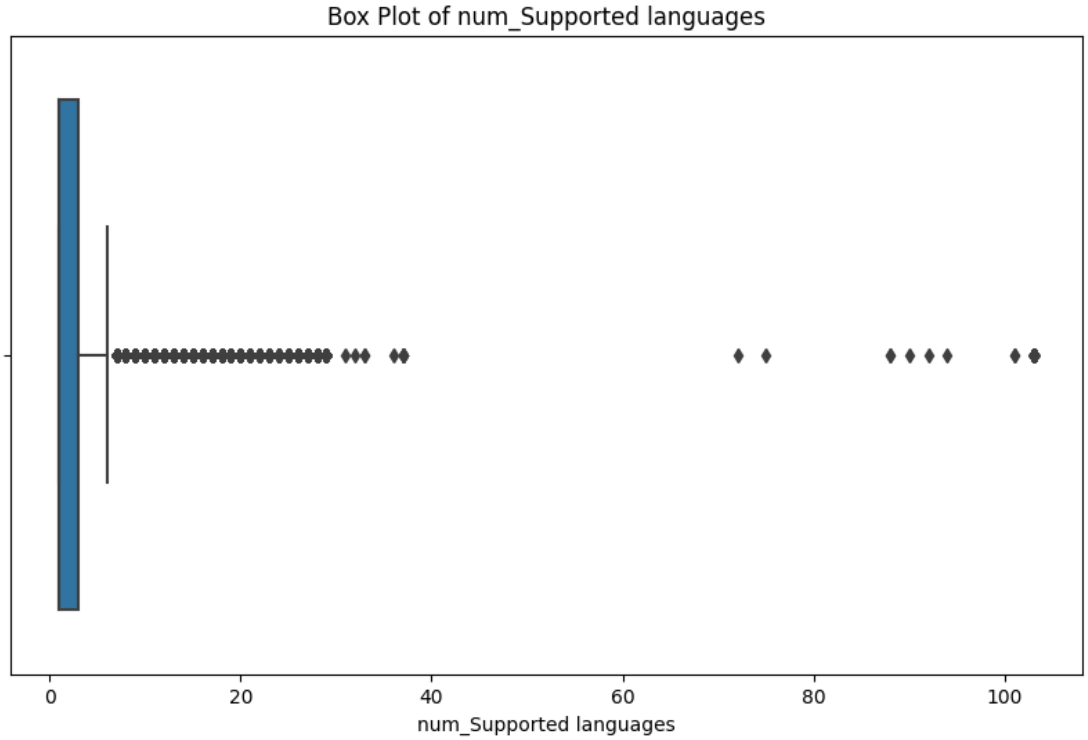


Figure [43]: Outlier detection of number of supported audio languages

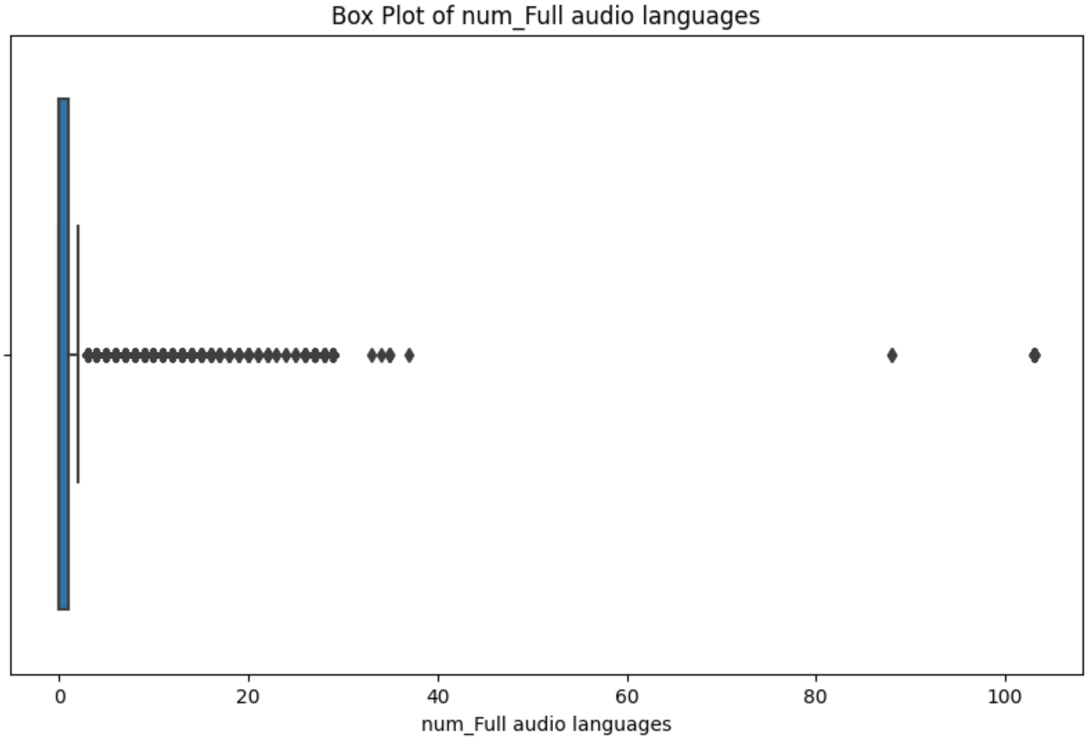


Figure [44]: Outlier detection of number of platforms

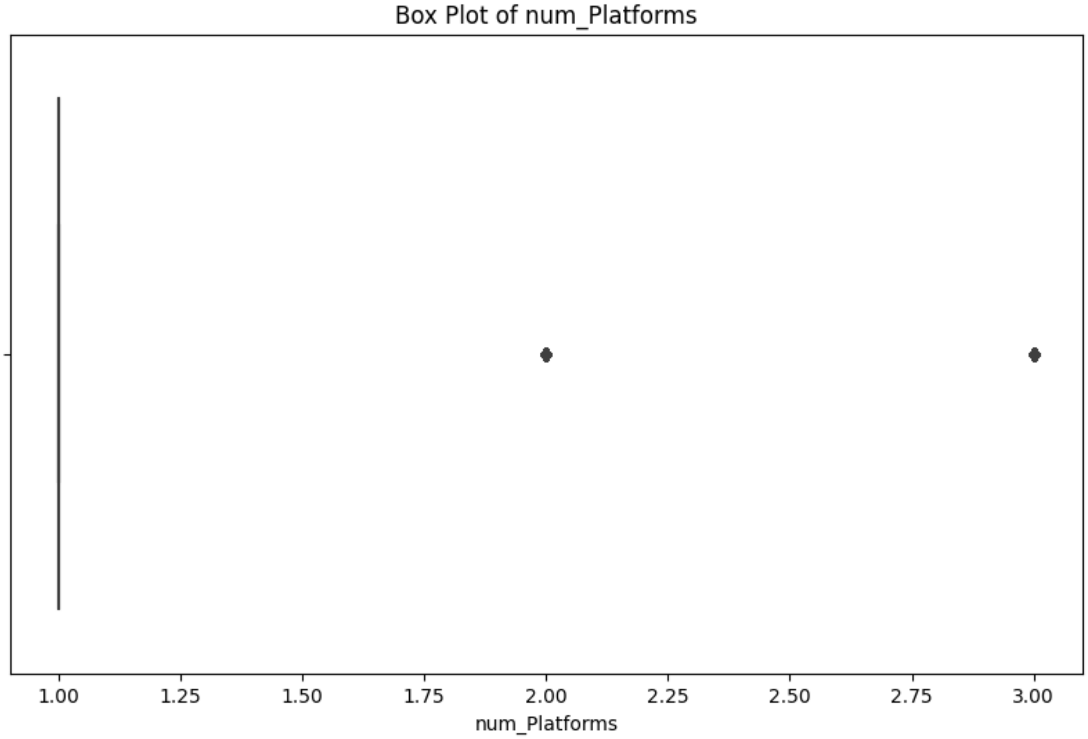


Figure [45]: Outlier detection of review ratio

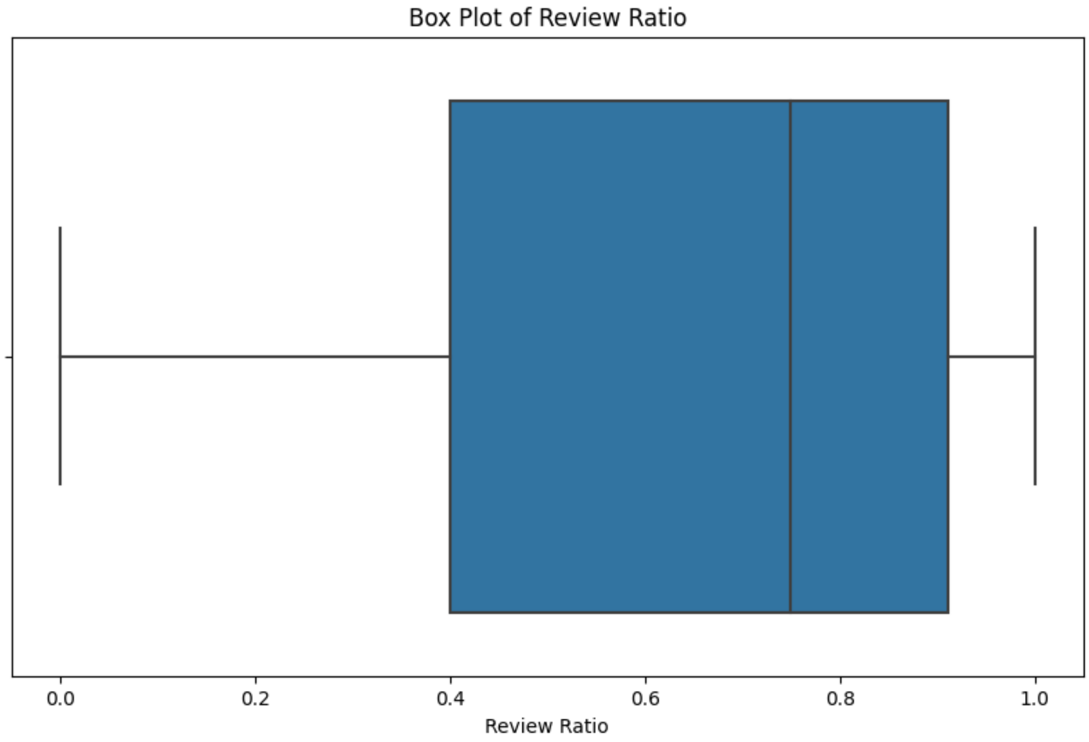


Figure [46]: Outlier detection of price per average playtime forever

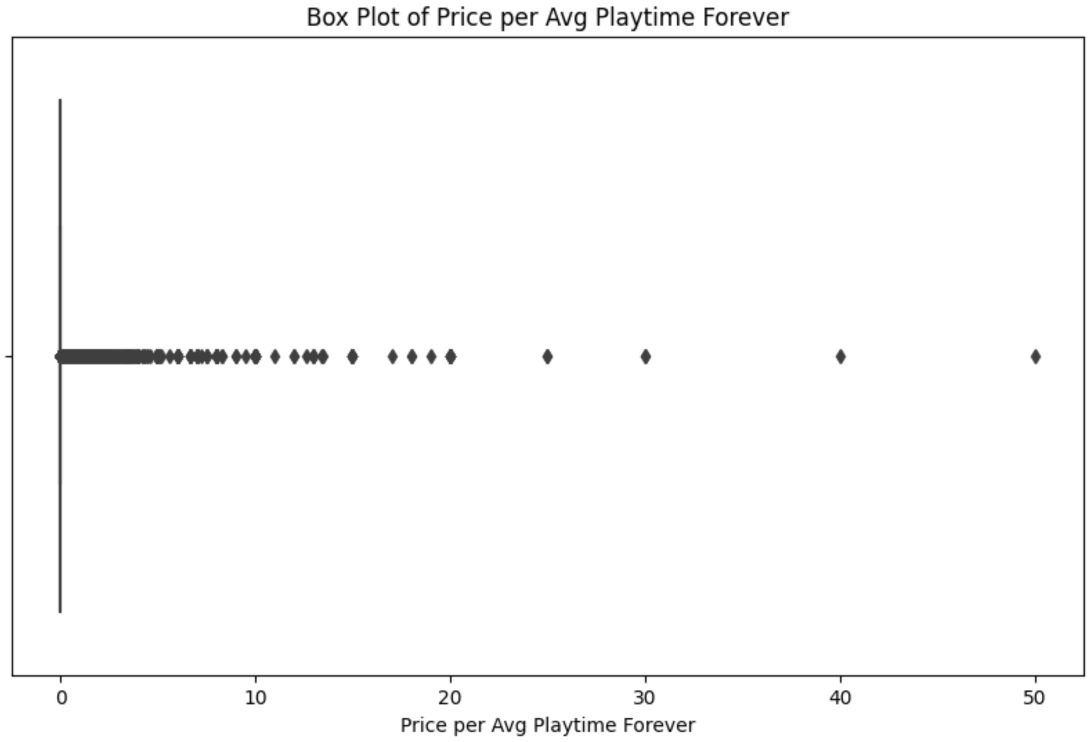


Figure [47]: Outlier detection of price per average playtime in two weeks

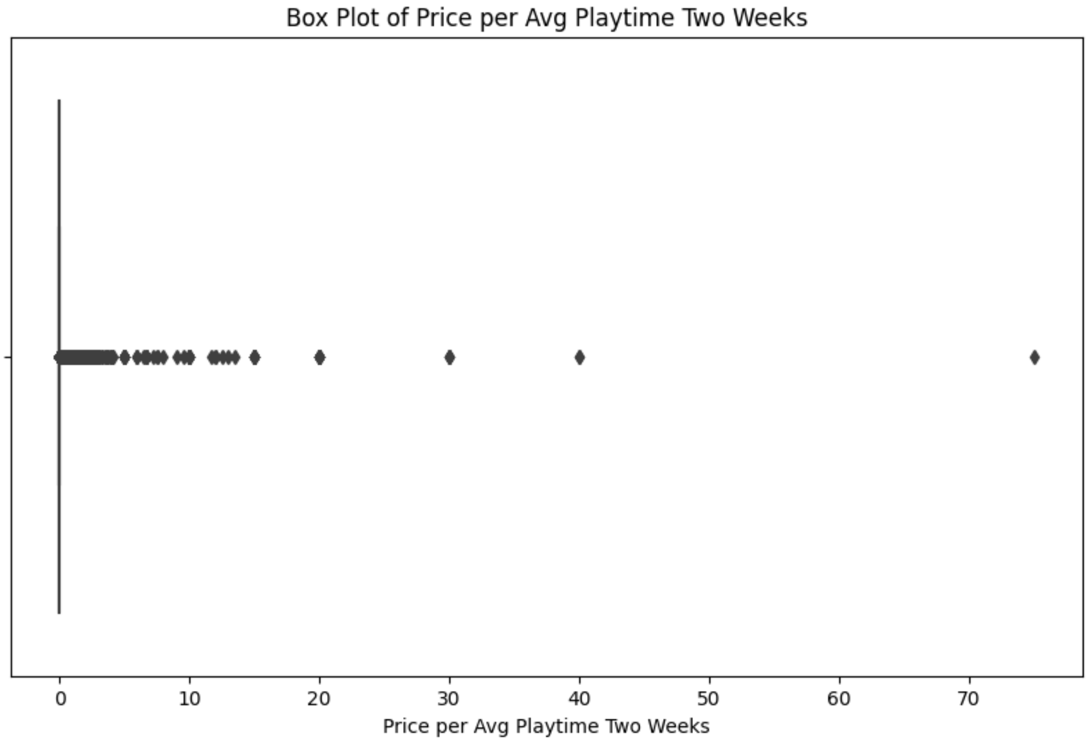


Figure [48]: Outlier detection of price per median playtime forever

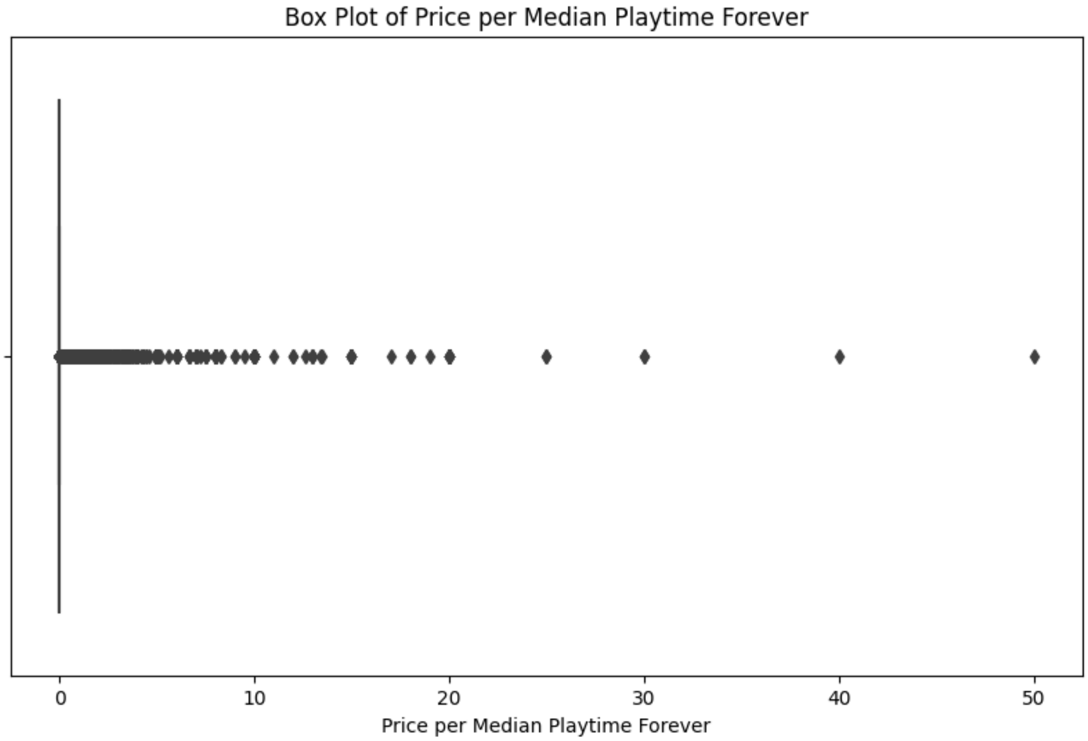


Figure [49]: Outlier detection of price per median playtime in two weeks

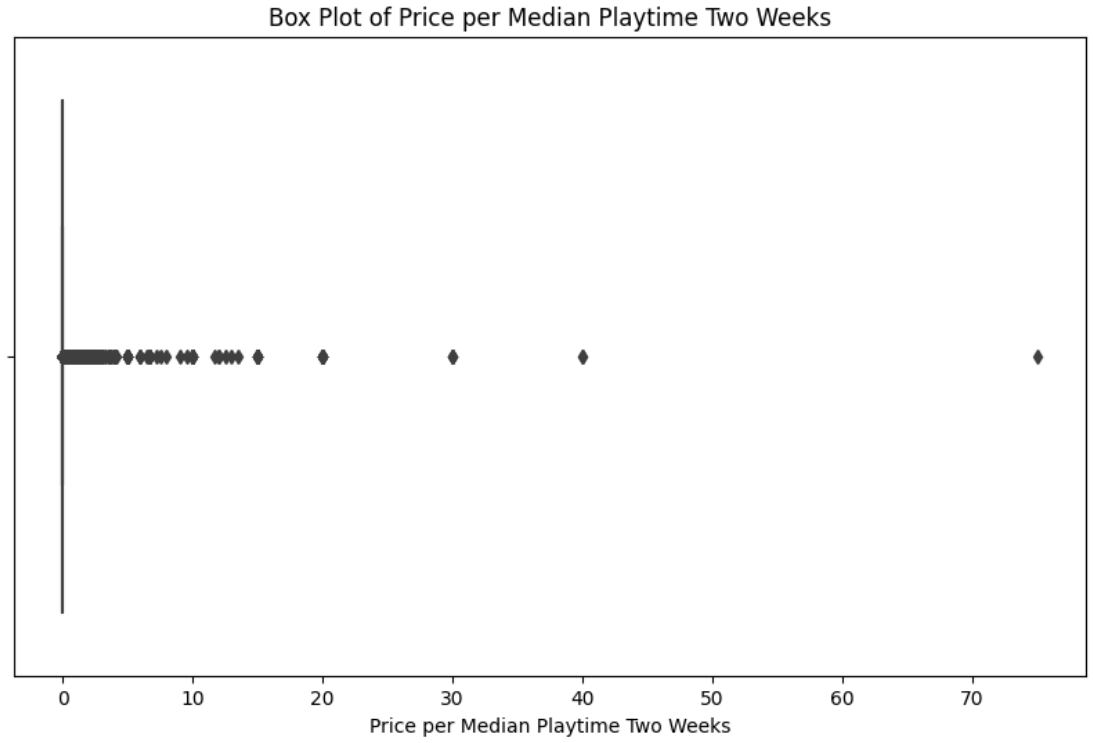
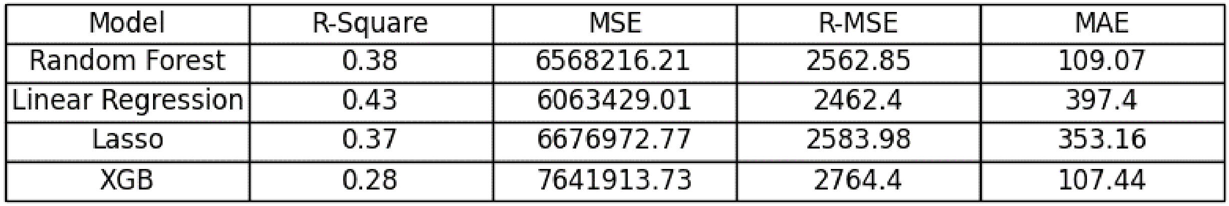
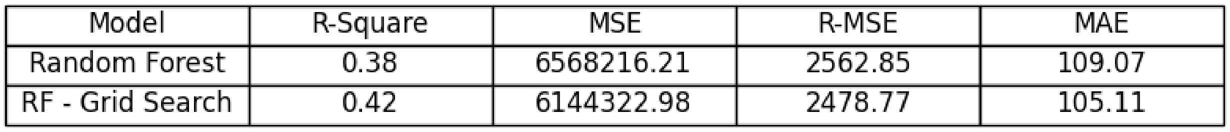


Figure [50]: Model comparison table



| Model | R-Square | MSE | R-MSE | MAE |
|---|---|---|---|---|
| Random Forest | 0.38 | 6568216.21 | 2562.85 | 109.07 |
| Linear Regression | 0.43 | 6063429.01 | 2462.4 | 397.4 |
| Lasso | 0.37 | 6676972.77 | 2583.98 | 353.16 |
| XGB | 0.28 | 7641913.73 | 2764.4 | 107.44 |

Figure [51]: Model comparison with optimized version



| Model | R-Square | MSE | R-MSE | MAE |
|---|---|---|---|---|
| Random Forest | 0.38 | 6568216.21 | 2562.85 | 109.07 |
| RF - Grid Search | 0.42 | 6144322.98 | 2478.77 | 105.11 |

Figure [52]: Feature importance of random forest

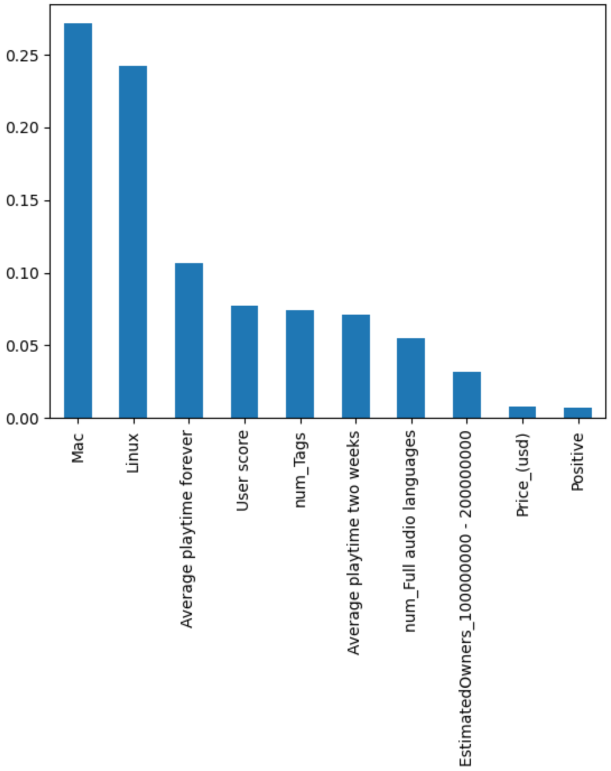


Figure [53]: Feature importance of XGB

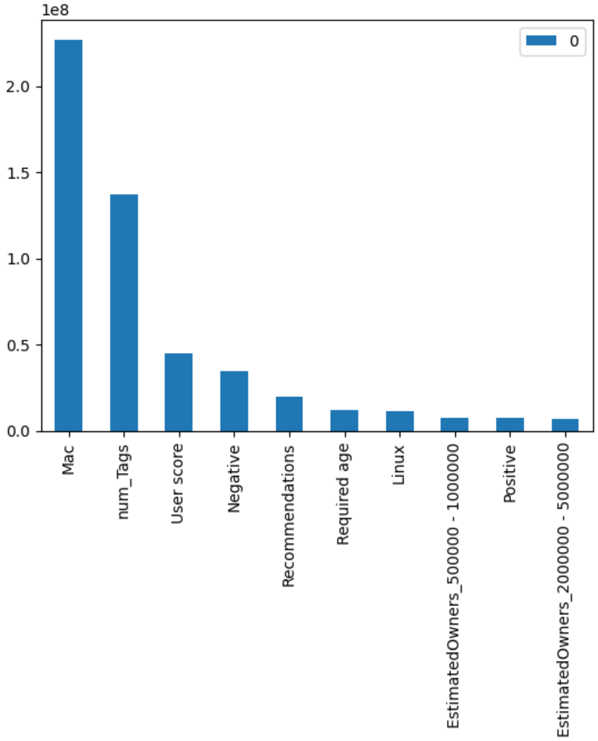


Figure [54]: Feature importance of optimized random forest

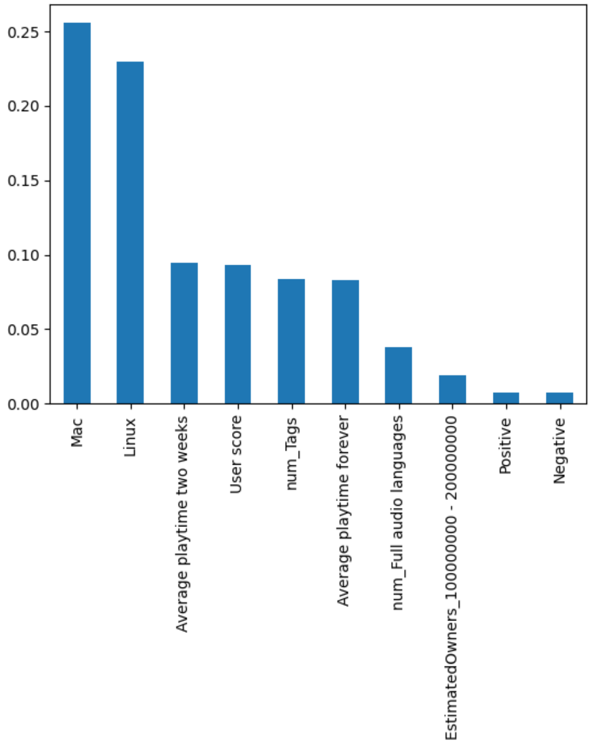


Figure [55]: PCA two-dimensions visualization

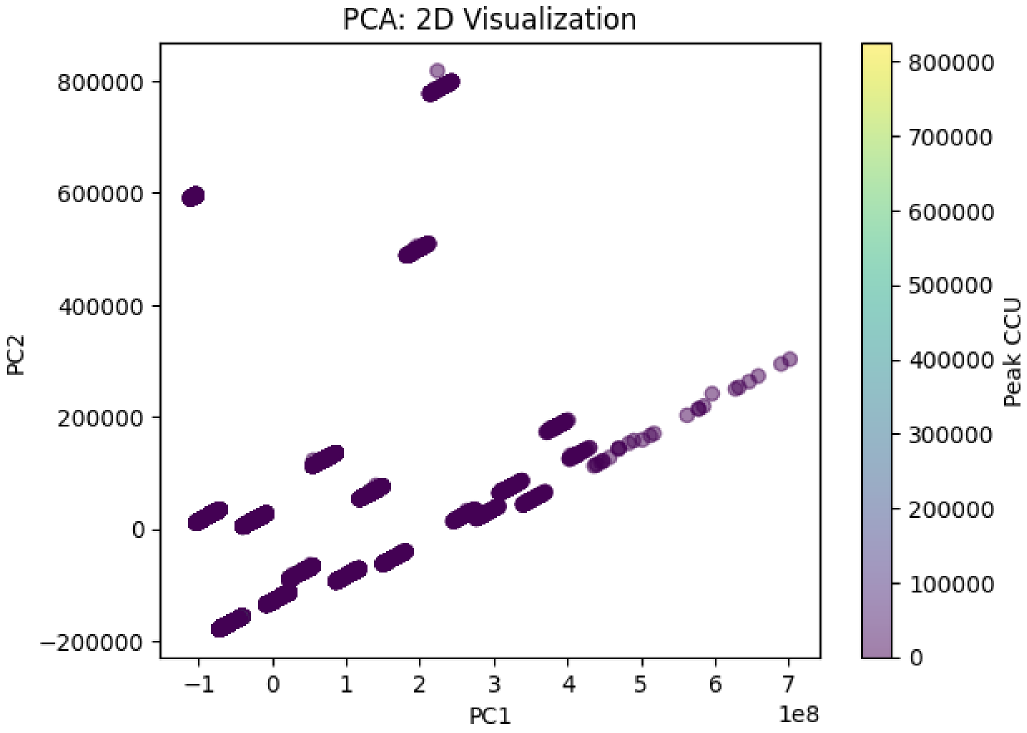


Figure [56]: Elbow function for clustering

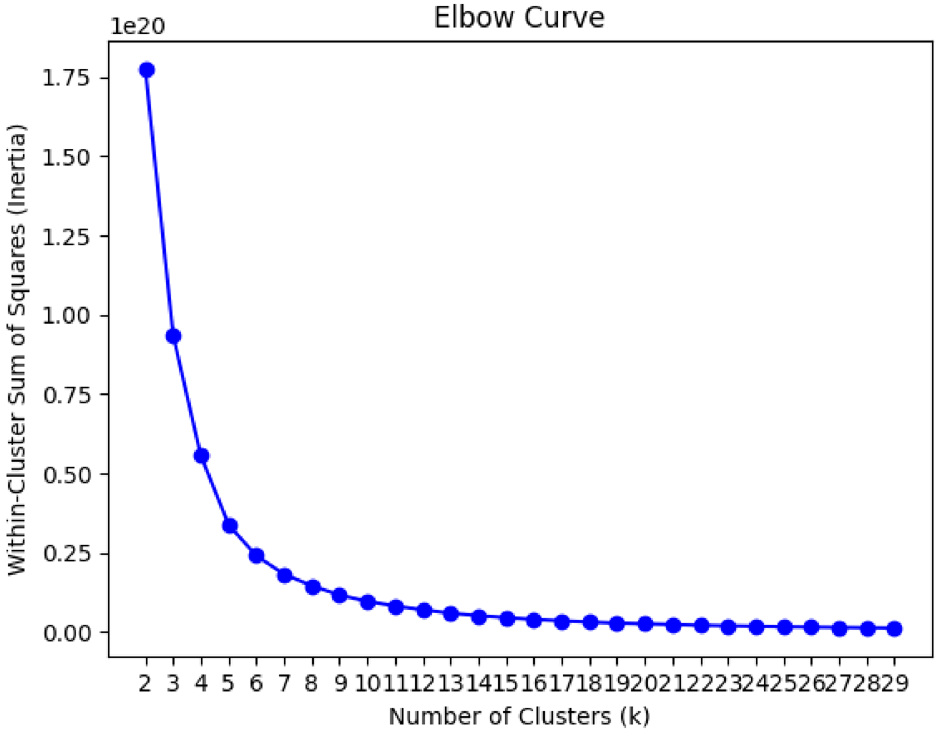


Figure [57]: Clustering visualization of normalized features

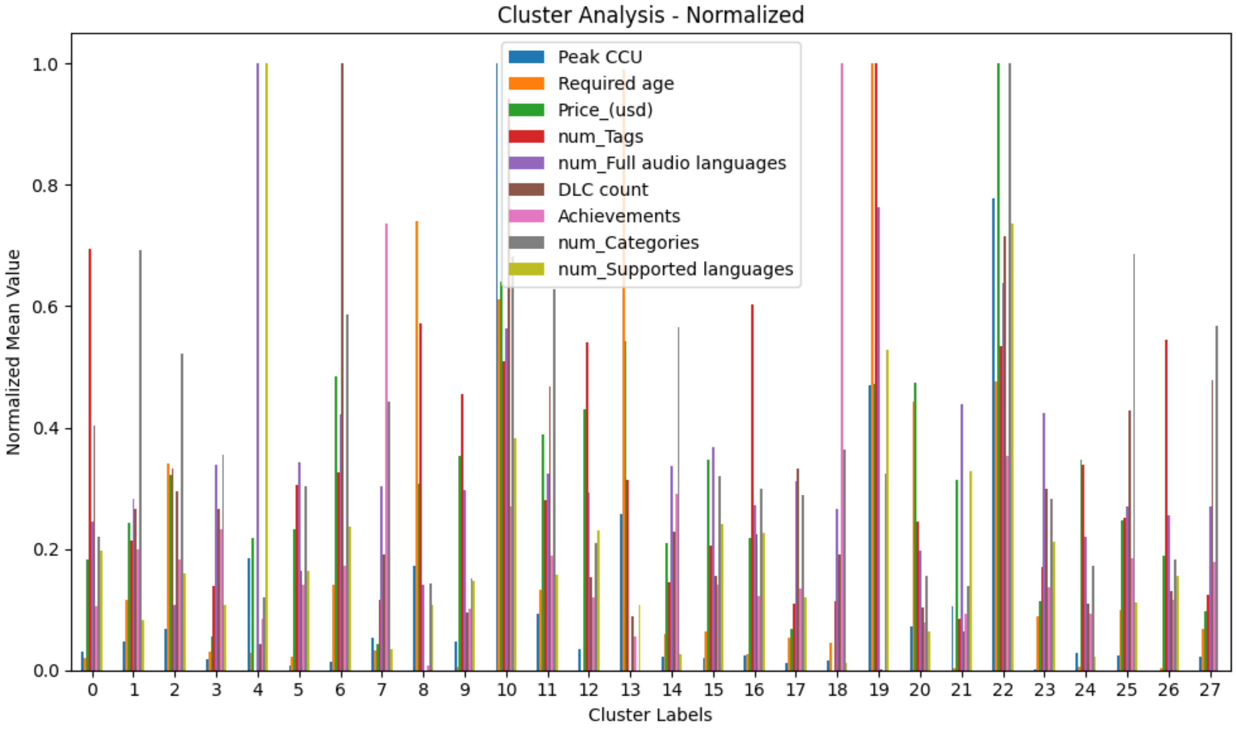
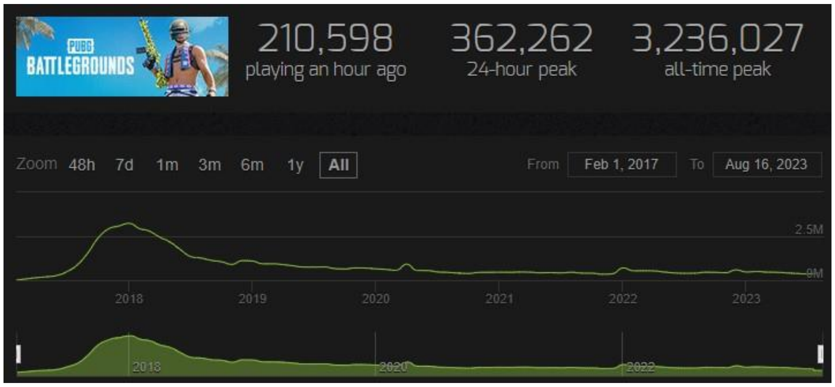


Figure [58]: Peak CCU over time for the game PUBG

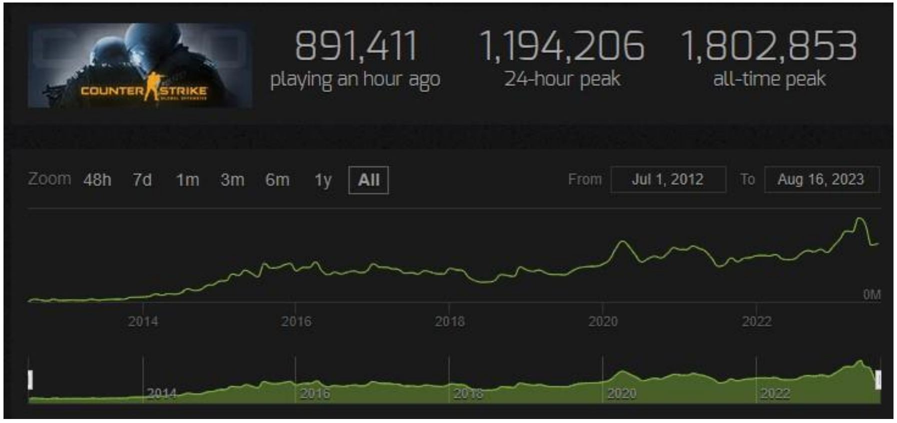

210,598 playing an hour ago

362,262 24-hour peak

3,236,027 all-time peak

From Feb 1, 2017 To Aug 16, 2023

Figure [59]: Peak CCU over time for the game CSGO


891,411 playing an hour ago

1,194,206 24-hour peak

1,802,853 all-time peak

From Jul 1, 2012 To Aug 16, 2023

Figure [60]: Peak CCU comparison over time of PUBG and CSGO

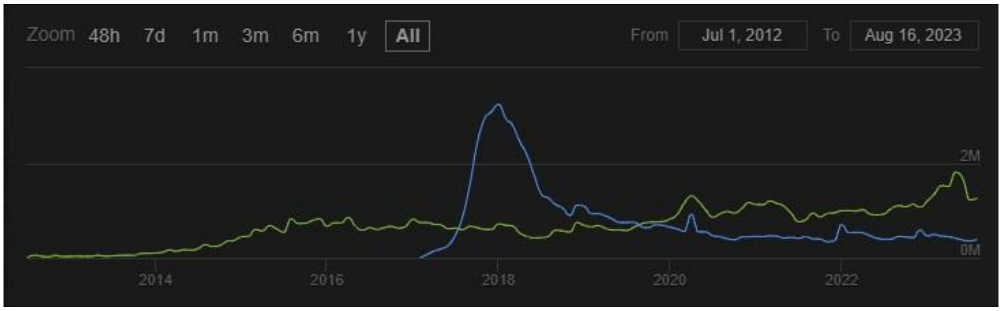

From Jul 1, 2012 To Aug 16, 2023

Figure [61]: Histogram of the frequency of log peak CCU

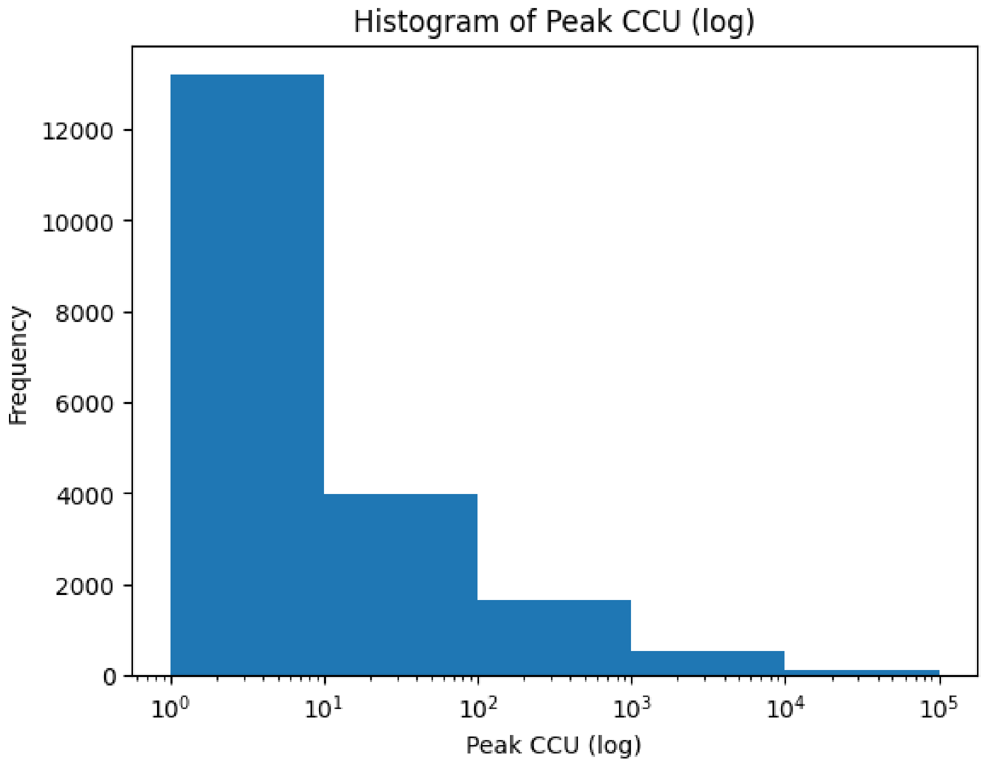
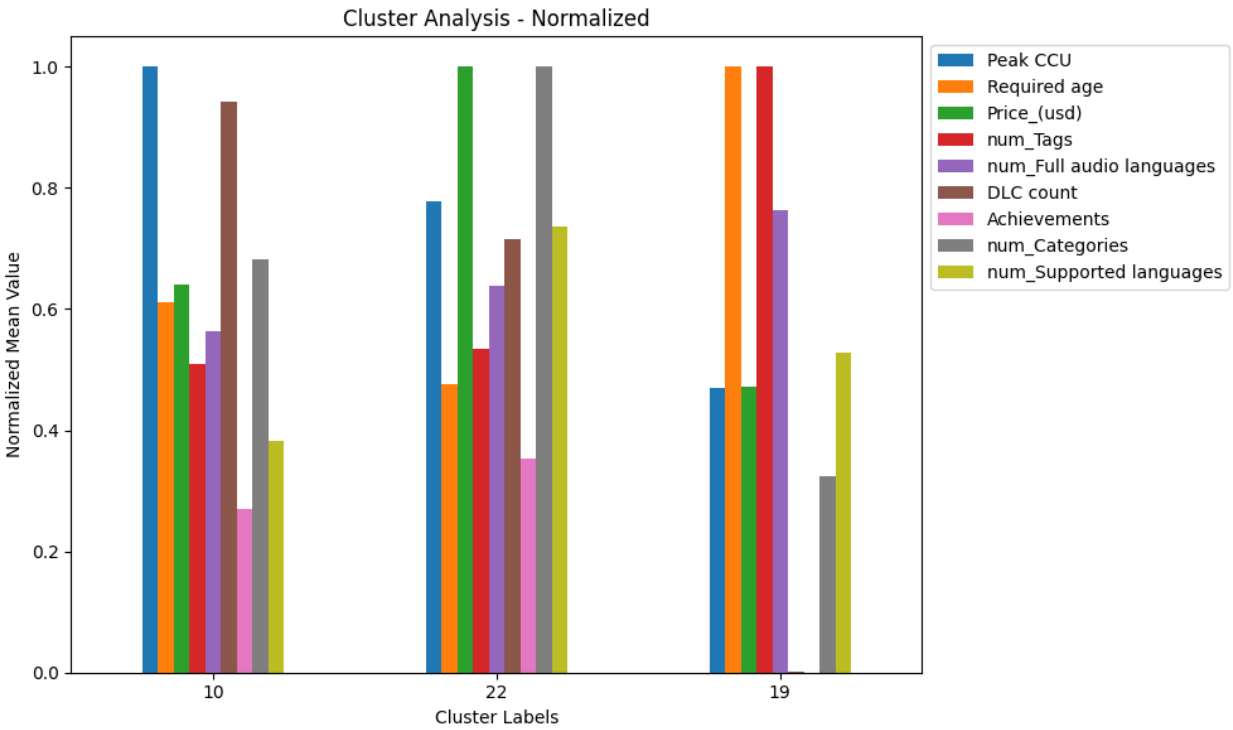


Figure [62]: A closer look at the 3 clusters with the highest peak CCU


Cluster Analysis - Normalized
Normalized Mean Value
Cluster Labels
Peak CCU
Required age
Price_(usd)
num_Tags
num_Full audio languages
DLC count
Achievements
num_Categories
num_Supported languages