Unsupervised Learning for PC Video Game Analysis: Clustering Based on Game, Market and User Data

Executive Summary

- The purpose of this project is to use various unsupervised learning methods to categorise PC Games based on multi-dimensional data without pre-defined labels. This allows for the discovery of hidden relationships within gaming data.
- Data was sourced from Steam, the largest PC gaming storefront and Metacritic, the foremost video game review aggregation site combining market metrics (Price, Sales) with reviewer feedback (user and critic scores).
- The analysis successfully identified distinct clusters of games, identify the important features for the categorisation (release date, critic score, % positive reviews on Steam), and classify the characteristics of each cluster in terms of the data. These clusters can be used to improve game recommendations, market analysis and genre classification.

Introduction:

The PC gaming market has witnessed a strong growth in recent years, with the global PC Game Market generating $80.27 billion (USD) revenue in 2023, projected to rise to 141.92 billion (USD) by 2028 [Source]. This has been driven by technological advancements, the growth of gaming distribution platforms combined with social features such as Steam and the rising demand for an immersive gameplay experience.

The problem of categorising PC games is both an intriguing and practical one. While games can be categorised according to gameplay and genre, combining this data with user reviews, critic feedback as well as market data can be difficult. This categorisation is important for games developers, who aim to target player demographics and players who seek recommendations based on their preferences.

In this project, I aim to answer the question ‘How can unsupervised learning be used to categorise PC games based on diverse data points from game, market and user data?’ I chose this project due to my interest in PC gaming, and the complexity, variety and availability of data make it an ideal candidate for a data-driven solution. I decided to use unsupervised learning to find and reveal hidden structures and relationships between data points, without the need for explicit labelling.

Methods:

Data Sources

The data used in this project was sourced from two prominent platforms in the PC Gaming Industry: Steam – the largest PC gaming platform with 132 million monthly active users and 75% of the PC Market Share in the US, and Metacritic (MC), regarded as the foremost review aggregation site for the video game industry. Combining these two data sources allows for a comprehensive analysis of PC Games, combining game metadata, player data, and market data. The Steam dataset contained game metadata and market data, while the Metacritic Dataset contained data based on player and critic reviews.

Steam Dataset Metacritic Dataset

Data cleaning

The MC dataset contained data for various consoles, so I started by trimming the dataset to only include those whose Platform was PC. I trimmed the Steam dataset to only include those that had a MC rating. I removed any duplicate names from both datasets, and kept only games whose name was in both datasets, to ensure both datasets had the same number of rows.

I removed columns from both datasets that I deemed unnecessary (e.g. empty columns, URL links etc.). I then performed an inner join on the 'Name' column to combine the datasets. I combined transformed columns (e.g. positive, negative reviews into total reviews, percentage positive). I changed the data type of columns to numeric e.g. (estimated owners from '20000-50000' using midpoint), removing rows with invalid data entries (1.8%) e.g. 'tbd'.

I identified a discrepancy between MC Critic Score on each dataset for 2.8% of datapoints. After modelling the difference as a normal distribution, I looked at entries that were more than 2 standard deviations (s.d.) away (0.9%). Upon manual review, 80% of these were due to naming differences between the two sources, referring to different releases e.g. 'The Wolf Among Us' referring to 'The Wolf Among Us: Chapter 1'. I removed these anomalies, as they referred to different releases, and such the combined data would not be useful. For the remaining 20%, I removed those lying outside of 2 s.d. and kept points inside, choosing to remove the inaccurate Steam MC Critic Score column instead.

I identified an error with the 'Price' column, where some games incorrectly had a price of 0. To fix this, I removed all entries with a price of 0 unless they had the 'Free to Play' tag.

Feature Selection

I computed a correlation matrix between columns, removing those that had a correlation coefficient >0.8. I then ran PCA and chose to keep 11 components to meet >95% cumulative explained variance.

I reformatted and combined the genre columns into a single column with data type list, ensuring no repetition within each list. I then looped through the column, removing any genres from each list with total count < 100. I then computed a correlation matrix, removing genres with overlap>90%, resulting in 18 genres and 99.95% of games covered.

I then created three datasets, one converting these 18 genres to a binary column, one converting only the 'Indie' genre (denoting Independent Game Studios), and one with genres removed, though

ultimately, I chose to continue analysis on only the genre removed dataset. This was due to poor clustering from the '18 genre' dataset and the 'Indie' column having too high feature importance on the 'Indie' column.

Exploratory Data Analysis

To understand the data better, I looked at the distribution of user scores, modelling them with both a normal and skewed normal distribution, and found the skewed distribution fit better using log-likelihood and AIC.

I then looked at the correlation between MC user score vs MC critic score, and user score vs percentage positive reviews on Steam, showing a positive correlation of 0.58, 0.62 respectively. I also looked at this for indie games vs non-indie games with minor differences but no major change. I performed a hypothesis test on the effect of price on MC User Score vs. MC Critic Score (as critics receive games for free), using Pearson correlation coefficient and Fisher's significance testing to conclude that price has a larger effect on MC User Score than MC Critic Score.

Unsupervised Learning:

Firstly, I split my data into 80% training data, 20% test data to prevent over-fitting. I then used auto-encoding to turn columns into latent variables, ensuring small validation loss by comparing different depths of latent space. I used k-means clustering using silhouette score (a measure of how good clusters are) and the elbow method to compute the best value of k. K-means clustering is good for looking at global properties of data set which is why I ran it first.

The best values were k=2,4 each with a silhouette score of 0.33 and 0.26 respectively. I computed the cluster means for each k-value and represented it as a heatmap, looking at the latent differences between clusters.

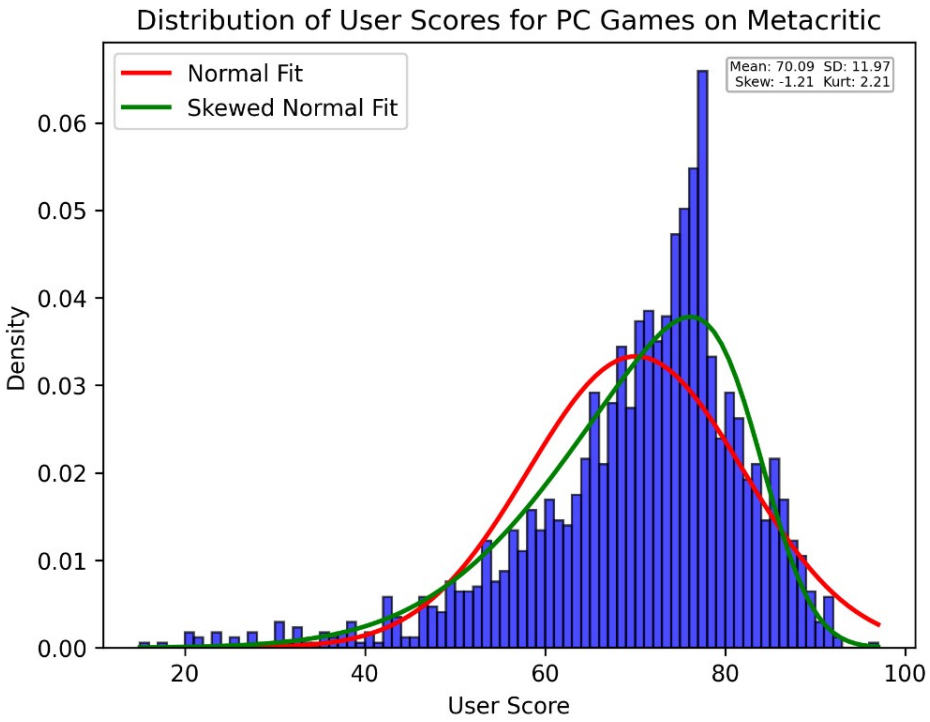
I also applied Gaussian Mixture Model (GMM) Clustering, experimenting with number of clusters though this gave considerably poorer clusters with lower silhouette values. I also looked at Density Based Clustering, experimenting with epsilon values for distance but this method detected too many outliers.

Still not satisfied, I then ran hierarchical clustering on non-auto-encoded data. Hierarchical clustering allows for better visualisation and understanding of the clustering because you build it from the ground up. I computed the dendrogram and looked at silhouette score vs. Euclidean distance to choose a Euclidean distance which in turn chose the number of clusters. Doing this gave me a silhouette score of 0.65 and 7 clusters.

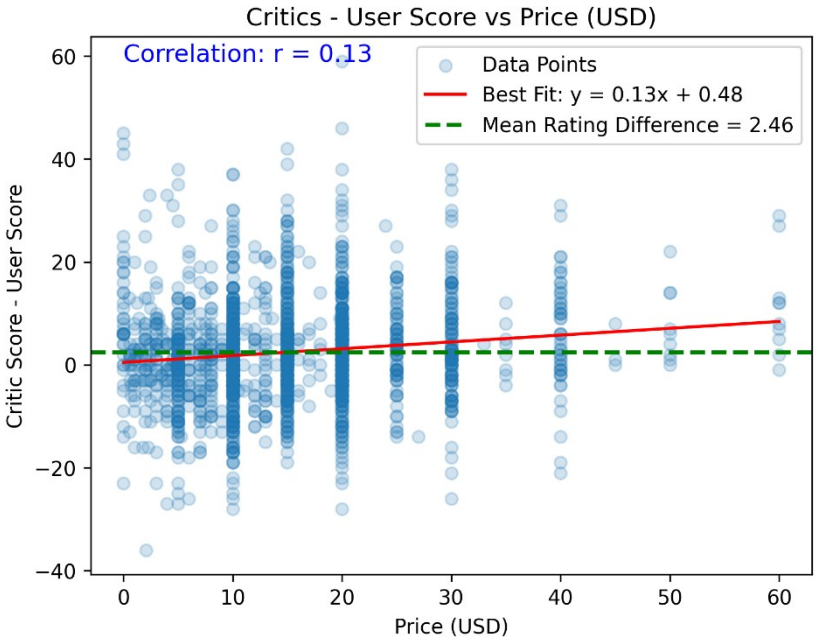
I then visualised all clusters using two methods: limiting the visualisation to 2 chosen latent features (except hierarchical), and PCA (all). Using random forest, I then computed feature importance on the columns, sorted them by importance and plotted them. With this I was happy with the clustering, applied the clustering to the test set and decided to compute pair plots in order to best categorise the clusters according to their features. After looking at the pair plots visually, I then identified the characteristics of each cluster.

Results

Initial Data Exploration



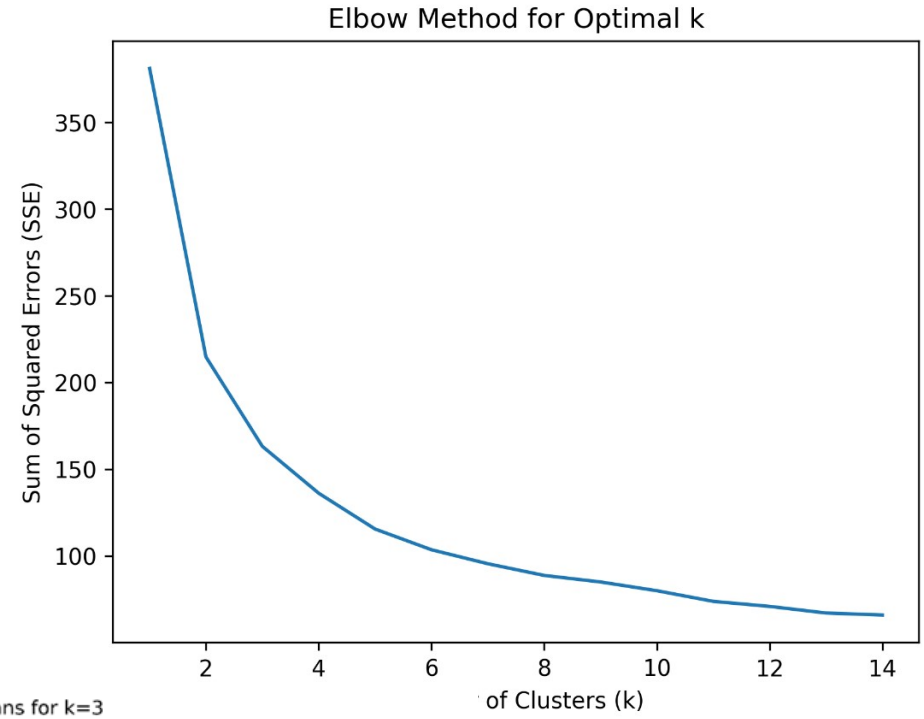
Graph showing the distribution of Metacritic User Score for PC Games, showing that the data best follows a skewed normal distribution.

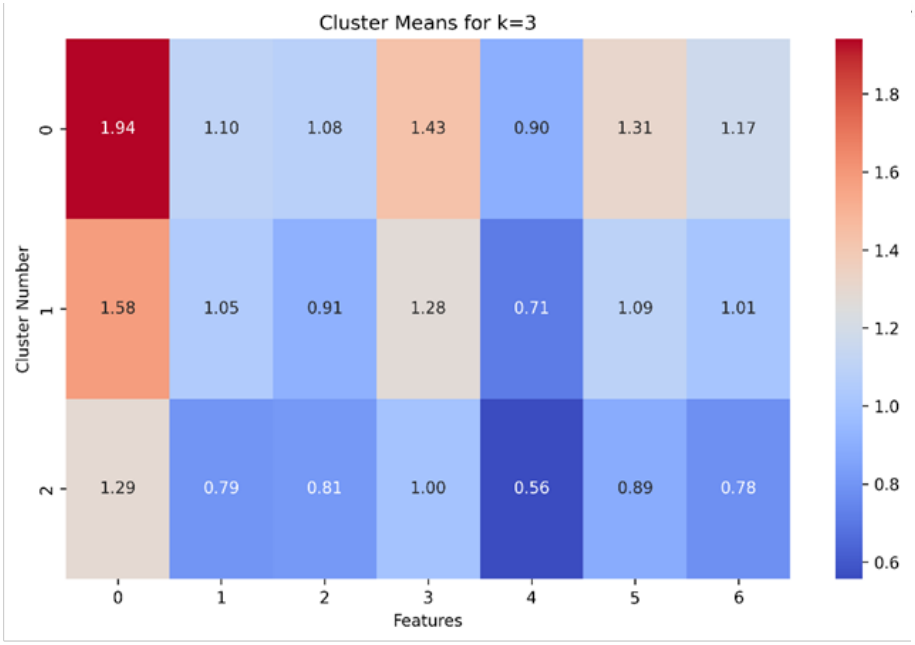


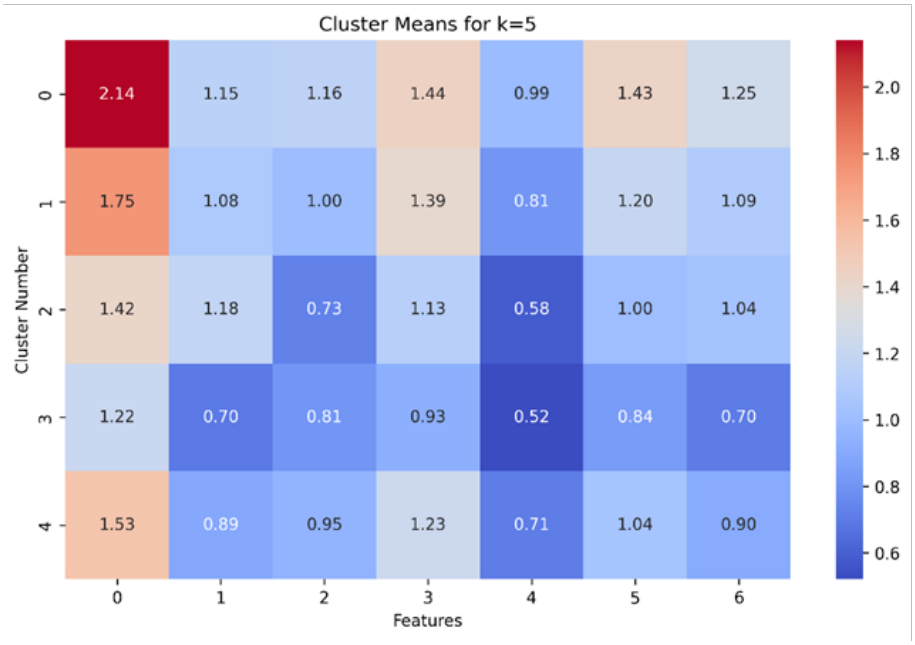
This graph shows User Score subtracted from Critic Score, plotted against price. The null hypothesis H_0 was that 'Price does not have a larger effect on user score than on critic score as price increases.' With H_1 'Price has a larger effect on user score than on critic score as price increases.' Using the Fisher Significance Testing, the probability that H_0 was true was 0.07% <1%, a level of statistical significance far smaller than my acceptance threshold so I rejected the null hypothesis and accepted H_1.

Unsupervised Learning

This graph shows how the sum of squared errors changes for varying values of k – number of clusters. From this graph I chose to look at k=3, k=5 using the elbow method to minimise in-cluster variance while preventing overfitting.



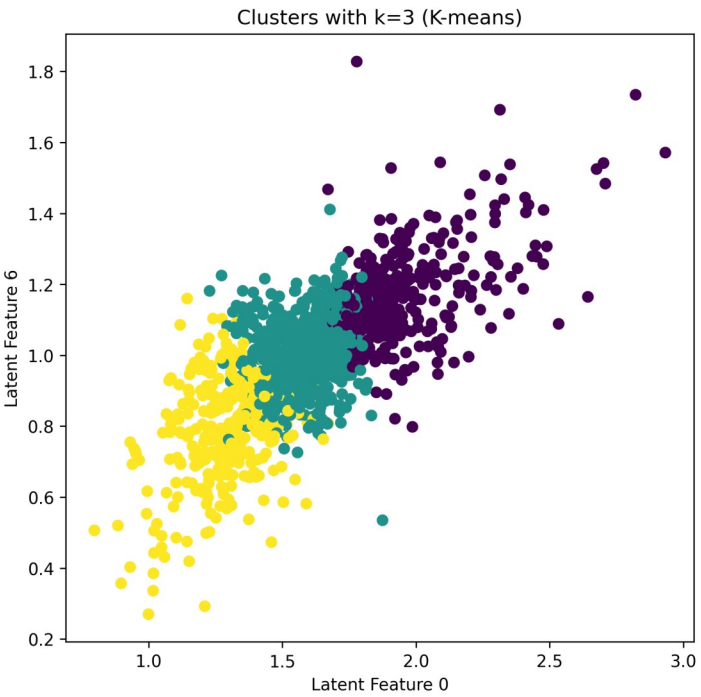


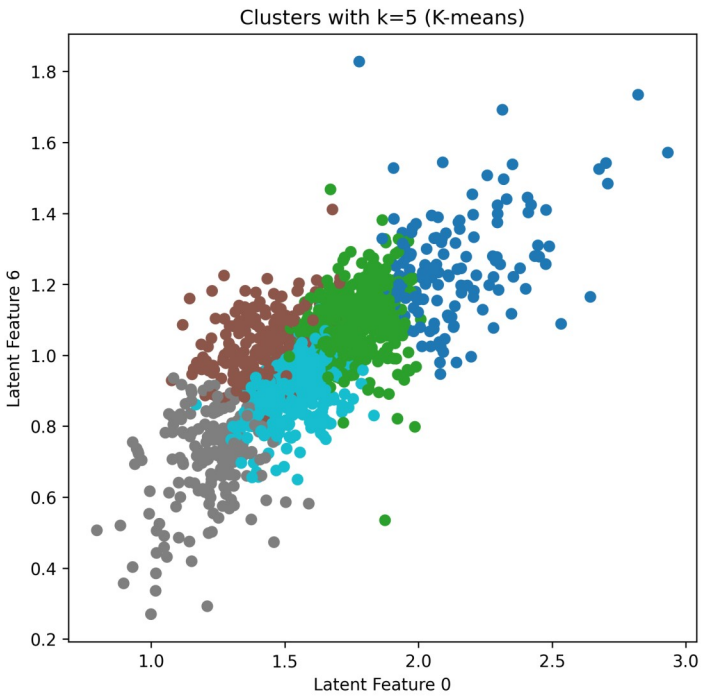


These heatmaps show the mean value of each latent feature for each cluster. From these we can see the distinguishing characteristics of each cluster in terms of the latent features.

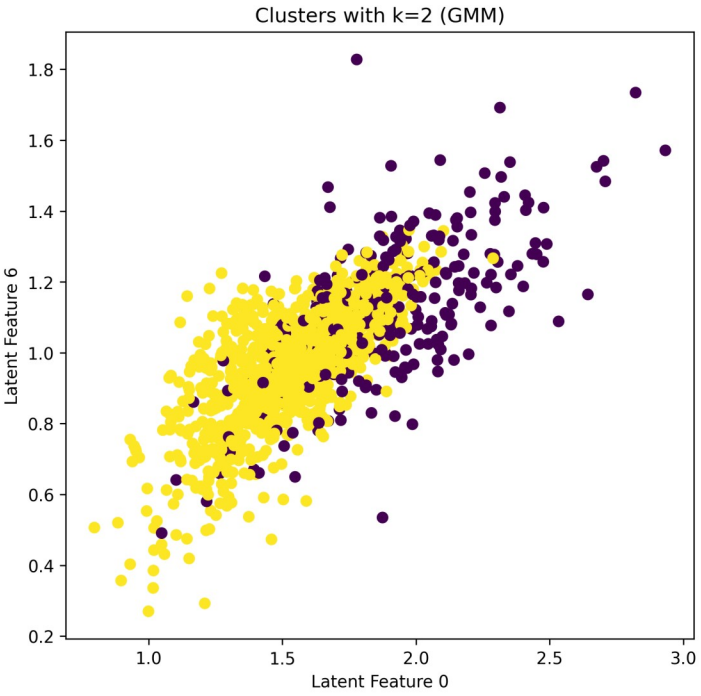
For example, for k=3, Cluster 0 is identified by having a high values for latent features, Cluster 1 having middling values and Cluster 2 having low values.

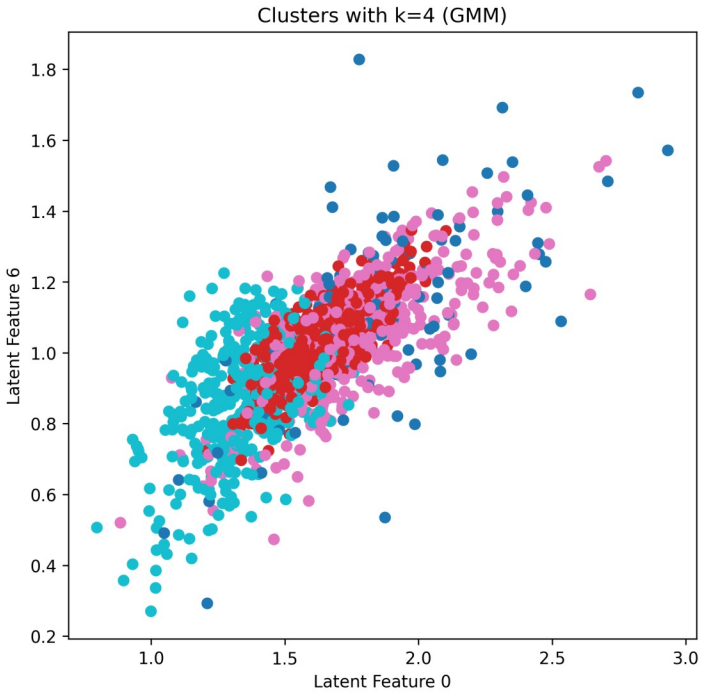
By contrast, for k=5 the clusters' characteristics are far less immediately obvious and rely on multiple variables to determine cluster

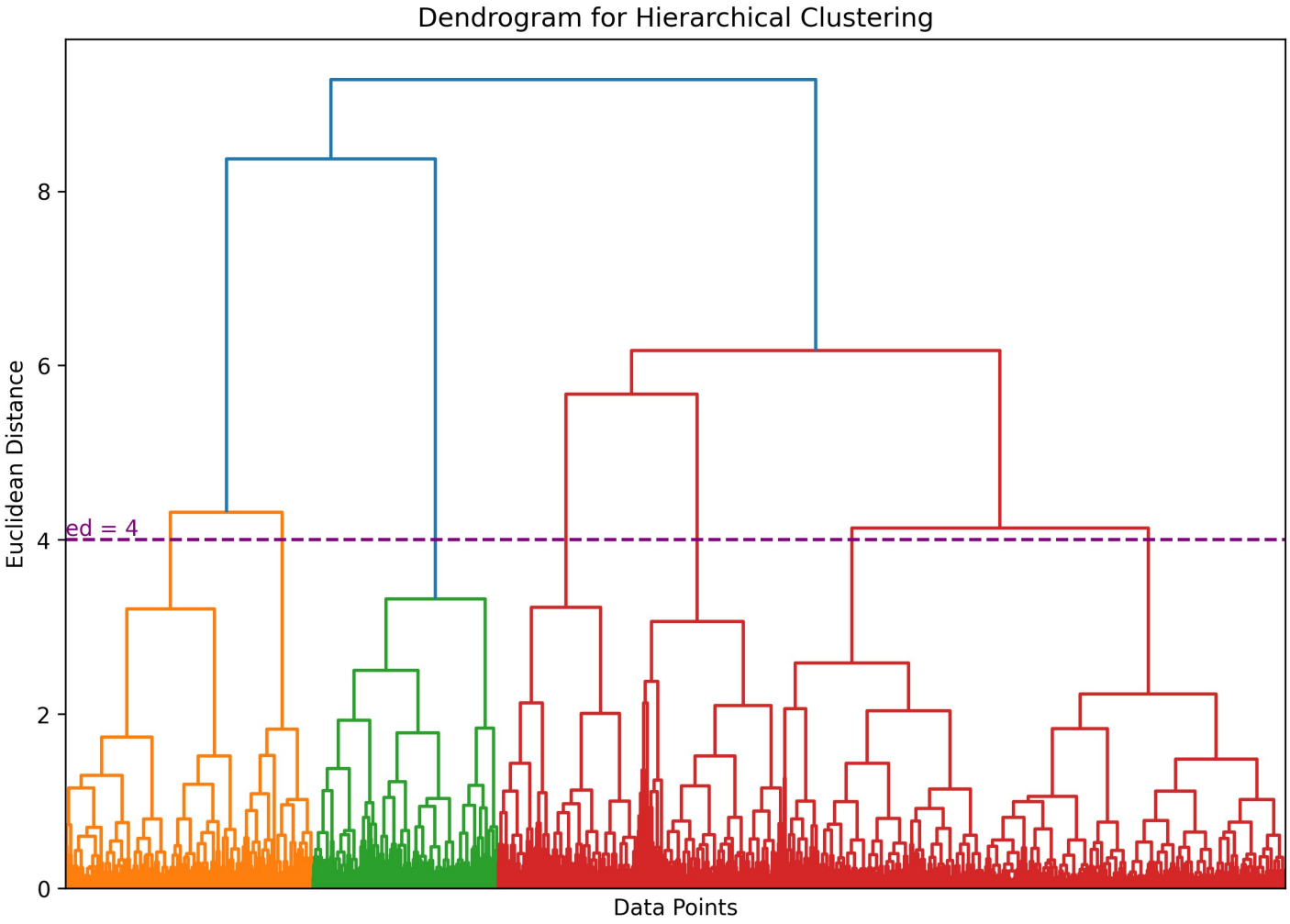




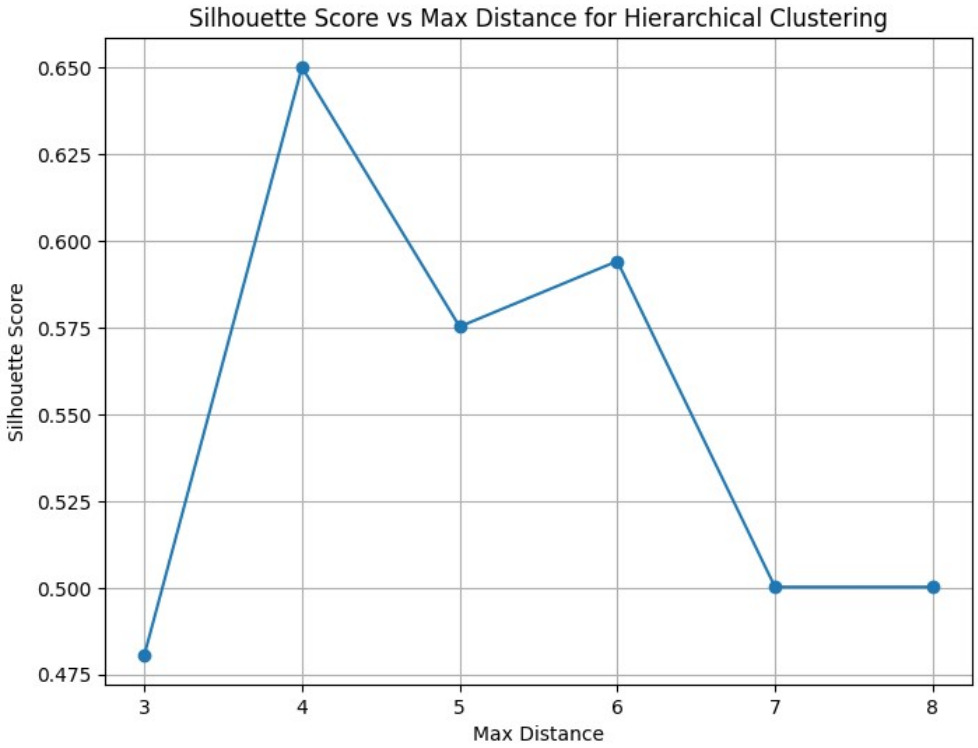
These graphs visualise the K-means clustering method (above) against the GMM clustering method (below). The data shows a 2D slice of the latent space, chosen to highlight the difference in clustering. As you can see the groups of points above are more well-defined and there is less overlap. This can be represented by the silhouette score, for which they are (0.30, 0.28) above and (0.19,0.09) below. For this reason, I chose to reject the clustering by GMM.

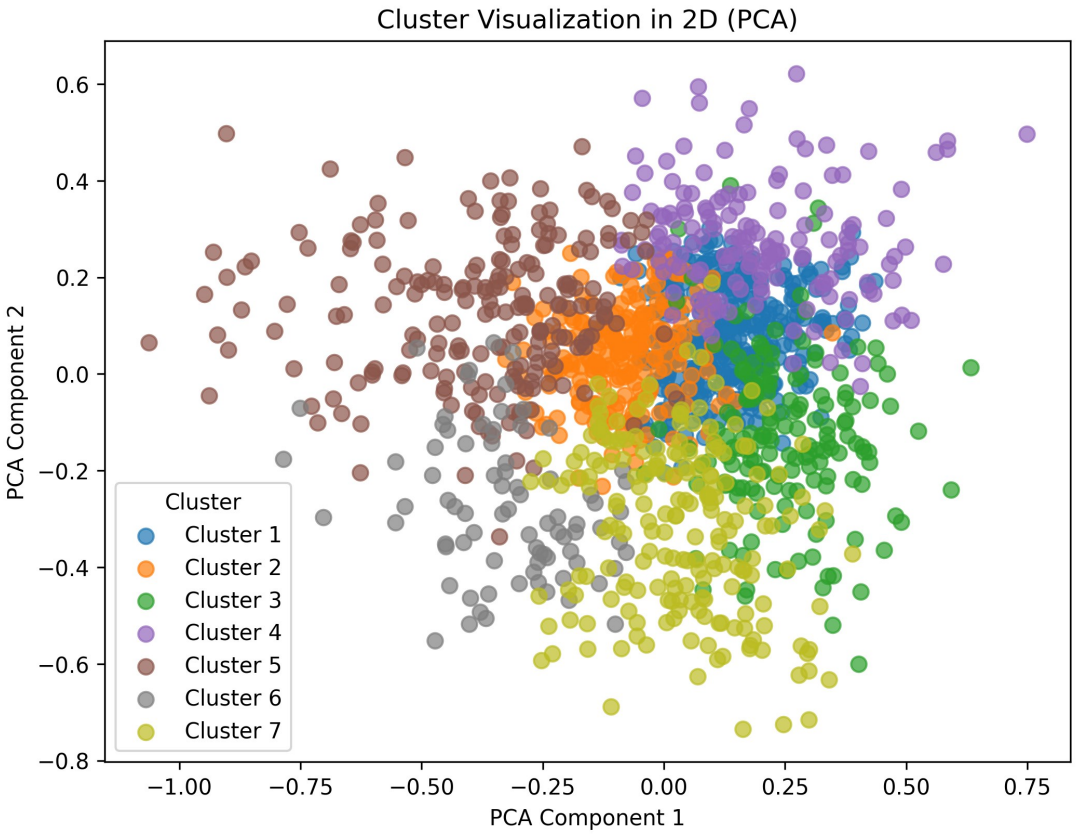




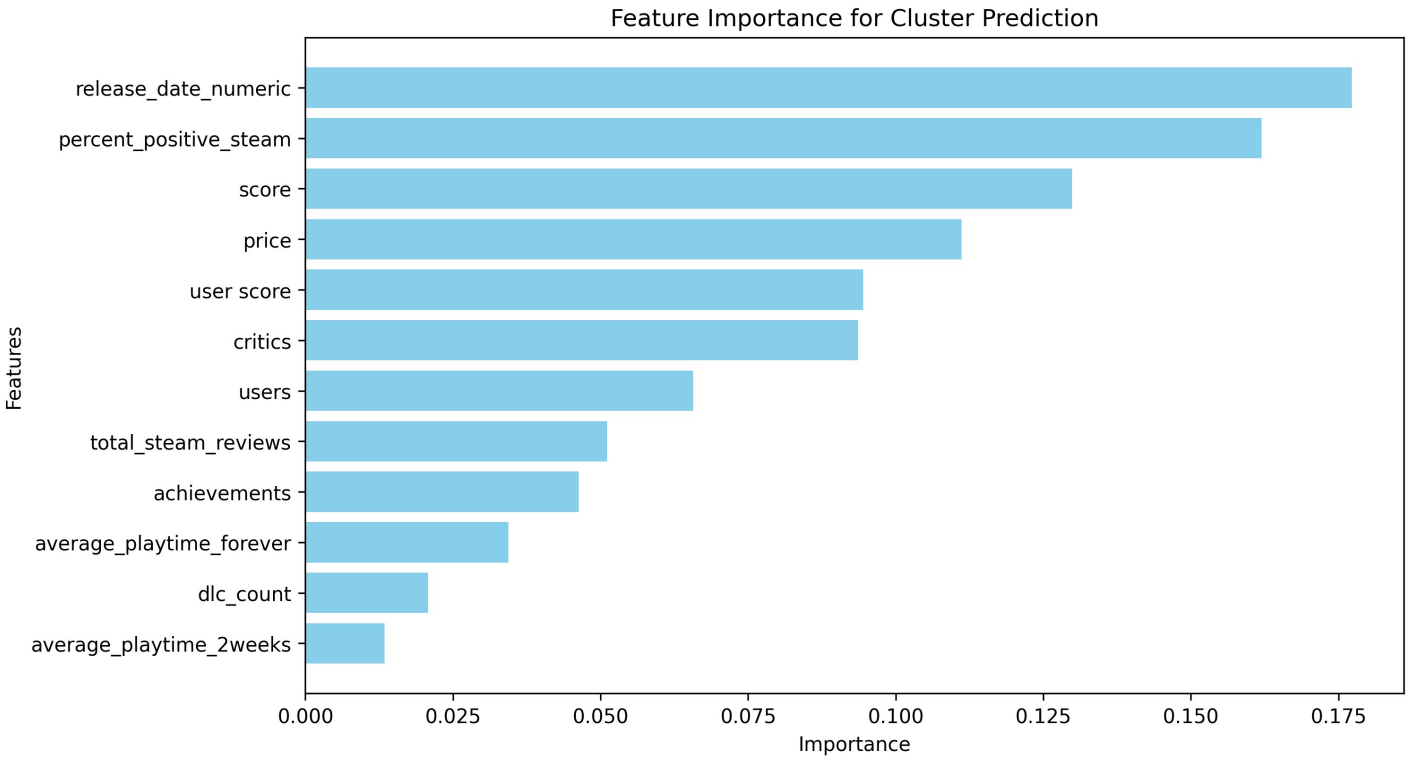


The above graph shows the dendrogram used in the process of hierarchical clustering. After computing the dendrogram, I computed the silhouette score as a function of distance, and used it to choose 4 as the distance. By drawing a horizontal line on the dendrogram at ed=4, we can see it makes 7 intersections corresponding to 7 clusters. Also note the high value of the silhouette score compared to previous methods, which is why I chose hierarchical clustering.

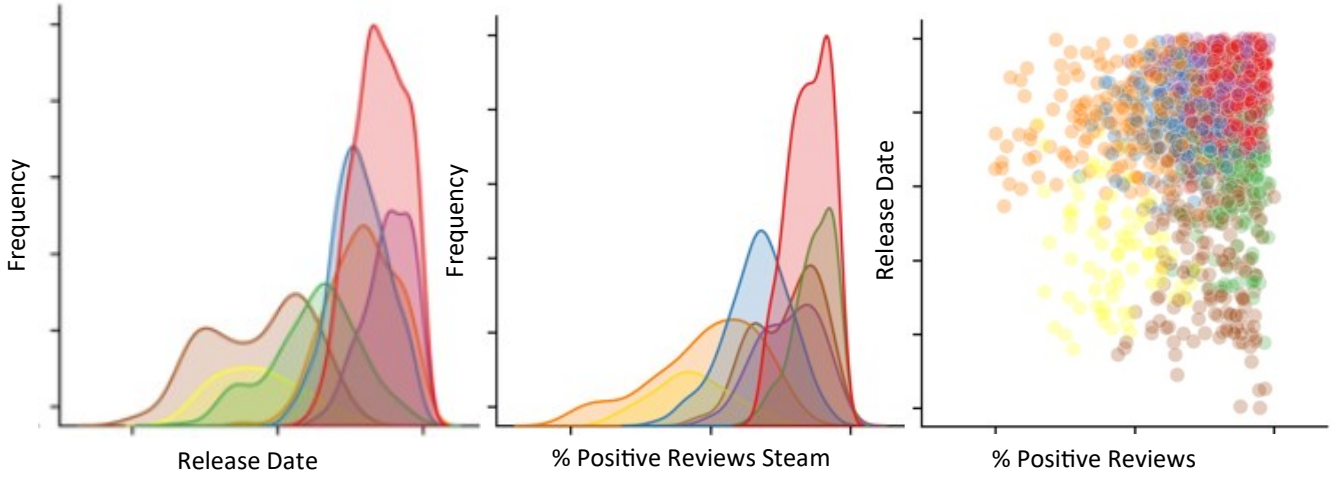




The above graph visualises the 7 clusters after doing PCA on the feature space to reduce it to 2 dimensions. As a result, it may seem that there is significant overlap, but that is largely due to the reduction of dimensionality of the space. We know this because of the high silhouette score from the clustering.



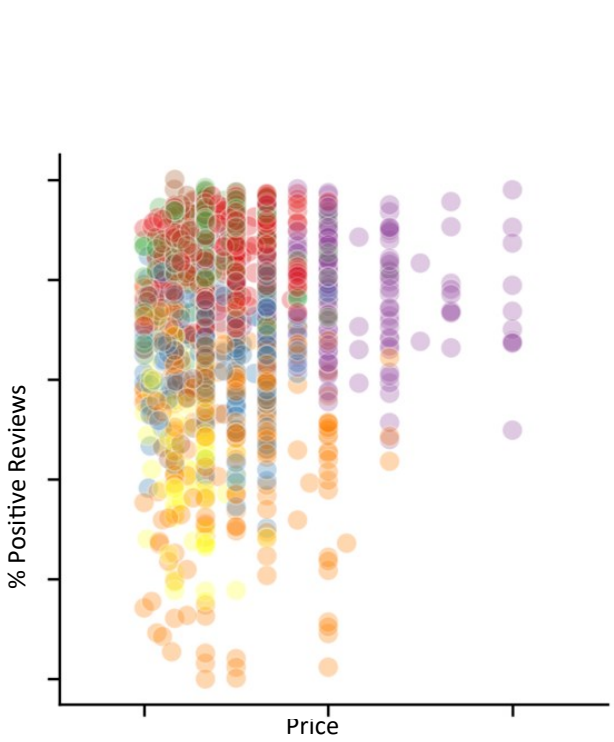
The above graph shows us the most important features for determining which cluster a data point belongs to. The most importance feature is given by release date, then by percentage positive reviews on steam and then by the Metacritic Critic Score. This graph is very useful when looking myopically at pair plots, as it can tell you which plots and relationships to look for.



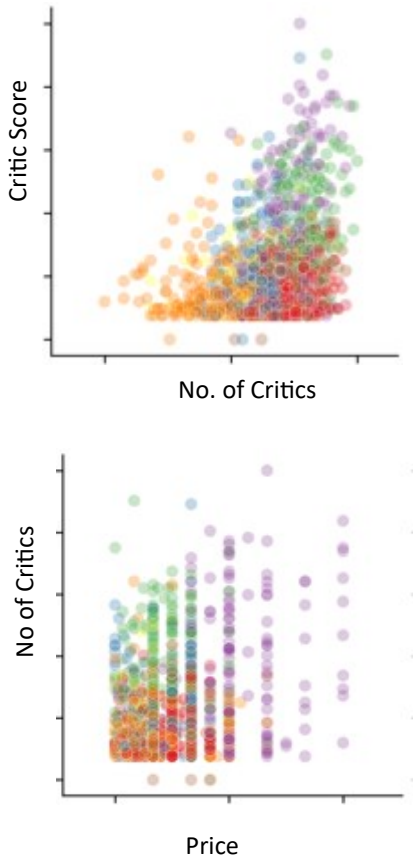
The left two graphs show the distribution for 'Release Date' and '% Positive Reviews Steam' for each cluster. For example, we can see games in the red, purple, blue and orange clusters have a more recent release date, whilst games in the yellow, brown and green clusters have older release dates. Similarly, we can see which clusters have a high percentage of positive reviews. The area coloured tells us the number of games within each cluster, e.g. the red cluster contains the most games.

However, looking at individual frequency diagrams often isn't enough to identify clusters. The plot on the right shows us the relationship between 'Release Date' and '% Positive Reviews Steam' by cluster. Now, we can start to see some separation.

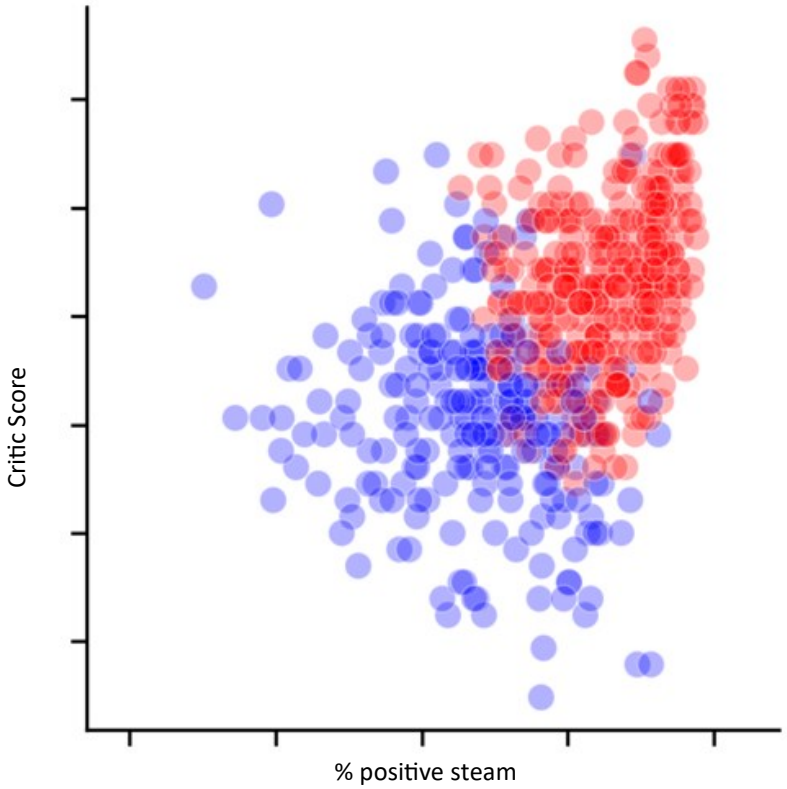
We can categorise that the yellow cluster on average contains games with an old release date and low percentage of positive reviews on Steam. Similarly, on average the orange cluster consists of games with a recent release date but low % of positive reviews. The brown cluster on average contains games with an old release date but a high % of positive reviews.



Looking at the above graph, we can categorise that on average games in the purple cluster will have a high price and a high % of positive reviews



We can see in the top graph that games in the green cluster have both a high critic score and number of critics alongside purple. However we can separate green and purple on the price as shown in the bottom graph.



Separating blue and red individually from the full pair plots were difficult to do visually but since we have categorised the other colours, we can plot only the red and blue clusters to see what distinguishes them from each other. From the above graph, so we can distinguish them from each other by looking at % positive reviews on steam and Critic Score from Meta-critic where red has high critic score and % positive reviews on steam, whereas blue may have at most a high score or high % positive reviews on steam but not both.

To summarise the identifiers of clusters, the table below:

Cluster Colour	Identifier
Yellow	Old Release Date + Low % Positive Reviews
Orange	Recent Release Date + Low % Positive Reviews
Brown	Old Release Date + High % Positive Reviews
Purple	High Price, High % Positive Reviews
Green	High Critic Number, High Critic Score, Low Price
Red	Separate from Blue by High Critic Score, High Positive % Reviews
Blue	Separate from Red

To conclude this section, after trying various clustering methods I chose hierarchical clustering for my categorisation, identified the important features used in making the categorisation and then identified the characteristics of each cluster.

Conclusion

The results demonstrated that unsupervised learning is a viable method for categorising games on a combination of game, market and user data. I successfully managed to cluster games using hierarchical clustering into 7 cohesive and distinct clusters. I was then able to identify the key features determining clusters – the top 3 being release date, % positive reviews on Steam and Critic Score from Metacritic. By computing pair plots, I was able to identify the characteristics of each cluster in terms of the original columns.

Recommendations for next steps:

- Apply the findings presented in this project to a game recommendation system that suggests similar games to users based on their preferences.
- Future research could incorporate Natural Language Processing, on textual reviews to analyse sentiment to enhance the depth and accuracy of the categorisation process.

By building on the findings, the approach could contribute to a more advanced game classification and recommendation system and offer deeper insight into player preferences.