Group 5: Project Report on Steam Game Data

CS 456: Data Mining

Adrian Rivera, Ian Seymour, Guillermo Leon

November 30, 2025

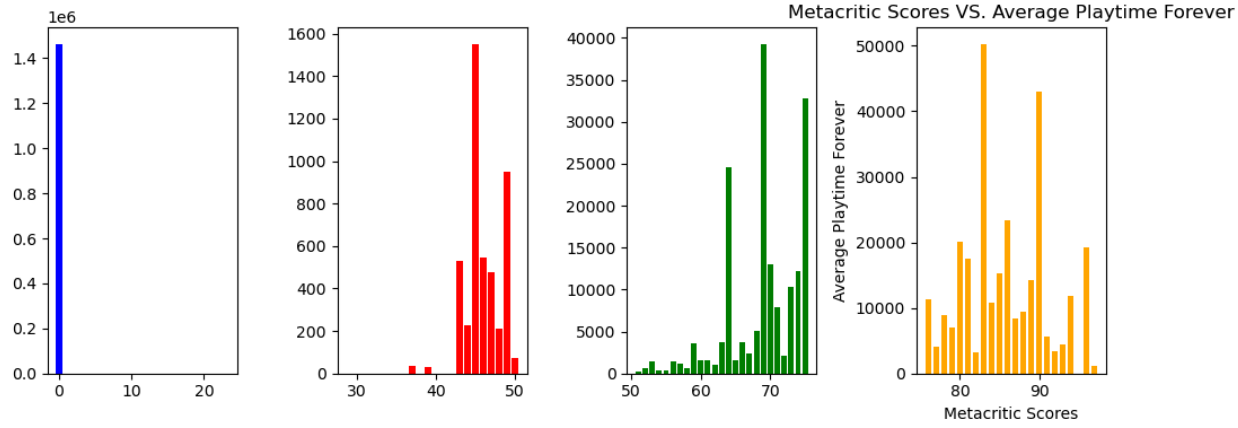
Introduction

The Steam Games Dataset was created by webscraping from the Steam platform to gather broad data that is publicly available on the numerous videogames sold and played on Steam. The dataset contains 46 columns with attributes ranging from categorical information on games (genre, category, tags), reviews (players and critics), playtime, price, game developers, and various identifiers for games (including name and Steam's AppID that is used to track games on Steam). The data provides an interesting challenge for data mining with its high amount of data on a topic many have an interest in.

Results

Part 1: Do review scores and recommendations affect the average playtime, amount of people playing games at one time, and price or discounts?

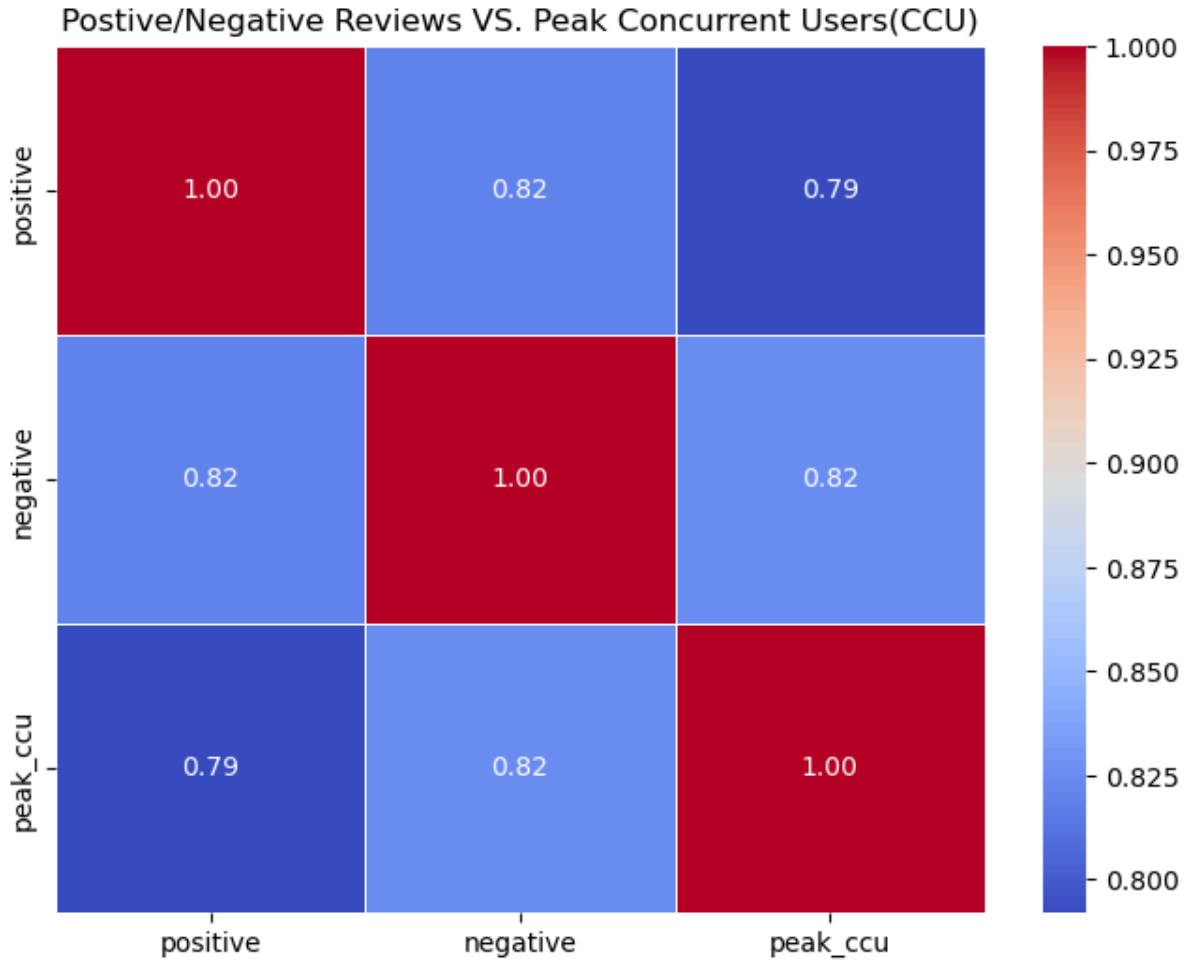
Question 1: Does metacritic scores affect the average playtime forever?



The purpose of these bar graphs is to show how different scores from metacritics will affect how the player on average will play throughout the lifetime of owning a game. When first created the graphs were all plotted onto a single graph, however being that the data has a lot of metacritic scores with '0' this created a hard to read graph with no way to see how the other data compares to each other. With this in mind I broke the graph into sections in order to show the data more legibly. Reading from left to right the metacritic score from '0-25', '26-50', '51-75', and '76-100' are on the x-axis and for all four graphs the average playtime forever is the y-axis. In the graphs we can see that there still is an overwhelming amount of '0' metacritic scores, so much so that it is hard to see any effect with playtime. However, when we take a look at the scores as they get higher we can see that the average amount of playtime slowly increases the higher the score is. We see this start to happen around the 43 metacritic score as the amount of playtime does increase rather significantly, then at around the 55 score mark, the playtime average begins to climb up even more so, and again at the score of 75 and higher. With the harvest we can see that overall while there are some playtimes being high with scores under 50

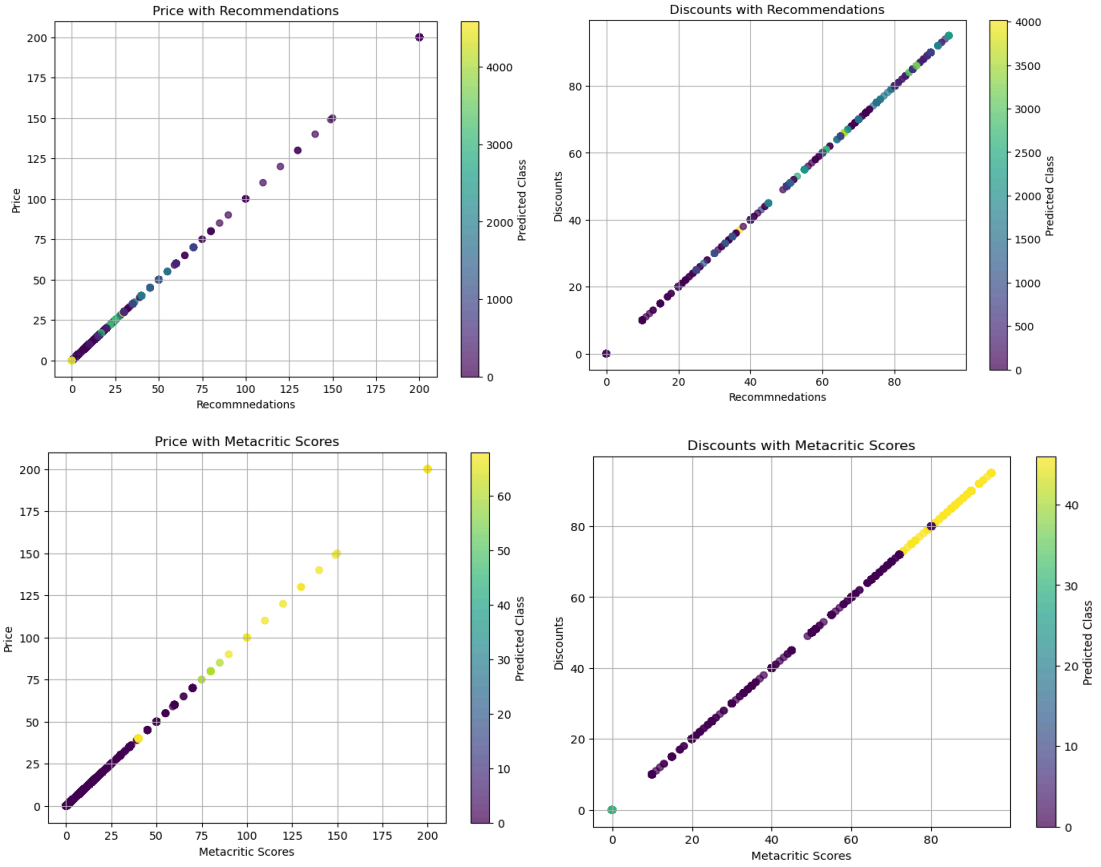
they are not nearly as high as what they are when you start getting into games with a score above 55 and 75.

Question 2: Do postive/negative reviews affect peak concurrent users?



The purpose of the heatmap is to show if there is any correlation between a game receiving positive or negative reviews and which has a higher effect on a games peak concurrent users (CCU). With this I opted to use a heatmap in order to showcase higher and or lower correlation between positive and negative reviews and also the peak CCU count. As we can see, marginally, negative reviews have a higher correlation than positive reviews when it comes to peak CCU. This was rather surprising as some would think that this would be the other way around. A possible explanation for this could be that a game is overall negatively reviewed, but people would still want to give it a try, another being the game has mixed reviews with more of an overwhelming negative, or even a game being review bombed if the fan base does not like a decision or statement the company made or said which has been done in the past.

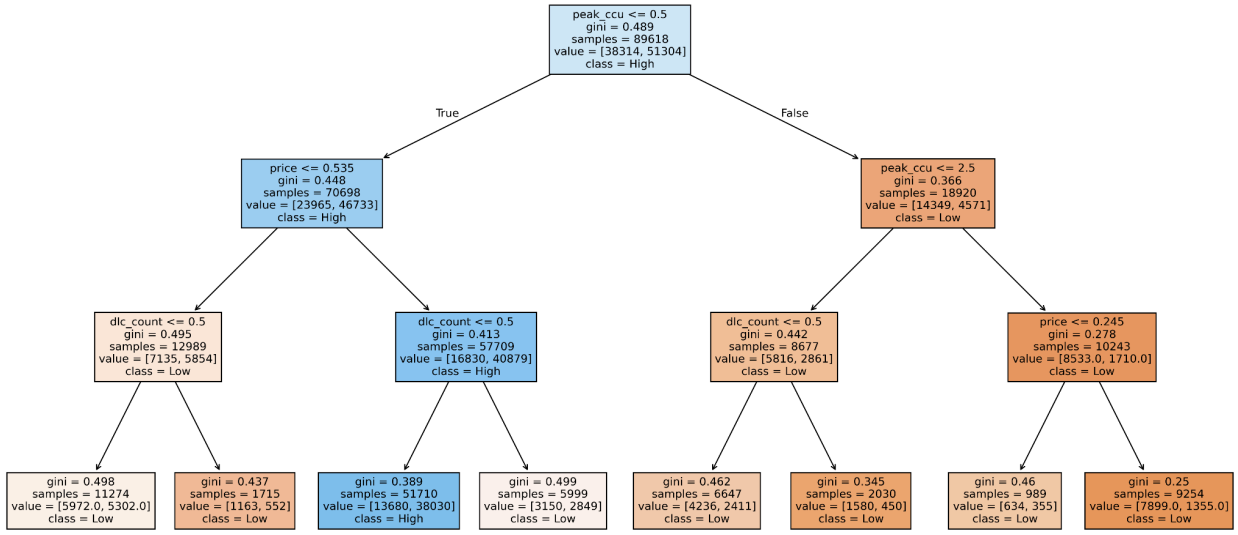
Question 3: Is price or discounts affected by the amount of recommendations and metacritic scores?



The purpose of making these naive bayes visualizations is to show if a higher metacritic score will affect how much a game will be priced at, and if a game was put on a discount how much the price would be with it. The other purpose was to also show if the higher the recommendations a game receives, will that also affect the price and or discounts for the game. With all the graphs we can see that there are in some ways that the price can be affected by either the amount of recommendations or the metacritic score. Though there were some outliers when it came to price with both as there are a few games that are higher than $80 with the biggest outlier being a game valued at $200. Though this problem did not arise when it came to showing the discounted price for both recommendations and metacritic score. The highlights I would say are seeing how it is predicted that if a game has a higher metacritic score there is a higher possibility that the game could go on sale at a discounted rate, but this was not the case for having higher recommendations, where as for the price itself could be higher if the metacritic score is high and lower if it is low, but the recommendations amount did not have as high of an affect on it as the metacritic scores did.

Part 2: Guillermo Leon

Question 4: Which game characteristics mostly strongly predict whether a game receives High or Low review scores?



The purpose of this decision tree classifier is designed to determine whether a game is likely to have High or Low review ratings based on measurable game attributes. This model analyzes numerical features and repeatedly splits the data at thresholds that best separate High-rated from Low-rated games. In the decision tree plot, nodes represent decisions made using game features for example peak_ccu < = 0.5. Branches show how data is split based on these conditions, and the leaf nodes at the bottom represent the final predictions of a High or Low ratings. The diagram flows from top to bottom, showing how the tree narrows from all games to a specific group based on feature thresholds. The big take away from the decision tree is that the tree prioritized peak_ccu, price, and dlc_count over playtime features. My interpretation of this is that for peak_ccu we receive high engagement often correlates with popularity and social validation, which leads to higher positive reviews. But on the contrary lower peak_ccu can be associated with poor retention which leads to lower review scores. The price features might be because very cheap or free-to-play games can attract large audiences but may have polarizing opinions. Or a higher priced game usually indicates a higher production or quality game which can correlate with higher reviews. And the dlc count indicates that games with more dlc are supported long-term by developers and have more replay value which can correlate higher satisfaction and positive reviews, and the opposite for games that have lower counts in dlc.

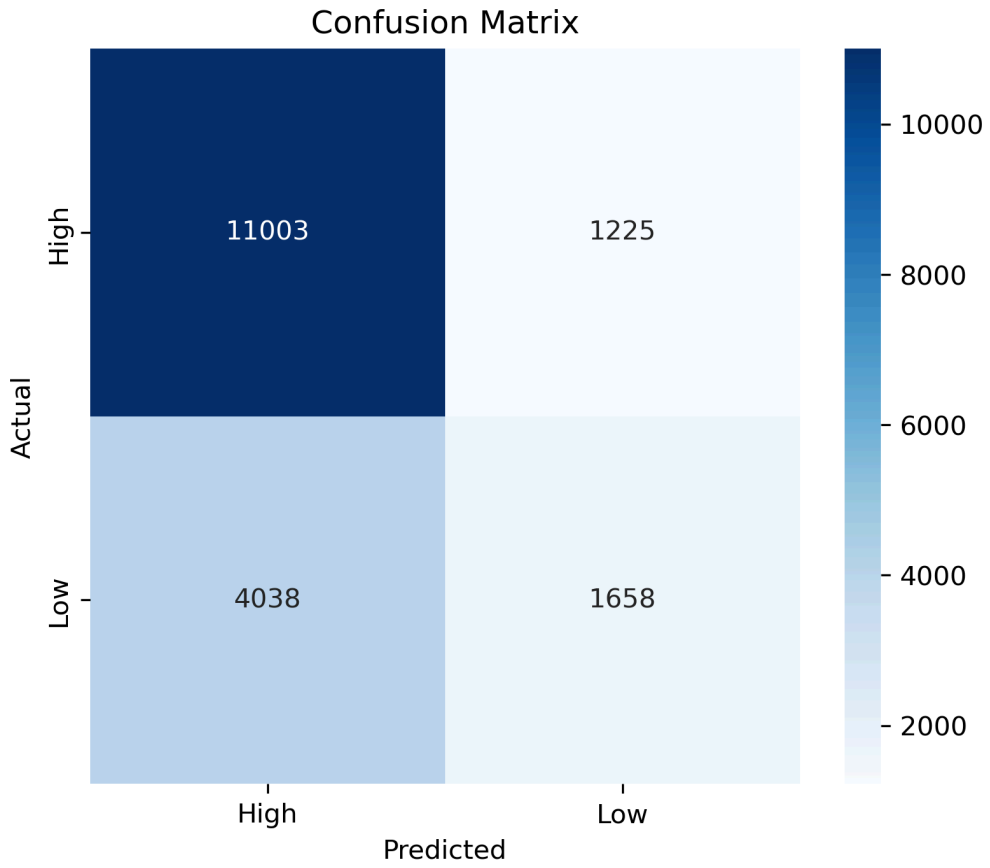
Question 5: Based on the price of a game, is it likely to have mostly positive reviews (“High”) or mostly negative reviews (“Low”)?

```
Classification Report:
              precision    recall  f1-score   support

        High       0.73      0.90      0.81     12228
         Low       0.58      0.29      0.39      5696

    accuracy                           0.71     17924
   macro avg       0.65      0.60      0.60     17924
weighted avg       0.68      0.71      0.67     17924

Confusion Matrix:
[[11003  1225]
 [ 4038  1658]]
```



The purpose of this analysis is to predict whether a game is likely to receive mostly positive or negative reviews based solely on its price. This can help identify patterns such as whether more expensive games tend to be better received. The feature I used was price, and the target variable was review_category which was labeled “High” if the game had more positive reviews than negative “Low”. The model uses the Gaussian Naive Bayes which assumes price is normally distributed within each class. And computes the probabilities for each class and predicts the one with the highest probability. I evaluated this data by splitting it into training and test sets of

(80/20). Then calculated the classification report and confusion matrix. The confusion matrix has an x-axis labeled predicted class (High or Low) and the y-axis is labeled actual class (High or Low). Within the four cells it will count the predictions, the top left predicts the true high as high (correct), the top right predicts the true high as low (misclassified). The bottom left predicts the true low as high (misclassified) and the bottom right predicts true low as low (correct). Some key takeaways from the classification report is that overall accuracy is 71% which is pretty decent given the fact of how imbalanced the dataset is. It predicted most popular games correctly with a recall of 0.90. Although price alone might not be sufficient enough to capture the full complexity of review outcomes. Compared to other features, price was more evenly distributed across games, making it more reasonable as a starting point for the Naive Bayes model.

[Question6]: Which platforms tend to appear together for the same games?

Association Rules: 12

antecedent_str	consequent_str	support	confidence	lift
windows	mac	0.194336	0.194401	0.999073
mac	windows	0.194336	0.998738	0.999073
windows	linux	0.140798	0.140845	0.999622
linux	windows	0.140798	0.999287	0.999622
mac	linux	0.104644	0.537791	3.816881
linux	mac	0.104644	0.742694	3.816881
windows, mac	linux	0.104633	0.538413	3.821295
windows, linux	mac	0.104633	0.743145	3.819196
mac, linux	windows	0.104633	0.999893	1.000228
windows	mac, linux	0.104633	0.104668	1.000228

Purpose: The purpose of this analysis is to identify patterns between platforms (Windows, Mac, Linux) in the dataset of games. If a game is available on one platform, which other platforms is it likely to be available on. This can help us understand platform relationships and can inform decisions such as marketing, cross-platform support, or targeting a multi-platform release. Methodology: The platforms were encoded as boolean columns, and used the Apriori algorithm to generate frequent itemsets and then derived association rules with confidence >= 0.1. I only considered rules where both antecedents and consequents have at least one item. Explain the graph: Antecedents is the platform for the “if” condition. Consequents are the platforms that form the “then” condition. Support is a fraction of all games containing both antecedent and consequent. Confidence is the probability that the consequent occurs given the antecedent. Lift is the strength of association, >1 means positively correlated. Harvest Highlights: Windows and Mac are frequently paired, which suggest that games on Mac are often also on Windows. Linux tends to occur with Mac or Windows in a smaller subset of games but with strong lift values which mean positive associations. Some of the highlights show that multi-platform like “if Windows, Mac then Linux” suggest that games supporting the major platforms are more likely to

support Linux as well. The analysis overall confirms that platform availability is not random and certain combinations occur consistently.

Part 3: Most Important Features for Positive Steam Recommendations, Release Timing, and Clusters

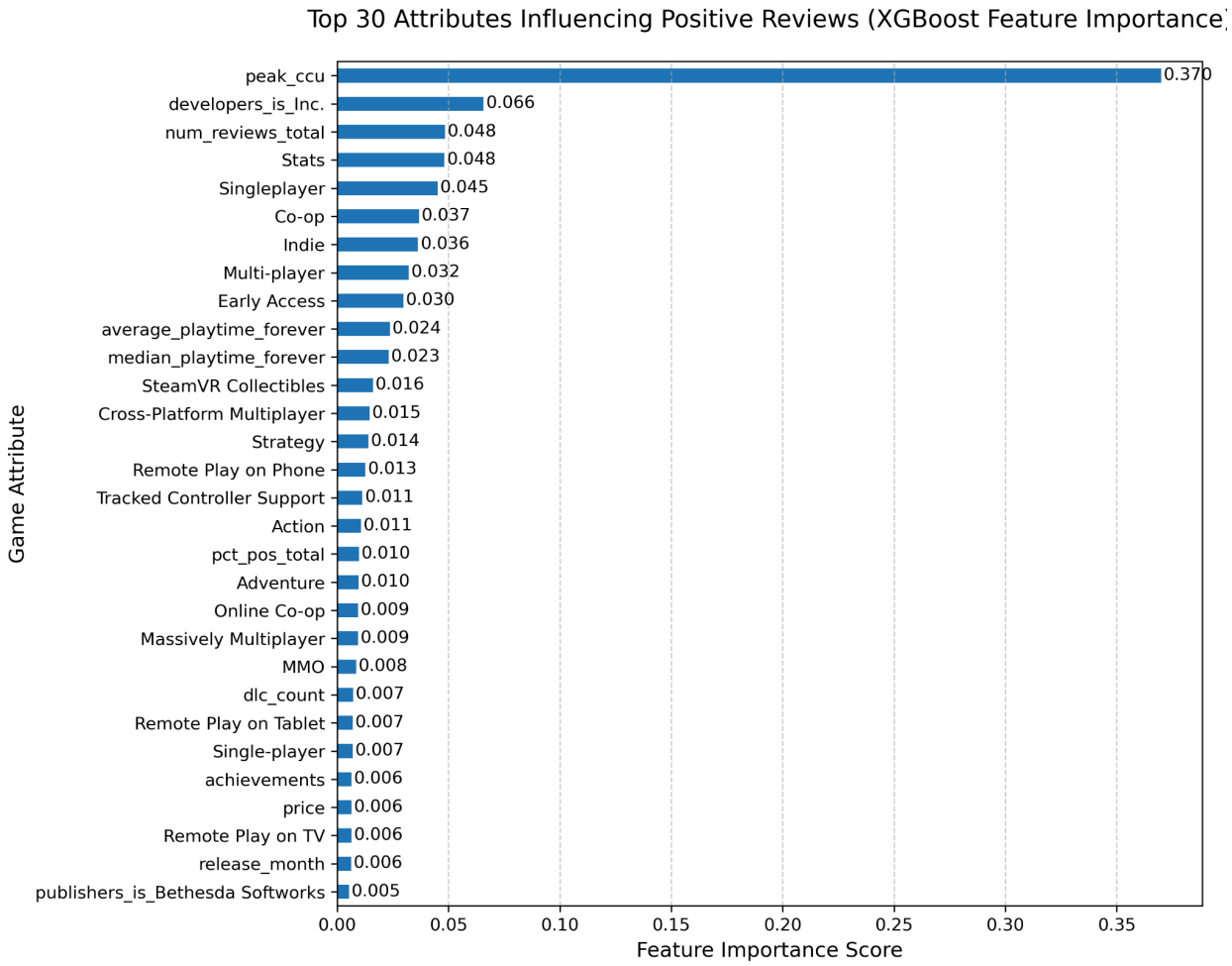
Question 7: What attributes have the most impact on positive reviews?

7.1 Purpose

The intent of finding the most important features for positive Steam recommendations is important to understand what features, if any, consistently lead to high-performing games. These results will also likely reveal if positive Steam recommendations actually leads to more play-time and satisfaction from users, or if users are motivated to leave positive recommendations for other reasons (perhaps a low sale price).

7.2 Methodology

To find the most important Steam game attributes, XGBoost Regression was used due to its effectiveness on large datasets that have potentially unbalanced, or sparse data (unbalanced when one or two data points may be substantially larger than the rest). Preparing the data required dropping columns that would not be useful, like the names and identifiers of games. Also, data that included notes or descriptions of games were also dropped from the dataset. For the release date, the day was dropped but the month and year were split into two new columns. Numerical data was scaled and non-numeric data was One-Hot-Encoded. The methods used focused on minimizing computing overhead in order to perform on lower-end machines. The XGBoost Regression was assessed by determining the percentage of variance that the model accounts for by finding the root mean squared error rate for the model. The model results in 30% variance accounted for, meaning that it is likely accurate for the most important game attributes, there is still another 70% of unaccounted for variance that can influence how many positive reviews a game receives.



7.3 Graph Explanation

The x-axis represents the importance score for the specific game attribute, which is the result of the XGBoost regression that returns values for how important a particular attribute is to predicting whether or not a positive recommendation is left for a game. The y-axis represents the top 30 game attributes for their importance to positive reviews.

7.4 Data Insights

The results show the most important features for predicting positive recommendations for games. The top attribute is having high peak concurrent users (peak-CCU), which may seem unsurprising but it does confirm that games that players give lots of positive recommendations to are also the games that have high amounts of people playing. The importance of peak-CCU means that players are not most influenced by a game's price, or other features unrelated to gameplay, and that players actually do play the games they say they like. However, it is worth noting that average and median playtime is not ranked as highly as peak-CCU (but still quite important). This means that the games that are most likely to have lots of positive reviews are likely what can be considered "blockbuster" games where they might have high peaks of people

playing but might not have players stick with the games for a long time. There are a lot of factors that could skew average and median playtime, to include a game's length and genre. The presence of Steam Stats (achievements, player rankings, etc.) near the top is also a big predictor of a game being well received by players, indicating that general player engagement is important. A game being developed by a larger company is also a predictor of its success, with "developer_is_Inc." near the top which means the developer who made the game is an incorporated company. Bethesda Softworks is named specifically as a top predictor when they are listed as a game's publisher, but this is at the very bottom of the top 30 features. Genre, category, and tags do have an influence as well but a large variety can be seen in the top 30 showing that one does not likely dominate for top features (note: genre, categories, and tags all have attributes that may be repeats of each other, "singleplayer" is both a category and a tag and appears in the list twice).

Question 8: Does release date timing influence positive reviews?

8.1 Purpose

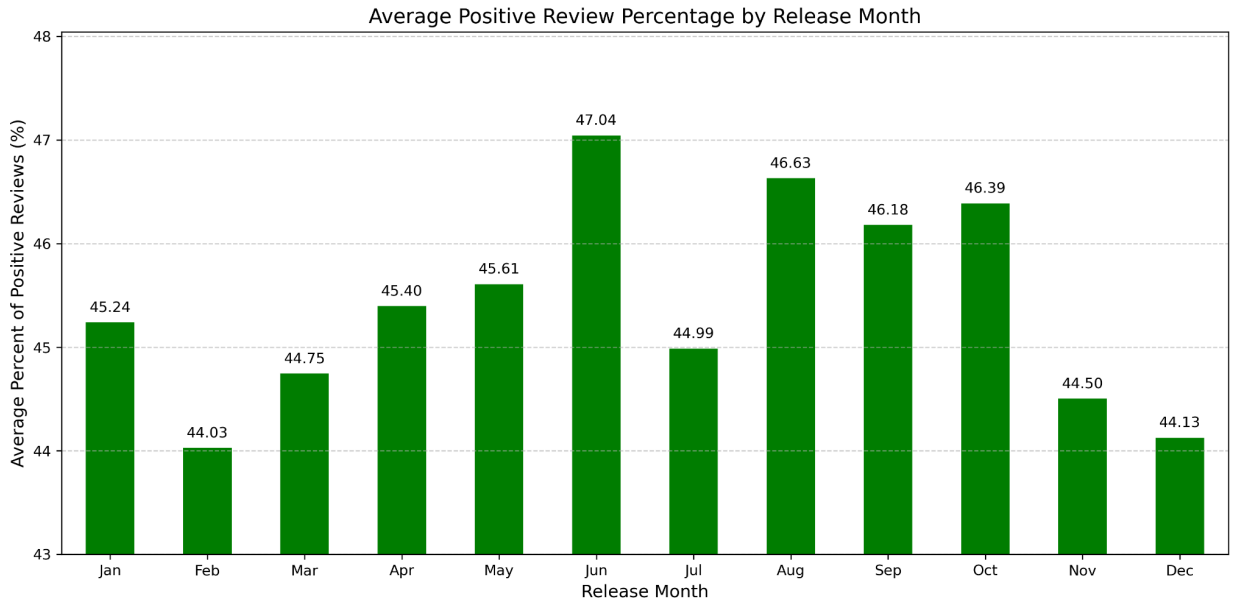
The previous XGBoost Regression shows that release month did make it into the top 30 features most important to predicting a game that receives positive reviews. This does not reveal which month is best to release a game in. Also, the release year might have an impact as well. Knowing the impact of the time a game is released is important, especially if a publisher intends to maximize positive recommendations from Steam players.

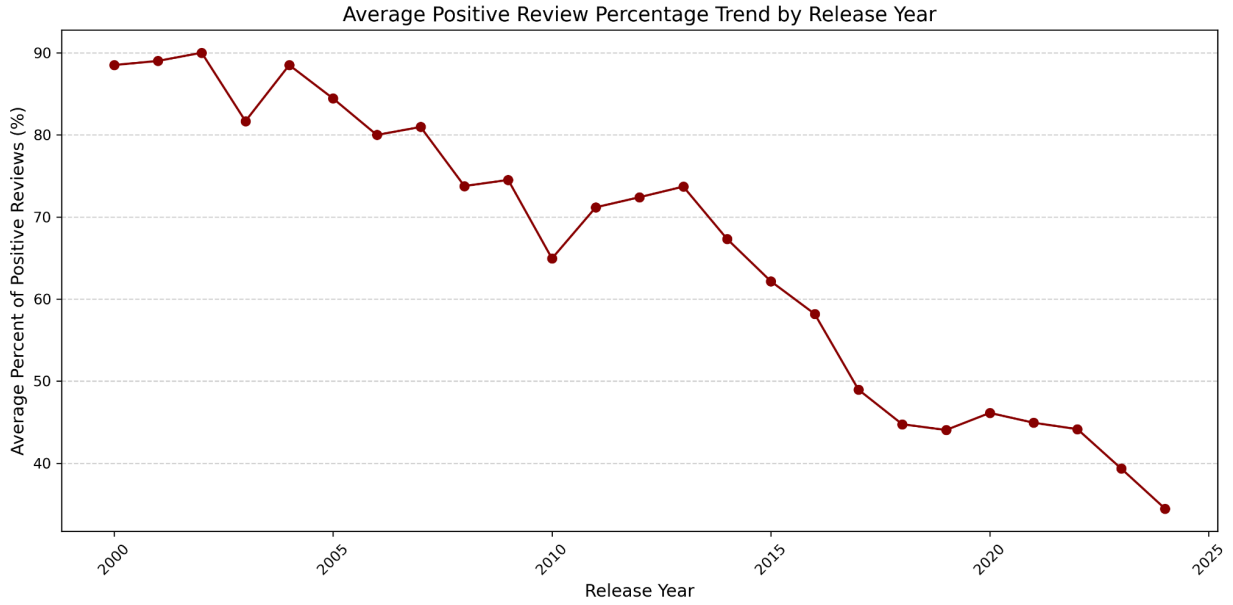
8.2 Methodology

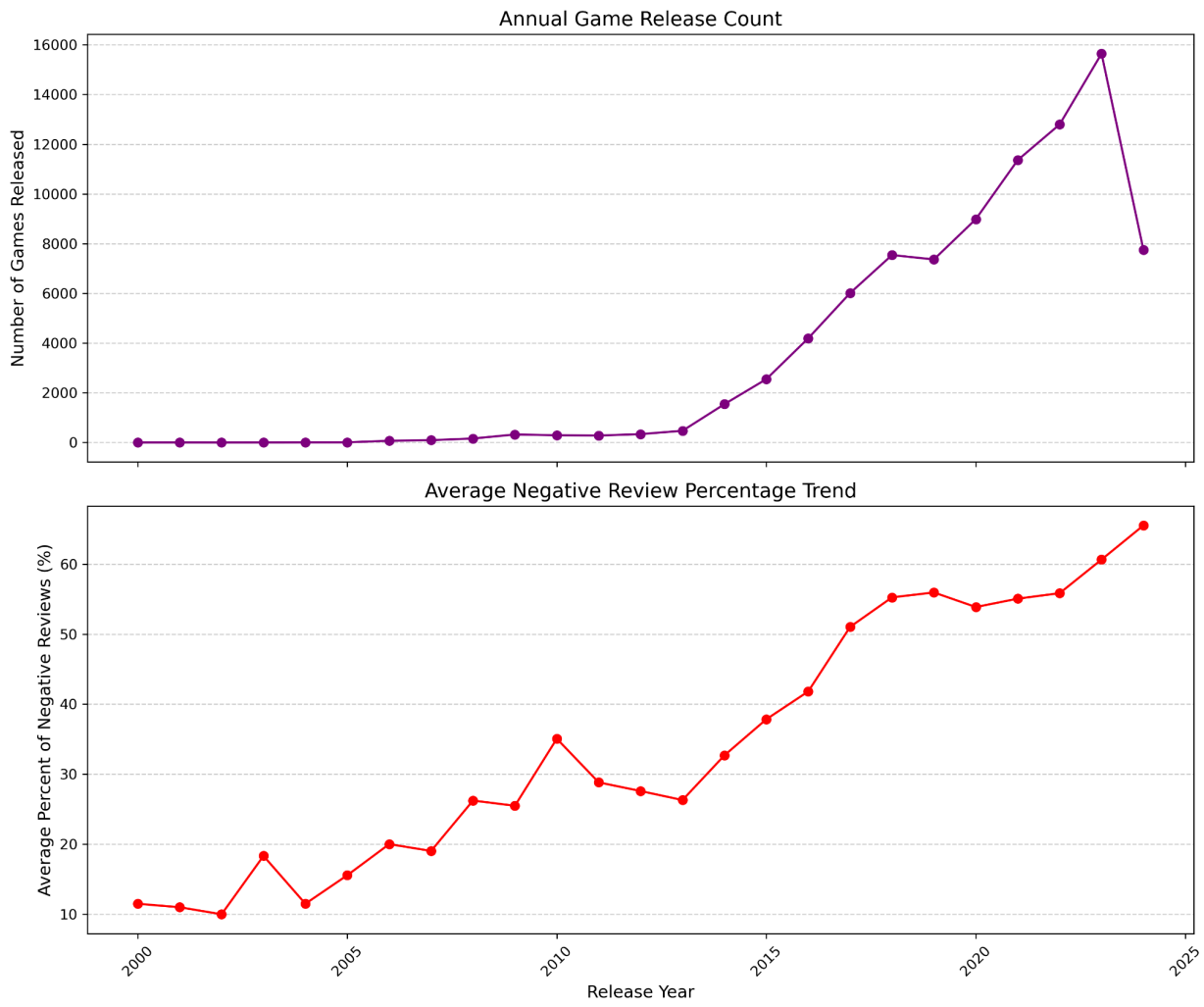
For analyzing release date, a key change will be made to how the data is analyzed: percent of positive reviews will be used rather than the raw count of positive reviews. This is to eliminate the possibility that an older game has simply had more time to collect positive reviews than a new game. To determine what month is optimal for releasing a game if the goal is positive reviews, a simple mean percent of positive reviews will be calculated for each month. The same methodology will be used with years to create a time series of average positive review percentage for games released in each year since 2000.

8.3 Graph Explanation

The first graph presents each month of the year on the x-axis and the y-axis represents the average percent of positive reviews games released in that month received over their lifetime. The second graph shows a time series chart of average positive review percentage of all games released for each year from 2000 to 2024. Additional graphs are also included: a comparison of the number of games released each year side-by-side with the average percentage of negative reviews for each game released in every year 2000-2024.


Average Positive Review Percentage by Release Month
Average Percent of Positive Reviews (%)
48
47
46
45
44
43
45.24
44.03
44.75
45.40
45.61
47.04
44.99
46.63
46.18
46.39
44.50
44.13
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
Release Month


Average Positive Review Percentage Trend by Release Year
Average Percent of Positive Reviews (%)
90
80
70
60
50
40
2000
2005
2010
2015
2020
2025
Release Year



8.4 Data Insights

For release month, it is immediate that June and the Fall months are optimal for releasing a game. However, it is important to note that the difference between the best month and worst month is only about three percent. It is possible that June performs well due to its position to to the school schedule, where many players may have more time to play games in summer. Fall is also likely when players purchase games leading into the holiday season, and that games released later in the holiday season perform worse.

The year analysis shows perhaps one of the most interesting results: players seem to be increasingly dissatisfied with games year-over-year. The exact reason for this is not shown in the data, but there are a few convincing explanations. First, it is critical to know that Steam did not begin hosting its original recommendation system until 2010 and rolled that system into a player review system in 2013. Steam as a platform was initially released in 2003. These start dates for game reviews may partially explain the results. Steam is also a curated selection of older games with the only ones being on its platform are the older games that are still compatible with modern systems or were likely popular enough to have been modified to function on modern systems. This means that older games on Steam are likely the best of the older games, and as

such, likely have a higher percent of positive reviews. However, some very popular games were released in the past decade, so this explanation does not likely account entirely for this declining trend in player satisfaction. Another possible explanation is that the game market is becoming increasingly saturated with low quality games (Steam itself facilitating easier release pathways for smaller developers, but potentially those with less experience and resources as well, where older games likely needed the help of a publisher to be released). When plotting a comparison between the number of games released each year (on Steam) and the average percent of negative reviews for games released in those years, it shows that two lines increase at a similar rate. Though, this comparison also shows an interesting anomaly where 2020 saw a small decline in negative reviews while studios continued producing more games. The anomaly in 2020 could be due to people having more time to play games, and in the previous analysis, it seems that the more time players spend playing a game, the more likely they are to also give it a positive review. The causality between time spent playing a game and player satisfaction toward the game may be less straightforward than initially thought, and potentially more time to play a game leads players to be more favorable toward reviewing it.

Question 9: Can the Steam data be shaped into clusters?

9.1 Purpose

Categorizing games into clusters may yield predictive power to see what games are similar, what attributes are a part of these clusters, and if some clusters of games are more likely to receive positive player recommendations than others.

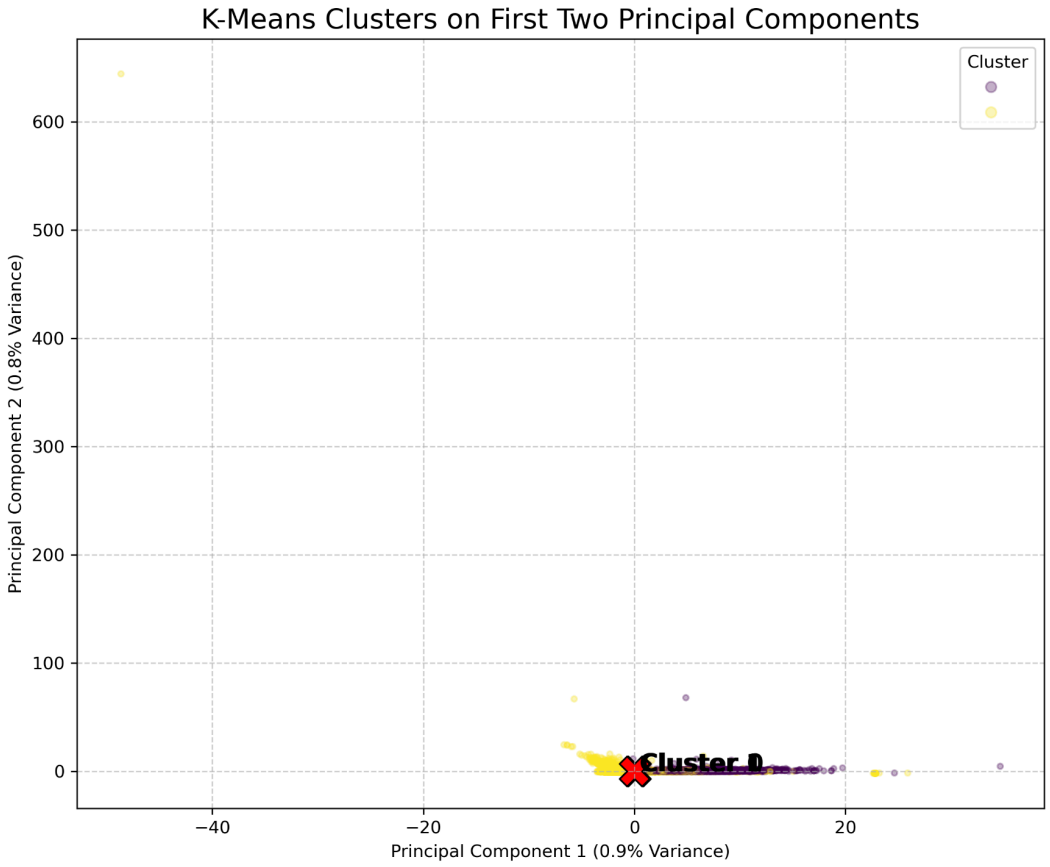
9.2 Methodology

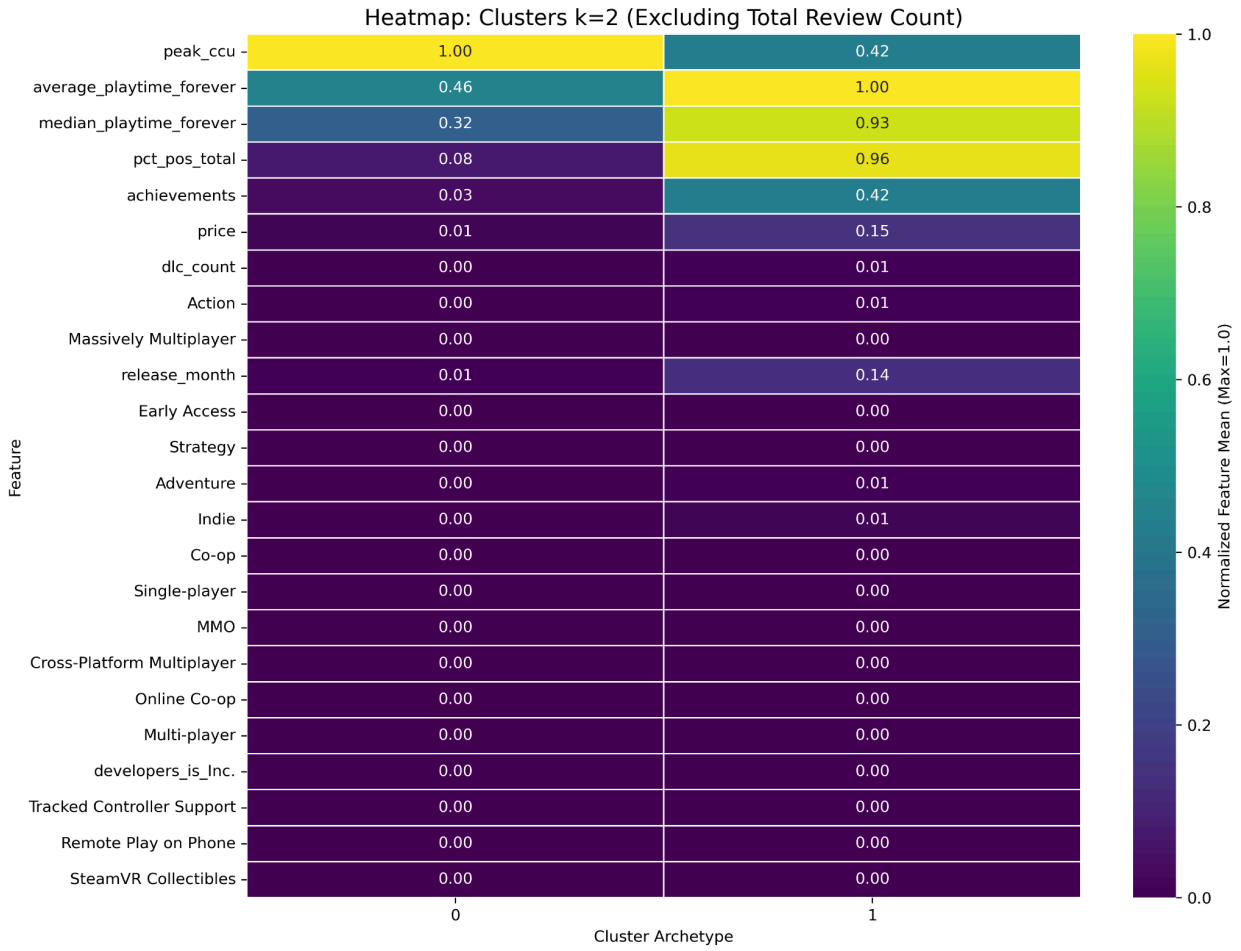
K-Means was used to find the clusters with the optimal k value being found by assessing silhouette score. Initially, $k = 8$ seemed optimal when considering all the data together, but it resulted in clusters that were dominated by near zero values and had a large amount of unaccounted for variance and a few very high values skewing the clusters. This was due to the nature of the data, which is broad and sparse. To improve this, K-Means was attempted a second time with only the top 30 attributes identified by the XGBoost Regression to hopefully reduce the impact of the specific extreme attributes as well as the many near zero values. For only the top 30 attributes, $k = 2$ was optimal according to silhouette scores. The $k = 2$ model was tested using K-Nearest Neighbors to determine accuracy, which was about 91% (an acceptable accuracy). However, the $k = 2$ model had an equally high amount of unaccounted variance in the data with the model only representing about 1.7% of the total variance. This means that there is likely a lot not being captured by the model, even if it is accurate.

9.3 Graph Explanation

The first graph is the two clusters produced by K-Means with principal component 1 as the x-axis and principal component 2 as the y-axis. The resulting clusters are grouped extremely close together (with an outlier game up at the very top left of the graph, meaning the graph does not closely show the two clusters). The graph could have been adjusted to exclude outliers, but this would not have captured how truly messy the K-Means results were, which can be seen not

only in the outliers but the blending of the two clusters. The second visual is a heatmap of the top 30 defining features of the two clusters (without number of reviews included, which was ranked equally at the top of both and far above all other features, to better show the variance of the other 29 features).





9.4 Data Insights

The games on Steam cluster into two groups. Cluster 0 represents games that were “blockbusters” with a high peak-CCU, sizable average playtime, but not necessarily a high percent of positive reviews. Cluster 1 represents games that might not have had a large number of concurrent players but have higher total playtime and substantially higher amounts of positive reviews. Cluster 1 represents the games that have long-term player satisfaction and staying power, rather than the big blockbusters. While it is possible to divide games into these two clusters with a model that has 91% accuracy according to K-Nearest Neighbors, there are a lot of very low values included in the heatmap and the clusters are driven by a small number of attributes. Since the low total variance of 1.7% for the model combined with what is seen in the heatmap, there is clearly a lot going on that is not captured by the model.

Overview

One finding that is very important for game publishers is that games that receive larger numbers of positive reviews are also played more with higher concurrent players as well as total playtime. This may sound obvious, but it does reveal a few key things. It is unlikely for a

publisher to be successful by trying to garner good player reviews through means other than simply making a game that players actually enjoy. This means that pricing a game lower, focusing on a particular genre, or focusing on the publisher or developer's brand (this only has an impact for the biggest publishers) can make up for a game that is not satisfying to players. Another important finding related to player reviews is that players are less satisfied now than they were in the past on average and the market for games has likely become saturated with many low quality releases. Naive Bayes analysis of pricing and game reviews indicates that price has an interesting relationship with reviews: higher priced games tend to receive better reviews. This might be because games with higher prices generally have more production value. Alternatively, lower priced games may draw a wider audience but result in more polarized reviews.

Contributions

Adrian: My responsibility during this project was to create a series of bar graphs as well as a heatmap and naive bayes visualization in order to show the answers to the questions I came up with to answer. Initially when I created my first question I wanted to use a different column within the data called 'reviews' which was the scores that different critics gave as a review for a game that was given based on the press release copies given out, before it was released to the public. However, when I began to create my graph with this initial column I had realised that it was not compiling since I had selected a column that not only had an integer value but also character values as well in a string. So with this I had to change the graph from reviews to 'metacritic_scores' which had only integer values with no character strings. I also ran into a problem with the graph initially when I was able to get one to show as well, since there were so many games that had a value of '0' within it, the rest of the data that was to be shown was so small you could not tell what values were for what, so I broke up the graph into four separate graphs, all of which had different values for metacritic scores so that way the data can be seen much easier. There was another issue with trying to get the labels for the title and x, and y axis to align properly but thanks to my team I was able to get it sorted out with their input using plt.tight_layout() which solved the issue. For my second question I did not really have anything that held me up as it was a pretty easy to set up heatmap. The final question I had gave me a little issue as well. Initially I made two graphs and they each had grouped both the 'price' and 'discount' columns together on the y-axis with recommendations and metacritic score as the x-axis. Though when I ran the code to see how they came out they did look exactly the same though the predicted class for each one did show a slightly different variation between the two. I then broke up the two graphs into four graphs to show price and discounts separately with both the amount of recommendations and metacritic scores. This produced much easier to read graphs with an easier to see predicted class coloring for each one.

Guillermo: For this project I analyzed a steam dataset to understand patterns in reviews, pricing, and platform availability. For my first question I used a decision tree to study whether a game would receive high or low review scores based on features such as averages and median playtime, peak concurrent users, DLC count, and price. The model highlighted which game characteristics most strongly influenced review outcomes, this provided interpretable insights into player engagement and content features. A struggle I had encountered was overfitting, so I had to limit the tree's depth to 3 branches and select only a few relevant features like peak

concurrent users, DLC count, and price. This would balance the predictive accuracy and make the tree simple enough to visualize but still have meaningful data. My second question: I used a Naive Bayes model to examine the relationship between a game's price and its likelihood of receiving mostly positive or negative reviews. I first transformed the price feature using a log scale to reduce skew, then split the data into training and test sets, and trained the Gaussian Naive Bayes classifier. Then evaluated the model using a classification report and confusion matrix. The struggles I had faced with this question was mostly related to the simplicity of Naive Bayes assumptions. While the price was more balanced than other features, the actual distribution of "High" vs "Low" review games was still somewhat skewed. Which caused the model to predict the majority class better than the minority class. This led to lower recall for the "Low" review games, as seen in the confusion matrix. For my third question I used Association Rules to identify which platforms (Windows, Linux, Mac) tend to appear together in games, which uncovered occurrence patterns. I prepared the platform columns into boolean values and applied the Apriori algorithm to identify frequent itemsets. From these itemsets, I generated association rules. This allowed me to see which platforms were commonly bundled together, such as if a game is on Mac and Windows, it is likely also available on Linux. A problem I had with the Association Rules part was some rules were statistically strong but not very common. Which made it tricky to know which rules were actually meaningful or useful.

Ian: I tackled questions seven through nine focusing on what impact various factors had on positive reviews with XGBoost Regression to identify the top most important game attributes for positive review outcomes, the impact of release timing on a game's percentage of positive reviews (done with time series analysis on mean total percent of positive reviews), and to see if we could group the data broadly into clusters using K-Means (which was not as successful as I would have liked due to high dimensionality in the data, though still accurate according to K-NN). I also wrote the introduction and overview. The greatest challenge was ensuring my models were computationally efficient enough to run on a lower-end computer as well as the high-dimensionality of the data (along with being sparse with many values that are low or zero). A lot of concessions had to be made in selecting only the top attributes and otherwise limiting the computation. Ultimately, it shows that having many attributes can lead to less than ideal analytic outcomes without some direction to shape how that data is used. The XGBoost Regression was quite successful, though revealed some potential double counting ("singleplayer" being both a tag and a category). Analyzing the release dates was only challenging in that it required eliminating the day and splitting it into two new attributes for easier analysis, otherwise it was a simple aggregation with an average.